THE FORMAL SEMANTICS OF COMPUTER LANGUAGES AND THEIR IMPLEMENTATIONS

by

Robert Milne

of

Oxford University

Oxford University Computing Laboratory

Programming Research Group Technical Microfiche TCF-2

NOTE

This monograph is a copy of a doctoral dissertation submitted to Cambridge University. It is distributed by the Programming Research Group as Photocopy PRG-X13, and is also available as Programming Research Group Technical Microfiche TCF-2.

Oxford University Computing Laboratory,
Programming Research Group,
45 Banbury Road,
Oxford.

SUMMARY

This dissertation contributes to the mathematical theory of computer languages by extending the formalism due to Scott and Strachey to cover language features not considered before and by developing a framework in which implementations can be analysed. The features handled by the extensions include parallel processes and modes which may be declared or coerced; in fact two radically different treatments of modes are related by an approach that can also be used to prove the equivalence of appropriate interpreters and compilers. Implementation techniques are described in terms of valuations which convert program texts into transformations of stacks and stores; thus particular techniques may be deemed to be correct when the resulting semantic equations correspond with the standard ones of Scott and Strachey. Formulating the pertinent correspondence involves the construction of predicates which are not monotonic but which nevertheless connect the equations in a recursive manner. The proofs that these predicates are satisfied throughout entire computations entail inductions on the structure of programs and must therefore be carried out for one language at a time; however the language discussed in this dissertation is large enough to justify the claim that the proofs supplied for it can readily be adapted to apply to other languages. The standard semantic equations are not suited to establishing the equivalence of some varieties of program, which must be done with the help of the additional information provided by equations which embody the essence of certain implementations. On one such relationship of equivalence, which asserts that every reasonable program may be replaced by one wherein identifiers denote locations only, depend theorems which compare several methods of implementing recursion.

CONTENT

CHAPTER ONE

STANDARD SEMANTICS

1.1. The mathematical basis.

1.1.1. The framework of computing theory.

The aim of this dissertation is to examine some of the implications of the mathematical theory of computing propounded by Scott [17], which views the meanings of programs as elements of domains subject to partial orderings. Although a typical domain will belong to a certain category of complete lattices, all the results which will be assumed or proved can be qualified in such a way that they remain valid when domains are taken to be partially-ordered sets in which every countable chain has a least upper bound. In particular, both these interpretations of the term 'domain' are among those that can provide a rigorous foundation for the theory of programming languages conceived by Strachey [21], wherein one member of the range of a function σ may be a function θ which can itself be applied to σ. The present work will adopt the notation of the rigorous formulation of this theory [20] unless otherwise stated; it will also presume familiarity with all the other papers by Scott and by Strachey cited in the bibliography.

In the later sections of this chapter standard semantics will be developed from the formulation mentioned above in order to describe Mal, a large computer language in which the types of variables are checked dynamically. This language has greater expressive power than Algol 60, but it does not provide some features of other languages, such as parallel processing and types which can be declared or coerced during compilation; those features requiring a further expansion of the formalism will be

considered from 3.4.1 onwards. A formal definition of Algol 68, which will not be presented below, has confirmed the adequacy of the treatment to be outlined in 3.6.1.

The abstractness which gives standard semantics its great elegance also renders it unsuitable for the analysis of certain kinds of relation between programs. After 2.1.1 classes of Mal programs will be shown to be equivalent to one another with the aid of store semantics, which makes explicit much of what is left implicit in standard semantics. As store semantics represents a sensible implementation technique, the assertion that it gives every Mal program a meaning which corresponds with the meaning provided by standard semantics can be said to entail the correctness of this technique. Another method of implementing languages is embodied in stack semantics, which will be discussed in 3.1.1; in contrast to that inherent in store semantics, this method is correct only when the programs to which it is applied have variables with severely restricted scopes. Both store semantics and stack semantics can be given forms in which the details of particular implementations are made manifest and in which finitary objects take the place of functions, but the necessary intricacies will not be described.

1.1.2. Preliminary conventions.

The symbolism to be adopted for handling sets is perfectly normal and therefore needs little explanation; suffice it to remark that if X is a set the members of which are sets then $\bigcup X$ will be the union of the members of X and $\bigcap X$ will be their intersection, whereas if X is a set of elements of a domain then $\bigsqcup X$ will be its join and $\bigsqcap X$ will be its meet. When X has only two members these operators may be replaced by small infixed versions. The usual convention that the disjunction and conjunction of

truth values ε_0 and ε_1 be written as $\varepsilon_0 \vee \varepsilon_1$ and $\varepsilon_0 \wedge \varepsilon_1$ respectively will be extended to sets as follows: given any set X comprising truth values $\bigvee X$ will signify the disjunction of the members of X and $\bigwedge X$ will signify their conjunction, so that if X is empty $\bigvee X$ will be *true* and $\bigwedge X$ will be *false*. Likewise when X is a set of integers $\bigvee X$ will be its largest element and $\bigwedge X$ will be its smallest element (provided that these entities exist); moreover if $X=\{\nu_0,\nu_1\}$ $\bigvee X$ and $\bigwedge X$ may give way to $\nu_0 \vee \nu_1$ and $\nu_0 \wedge \nu_1$. In 3.6.3 the sum and product of the members of a set of integers will be needed; for any such set X they will be provided by $\sum X$ and $\prod X$. This notation will be modified when sequences of sets are considered: $\bigcup\{X_n \mid n\geq 0\}$ and $\bigcap\{X_n \mid n\geq 0\}$ will be abbreviated to $\bigcup X_n$ and $\bigcap X_n$, and analogous changes will motivate the use of $\bigsqcup X_n$, $\bigsqcap X_n$, $\bigvee X_n$ and $\bigwedge X_n$.

Any function ϕ taking a lattice A into a lattice B is strict if $\phi\bot=\bot$, monotonic if $\phi\xi_0 \sqsupseteq \phi\xi_1$ whenever $\xi_0 \sqsupseteq \xi_1$, and continuous if every directed set $X \subseteq A$ satisfies $\phi(\bigsqcup X)=\bigsqcup\{\phi\xi \mid \xi\in A\}$; most of the functions which will be required in the present work are continuous, but in fact it would be enough to assume that they are such that $\phi(\bigsqcup X)=\bigsqcup\{\phi\xi \mid \xi\in X\}$ whenever X is a countable chain. The word 'function' will usually, but not invariably, refer to a continuous function; similarly $A \to B$ will generally be the lattice of all functions taking A into B. Till 1.2.2 $A \times B$ will be the set of all pairs $\langle \xi,\eta \rangle$ with $\xi\in A$ and $\eta\in B$ under the product ordering. The forms of bracketting adopted will all be allowed to nest, but parentheses will rarely surround the arguments of functions unless the parsing so decrees; since the application of functions will associate to the left (so that $\phi\psi\xi$ could be written as $(\phi\psi)\xi$) the domain denoted by $A \to B \to C$, say, will be regarded as $A \to [B \to C]$, not as $[A \to B] \to C$. Both continuous and discontinuous functions will have their effects on their arguments specified by means of

λ-notation [10]; examples of such specifications can be found by consulting the index.

One continuous function of particular importance is *fix*, which is constructed in such a way that if ϕ is any function mapping a domain A into itself then $fix\phi=\bigsqcup\phi^n\perp$, where $\phi^0=\lambda\xi.\perp$ and $\phi^{n+1}=\phi\circ\phi^n$ when $n\geq 0$. The definition of continuity ensures that $\phi(fix\phi)=fix\phi$ and $\xi\sqsupseteq fix\phi$ if ϕ is continuous and ξ is any member of A for which $\phi\xi=\xi$. Strictly speaking, *fix* should be set up anew every time a fresh domain A is encountered, but in fact this will not be done. Some functions (such as those to be mentioned in 1.3.1) will be given recursive definitions in which *fix* does not appear; there will, however, always be an obvious means of eliminating the circularity by introducing *fix*.

Another function which should really be provided for a 'universal' domain or for each lattice which requires it is *cut*; for every lattice A and every $\xi\in A$ other than $\top$ $cut\xi=\bigsqcap\{\xi_0\mid\top\sqsupset\xi\sqsupseteq\xi_0\sqsupset\perp\}$ and $cut\top=\top$. As ξ_0 is a bound variable of the term $\{\xi_0\mid\top\sqsupset\xi\sqsupseteq\xi_0\sqsupset\perp\}$ *cut* is strict and continuous when $\top\prec\top$.

If ξ_0 and ξ_1 are arbitrary elements of a domain A the relation $\xi_0\prec\xi_1$ will mean that for every directed set $X\subseteq A$ having $\bigsqcup X\sqsupseteq\xi_1$ there is some $\xi_2\in A$ such that $\xi_2\sqsupseteq\xi_0$. A domain A in which $\top\prec\top$ and $cut\xi\sqsupset\perp$ for all $\xi\in A$ with $\xi\sqsupset\perp$ will be described as 'slit'; a domain A in which $cut\xi=\xi$ for all $\xi\in A$ will be described as 'flat'.

Many fundamental mappings between domains can be classified as injections, projections or retractions. When p maps A into B it is an injection if it is continuous and if there is a continuous mapping q from B into A having $q\circ p=\lambda\xi.\xi$ and $p\circ q\sqsubseteq\lambda\eta.\eta$. Conversely a continuous function q taking B into A is a projection if it has a continuous inverse p for which $q\circ p=\lambda\xi.\xi$ and $p\circ q\sqsubseteq\lambda\eta.\eta$; from 1.2.8 onwards functions akin to $p\circ q$ will also be regarded as

projections. A retraction of A is a continuous and idempotent function which takes members of A into other members of A.

Although ~ will provide the usual denial of a truth value, in a somewhat unorthodox manner $\varepsilon_0 \supset \varepsilon_1$ will be understood to mean simply that unless ε_0 is *false* it must be identical with ε_1. When θ_1 and θ_2 belong to some domain the conditional expression $\varepsilon \rightarrow \theta_1, \theta_2$ will be an entity which is θ_1 if ε equals *true* and is θ_2 if ε equals *false*. Because *true* and *false* are only two of the four elements in T (the lattice of truth values), in 1.2.6, for instance, it will be necessary to know the values of $\varepsilon_0 \vee \varepsilon_1$, $\varepsilon_0 \wedge \varepsilon_1$, $\sim\varepsilon$ and $\varepsilon \rightarrow \theta_1, \theta_2$ when ε_0, ε_1 and ε are $\bot$ or $\top$. These values must be such that the resulting operators give continuous functions of ε_0, ε_1, ε, θ_1 and θ_2, but there is no other criterion governing them. It is convenient to let $\top \vee \varepsilon$, $\varepsilon \vee \top$, $\top \wedge \varepsilon$, $\varepsilon \wedge \top$, $\sim\top$ and $\top \rightarrow \theta_1, \theta_2$ be $\top$ for all $\varepsilon \in \mathsf{T}$ and to let $\bot \vee \varepsilon$, $\varepsilon \vee \bot$, $\bot \wedge \varepsilon$, $\varepsilon \wedge \bot$, $\sim\bot$ and $\bot \rightarrow \theta_1, \theta_2$ be $\bot$ if ε is *true*, *false* or $\bot$, but the feeling that the precise choice of these values is just a matter of taste will be justified in 1.3.5.

Often it is useful to be able to 'filter out' $\bot$ and $\top$ before applying the test for membership of a domain; this can be achieved by using not $\xi \in A$, which is *true* whenever ξ is a member of A, but $\xi : A$, which is $\bot$ when ξ is $\bot$, $\top$ when ξ is $\top$ and *true* when ξ is any other member of A. An element of A for which $\xi : A$ is *true* will be called 'proper'; $\bot$ and $\top$ are improper. The sharp distinction between $\xi \in A$ and $\xi : A$ will not be observed in the test for equality, $\xi_0 = \xi_1$: though this test will sometimes be understood to be a discontinuous relation having $\bot = \bot$ and $\top = \top$ both equal to *true*, usually a form like $\xi_0 = \xi_1$ will refer to a test which could be written more accurately as $(\xi_0 = \xi_1) \wedge (\xi_0 : A) \wedge (\xi_1 : A)$. The context should always be sufficient to clarify which version of the equality predicate is intended.

1.2. More abstract models for storage.

1.2.1. Arrangements for obtaining locations.

One feature common to all the languages we shall study is the store. As a first approximation to a domain S which models this we adopt a lattice of continuous functions mapping each member of a flat lattice of locations, L, to a pair comprising a truth value in the four element lattice T and a stored value in some lattice V; thus ignoring input and output $S=L\to[T\times V]$. We take *area* and *hold* to satisfy $\sigma\alpha=\langle area\alpha\sigma, hold\alpha\sigma\rangle$ for all $\sigma\in S$ and $\alpha\in L$; intuitively $hold\alpha\sigma$ represents what the location contains whereas $area\alpha\sigma$ indicates whether it is inside the region of store in use. This region can be extended by adjoining $new\sigma$, a location not currently within it. More accurately, *new* is any monotonic function from S to L such that if $area\alpha\sigma$ is proper for all proper $\alpha:L$ and if $area\alpha\sigma=false$ for some α then $new\sigma$ is proper and $area(new\sigma)\sigma=false$. Plainly there are many possible *new* functions and any stack implementation will have an operation which can be regarded as a restriction of one of them to proper stores. Additional constraints (such as those to be given in 3.1.3) can be imposed on them to bring about a closer correspondence with the particular storage allocation mechanisms involved. Notwithstanding this all these *new* functions possess the following property which is not shared with many practical implementations.

Suppose that σ_1 and σ_2 are members of S such that for all proper α $area\alpha\sigma_1$ is proper and equal to $area\alpha\sigma_2$ and such that $area\alpha\sigma_1=false$ for some such $\alpha:L$. By the postulate above $new\sigma_1$ and $new\sigma_2$ are proper elements of L. Furthermore $new(\sigma_1\sqcup\sigma_2)$ is proper as $area\alpha(\sigma_1\sqcup\sigma_2)=area\alpha\sigma_1\sqcup area\alpha\sigma_2=area\alpha\sigma_1$ for all proper α. Now $new(\sigma_1\sqcup\sigma_2)\sqsupseteq new\sigma_1\sqcup new\sigma_2$ (*new* being monotonic) so in fact $new(\sigma_1\sqcup\sigma_2)=new\sigma_1=new\sigma_2$ as L is flat.

With this model, then, the new location selected is independent of the contents of the store. Yet this should be a contingent truth about an implementation, not a necessary one, for if new_1 and new_2 are two functions satisfying the postulate above and if T is a summand of V it is natural to expect that $\lambda\sigma.hold\alpha\sigma\rightarrow new_1\sigma, new_2\sigma$ will satisfy the conditions for a *new* function.

1.2.2. Coalesced function spaces.

This blemish could be effaced by altering the properties of *new*. Abandoning monotonicity would cause difficulties whenever a fixed point was required and would run counter to our intuitions. The other property of *new*, however, can be weakened by permitting inferences to be made about $new\sigma$ only when σ can be reified. Thus we demand that *new* be a continuous function from S to L such that if $hold\alpha\sigma$ and $area\alpha\sigma$ are proper for all α:L and if $area\alpha\sigma=false$ for some α then $new\sigma$ is proper and $area(new\sigma)\sigma=false$. When this property is imposed on *new* the remarks of 1.2.1 become invalid, as $hold\alpha(\sigma_1\sqcup\sigma_2)$ need not be proper when $hold\alpha\sigma_1$ and $hold\alpha\sigma_2$ are. On the other hand, if $\sigma_1\alpha\sqsubseteq\sigma_2\alpha$ with $\sigma_1\alpha$ and $\sigma_2\alpha$ proper for all α:L and if $area\alpha\sigma_1=false$ for at least one proper location α then $new\sigma_1=new\sigma_2$. This accords with experience unless the lattice of real intervals is regarded as a summand of V.

Though the defect is removed by using this stipulation about *new* it remains disconcerting that the original *new* could be extrapolated from real stores to others in a way incompatible with practice. Basic functions such as *update* can give rise to similar anomalies unless they too are restricted to real stores; for instance it is difficult to envisage the effect of finding ι in a proper location or 0 in an improper one. Moreover the proof

that if the body of a while loop can be regarded as a multiple assignment then so can the entire loop involves assuming that $update\alpha\perp\sigma=\perp$ when $\alpha:L$ and $\sigma:S$. In the light of the above this assumption must be suspect until we provide a model in which it is inherent, although theorems like 2.6.9 are independent of the roles assigned to $\perp$ in L, V and S because no semantic equations make use of them.

In any store σ which can be reified neither $area\alpha\sigma$ nor $hold\alpha\sigma$ is improper when α is proper. There is no physical difference between the stores $\lambda\alpha.\langle false,true\rangle$ and $\lambda\alpha.(\alpha:L\rightarrow\langle false,true\rangle,\langle\perp,\perp\rangle)$; to avoid the possibility that the basic functions have different effects on them we therefore allow only one to belong to the domain of stores. Thus we take this domain to be the set of functions σ such that $area\alpha\sigma$ and $hold\alpha\sigma$ are proper unless α is $\perp$ or $\top$, when both are $\perp$ or $\top$.

Given lattices A and B with $\top\prec\top$ in A we define a monotonic and idempotent mapping j from $A\rightarrow B$ to $A\rightarrow B$ by $j=\lambda\phi.\bigwedge\{\phi\xi:B\mid\xi:A\}\rightarrow(\lambda\xi.\xi:A\rightarrow\phi\xi,\top),\top$.

Suppose that $A\twoheadrightarrow B$ is $\{j\phi\mid\phi\in A\rightarrow B\}$; when $X\subseteq A\twoheadrightarrow B$ we take its join in $A\twoheadrightarrow B$ to be $j(\bigsqcup X)$, where the join within the brackets is formed in $A\rightarrow B$. With respect to this definition $A\twoheadrightarrow B$ becomes a lattice which possesses the attribute of continuity [19] under the conditions detailed below.

We now dispense with our earlier version of the product of two lattices, henceforth taking $A\times B$ to be $\{\langle\xi,\eta\rangle\mid\xi:A\wedge\eta:B\}\cup\{\langle\perp,\perp\rangle,\langle\top,\top\rangle\}$ with $\langle\xi_1,\eta_1\rangle\sqcup\langle\xi_2,\eta_2\rangle=((\xi_1\sqcup\xi_2:A\wedge\eta_1\sqcup\eta_2:B)\rightarrow\langle\xi_1\sqcup\xi_2,\eta_1\sqcup\eta_2\rangle,\perp)$. This coalesced product is an analogue of the coalesced sum which will be adopted; in terms of it the domain of stores might be given by $S=L\rightarrow[T\times V]$, but actually in 1.3.1 we shall adopt a variant of this.

Note that under this convention the domain A^* to be described in 1.2.8 is such that if any member ξ^* has $\xi^*\downarrow\nu=\perp$ for some ν then $\xi^*=\perp$.

1.2.3. Proposition.

Suppose that A and B are continuous lattices such that A is flat and B is slit. Then $A\mapsto B$ is a slit continuous lattice.

⊲We shall show that if ϕ_1 and ϕ_2 are proper elements of $A\mapsto B$ then $\phi_1\prec\phi_2$ if and only if for every $\xi:A$ $\phi_1\xi\prec\phi_2\xi$ and for all except finitely many ξ $\phi_1\xi=cut(\phi_2\xi)$.

Suppose that ϕ_1 and ϕ_2 are proper and that $\phi_1\prec\phi_2$. Let Y be a directed set such that $\bigsqcup Y\sqsupseteq\phi_2\xi_0$ for some $\xi_0:A$; then $\bigsqcup\{\lambda\xi.\xi=\xi_0\to\eta,\phi_2\xi\,|\,\eta:Y\}\sqsupseteq\phi_2$ so, since $\phi_1\prec\phi_2$, there exists an $\eta_0\in Y$ for which $(\lambda\xi.\xi=\xi_0\to\eta_0,\phi_2\xi)\sqsupseteq\phi_1$. Moreover if there is a chain of finite subsets $X_1\subseteq X_2\subseteq X_3\subseteq\ldots$ having $\bigsqcup X_n=X\subseteq A$ and a set $\{\eta(\xi)\,|\,\xi:X\}$ with $\phi_2\xi\sqsupseteq\eta(\xi)$ for all $\xi:X$, $\{\lambda\xi.\xi:X\wedge\sim(\xi:X_n)\to\eta(\xi),\phi_2\xi\,|\,n\geq 0\}$ is a directed set of members of $A\mapsto B$ with join ϕ_2; hence for some n $(\lambda\xi.\xi:X\wedge\sim(\xi:X_n)\to\eta(\xi),\phi_2\xi)\sqsupseteq\phi_1$ and $\eta(\xi)\sqsupseteq\phi_1\xi$ when ξ is not in X_n. Thus $\phi_1\xi=cut(\phi_2\xi)$ unless $\xi\in X_n\cup\{\perp,\tau\}$.

Conversely suppose that for every $\xi:A$ $\phi_1\xi\prec\phi_2\xi$ and for all except finitely many ξ $\phi_1\xi=\bigsqcap\{\eta\,|\,\tau\sqsupset\phi_2\xi\sqsupseteq\eta\sqsupset\perp\}$. Let Z be a directed set with $\bigsqcup Z\sqsupseteq\phi_2$; we shall show that $\phi_0\sqsupseteq\phi_1$ for some $\phi_0\in Z$ and thus that $\phi_1\prec\phi_2$. For any proper $\phi:Z$ $\phi\xi\sqsupseteq\phi_1\xi$ except perhaps when ξ belongs to the finite set X having $\phi_1\xi\sqsupset cut(\phi_2\xi)$. Even when $\xi:X$, $\phi_1\xi\prec\phi_2\xi$ and $\bigsqcup\{\phi\xi\,|\,\phi\in Z\}\sqsupseteq\phi_2\xi$ so, as $\{\phi\xi\,|\,\phi\in Z\}$ is directed, $\phi(\xi)\xi\sqsupseteq\phi_1\xi$ for some $\phi(\xi)\in Z$. As Z is directed there is some $\phi_0\in Z$ with $\phi_0\sqsupseteq\bigsqcup\{\phi(\xi)\,|\,\xi:X\}$. Thus $\phi_0\xi\sqsupseteq\phi_1\xi$ for all $\xi\in A$ and $\phi_0\sqsupseteq\phi_1$.

Should Z be a sequence of members of $A\mapsto B$ with $\bigsqcup Z=\tau$ then $\bigsqcup\{\phi\xi\,|\,\phi\in Z\}=\tau$ for some proper $\xi:A$. Since $\tau\prec\tau$ in B $\phi\xi=\tau$ for some $\phi\in Z$ and $\phi=\tau$. Hence $\tau\prec\tau$ in $A\mapsto B$ and, more generally, $\phi_1\prec\phi_2$ if

and only if $\phi_1=\bot$, $\phi_2=\top$ or $\phi_1\xi\prec\phi_2\xi$ for all $\xi:A$ and $\phi_1\xi=\bigsqcap\{\eta\mid\top\supset\phi_2\xi\sqsupseteq\eta\supset\bot\}$ except at finitely many $\xi:A$.

Now let ϕ_0 be an arbitrary proper member of $A\mapsto B$. By the continuity of B we can select $\eta(\xi):B$ such that for every $\xi:A$ $\eta(\xi)\prec\phi_0\xi$ and thus $cut(\phi_0\xi)\prec\phi_0\xi$; hence if X is any finite subset of A $(\lambda\xi.\xi\in X\to\eta(\xi),cut(\phi_0\xi))\prec\phi_0$ by the characterization above. As B is continuous and slit,

$\phi_0=\bigsqcup\{\bigsqcup\{\lambda\xi.\xi\in X\to\eta(\xi),cut(\phi_0\xi)\mid\bigwedge\{\eta(\xi)\prec\phi_0\xi\wedge\eta(\xi):B\mid\xi:X\}\}\mid X\subseteq A\wedge(X\text{ is finite})\}$

and $A\mapsto B$ is continuous. Finally, taking X to be the empty set shows that $cut\phi_0=\bot$ when $\phi_0=\bot$, so $A\mapsto B$ is slit and the proof is complete.⊳

The hypotheses of the proposition may appear bizarre but in fact they are satisfied by those lattices to which it will be applied. Although these hypotheses will be the ones invoked below the next result will show that they are not the only possibilities.

1.2.4. Proposition.

Suppose that A and B are continuous lattices such that A is finite and $\top\prec\top$ in B. Then $A\mapsto B$ is a continuous lattice.

⊲Any retraction of a continuous lattice gives rise to a continuous lattice, so we need show only that the mapping j defined above is a retraction of $A\to B$ into itself. Let Z be a directed set in $A\to B$; as j is monotonic it is sufficient to prove that $\bigsqcup\{j\phi\mid\phi\in Z\}\sqsupseteq j\phi_0$ where $\phi_0=\bigsqcup Z$ and $j\phi_0\supset\bot$.

If $j\phi_0$ is proper $\phi_0\xi$ is proper for all proper $\xi:A$, and given any $\eta(\xi):B$ with $\eta(\xi)\prec\phi_0\xi$ there is some $\phi(\xi):Z$ such that $\phi(\xi)\xi\sqsupseteq\eta(\xi)$. As Z is directed there is a $\phi_1\in Z$ with $\phi_1\sqsupseteq\bigsqcup\{\phi(\xi)\mid\xi:A\}$ and thus $\phi_1\xi\sqsupseteq\eta(\xi)$ for all proper $\xi:A$. Hence for every choice of proper $\eta(\xi):B$ such that $\eta(\xi)\prec\phi_0\xi$ for every $\xi:A$ there is a $\phi_1\in Z$ having $j\phi_1\xi\sqsupseteq\eta(\xi)$ for all $\xi:A$. Because B is continuous and $\eta(\xi)$ is arbitrary $\bigsqcup\{j\phi\xi\mid\phi\in Z\}\sqsupseteq\phi_0\xi=(j\phi_0)\xi$ for every

proper $\xi:A$. Necessarily $j\phi_1\bot=j\phi_0\bot$ and $j\phi_1\top=j\phi_0\top$ when ϕ_1 is a proper member of Z, so $\bigsqcup\{j\phi\,|\,\phi\in Z\}\sqsupseteq j\phi_0$.

If $j\phi_0=\top$, $\phi_0\xi=\bot$ for every proper $\xi:A$ and $\phi_0\xi=\top$ for some proper $\xi:A$, so there exist $\phi(\xi)\in Z$ such that $\phi(\xi)\xi=\bot$ for every proper $\xi:A$ and $\phi(\xi)\xi=\top$ for at least one proper $\xi:A$. Taking $\phi_1\in Z$ with $\phi_1\sqsupseteq\bigsqcup\{\phi(\xi)\,|\,\xi:A\}$, $j\phi_1=\top$ as $\phi_1\xi=\bot$ for all proper $\xi:A$ and $\phi_1\xi=\top$ for some proper $\xi:A$. Thus $\bigsqcup\{j\phi\,|\,\phi\in Z\}=\top\sqsupseteq j(\bigsqcup Z)$; this proves the result.⊁

In fact the continuity of j requires A to be finite, for given $X\subseteq A$ define $\phi(X)=\lambda\xi.\bigvee\{\xi_0\prec\xi\,|\,\xi_0\in X\}\to\xi,\bot$; $\{\phi(X)\,|\,X\subseteq A\wedge(X\text{ is finite})\}$ is a directed set with join $\lambda\xi.\xi=j(\lambda\xi.\xi)$. When A is infinite, however, $\bigsqcup\{j(\phi(X))\,|\,X\subseteq A\wedge(X\text{ is finite})\}=\bot$.

Suppose that $p':A_0\to A_1$ and $p'':B_0\to B_1$ are injections on continuous lattices with $\top\prec\top$ and that $q':A_1\to A_0$ and $q'':B_1\to B_0$ are the reciprocal projections. Let j' be the mapping of $A_0\to B_0$ into $A_0\twoheadrightarrow B_0$ regarded as a subset of $A_0\to B_0$ and let j'' be the corresponding mapping on $A_1\to B_1$. Define $p'\twoheadrightarrow p''=j''\circ(p'\to p'')$ and $q'\twoheadrightarrow q''=j'\circ(q'\to q'')$; if j' is continuous, $p'\circ q'=\lambda\xi.\xi$ and $p''\top=\top$ then $p'\twoheadrightarrow p''$ is an injection with reciprocal projection $q'\twoheadrightarrow q''$. In general A_0 is not finite so j' is not continuous, but there are further conditions which ensure the continuity of $p'\twoheadrightarrow p''$ and $q'\twoheadrightarrow q''$.

1.2.5. Proposition.

Suppose that $p':A_0\to A_1$ and $p'':B_0\to B_1$ are injections on continuous lattices with $\top\prec\top$ and that $q':A_1\to A_0$ and $q'':B_1\to B_0$ are the reciprocal projections. Suppose further that p' is an isomorphism, that $p''\top=\top$ and that if $q''(cut\eta)=\bot$ for any $\eta:B_1$ $q''\eta=\bot$. Then $p'\twoheadrightarrow p''$ is an injection with reciprocal projection $q'\twoheadrightarrow q''$ whilst if B_1 is slit so is B_0.

⊀Note first that $p''\top=\top$ if and only if when $q''\eta=\top$ necessarily $\eta=\top$, since $q''(p''\top)=\top$.

Let $\phi:A_0 \mapsto B_0$ and $\xi \in A_1$; then $p''(\phi(q'\xi))=\bot$ if and only if $\phi(q'\xi)=\bot$ and hence if and only if $\xi=\bot$, whilst $p''(\phi(q'\xi))=\top$ if and only if $\phi(q'\xi)=\top$ and thus if and only if $\xi=\top$. Thus $p' \mapsto q' = \lambda\phi.q' \circ \phi \circ p''$ and $(q' \mapsto q'') \circ (p' \mapsto p'') = \lambda\phi.j'(q'' \circ p'' \circ \phi \circ q' \circ p') = \lambda\phi.\phi$. Furthermore,

$$q' \mapsto q'' = \lambda\phi.\bigwedge\{q''(\phi(p'\xi)):B_0 \mid \xi:A_1\} \to (\lambda\xi.\xi:A_0 \to q'' \circ \phi \circ p', \top), \top$$
$$= (\lambda\phi.\bigwedge\{q''(\phi\xi):B_0 \mid \xi:A_1\} \to q'' \circ \phi \circ p', \top)$$

so that $(p' \mapsto p'') \circ (q' \mapsto q'') \sqsubseteq \lambda\phi.\phi$ and it remains to be shown only that $q' \mapsto q''$ is continuous.

Suppose that Z is a directed set in $A_1 \mapsto B_1$ and that $\phi_0 = \bigsqcup Z$; we shall show that $\bigsqcup\{(q' \mapsto q'')\phi \mid \phi \in Z\} \sqsupseteq (q' \mapsto q'')\phi_0$.

If $(q' \mapsto q'')\phi_0$ is proper, $q''(\phi_0\xi)$ is proper for all proper $\xi:A_1$. Now for every $\phi:Z$ and $\xi:A_1$ $\phi_0\xi \sqsupseteq \phi\xi \sqsupseteq cut(\phi_0\xi)$, so if $q''(\phi\xi)=\bot$ in fact $q''(\phi_0\xi)=q''(cut\phi_0\xi)=\bot$ contrary to hypothesis. Hence whenever $\phi:Z$ and $\xi:A_1$ are proper $q''(\phi\xi)$ is proper, thereby proving that $\bigsqcup\{(q' \mapsto q'')\phi \mid \phi \in Z\} = \bigsqcup\{q'' \circ \phi \circ p' \mid \phi \in Z\} = q'' \circ \phi \circ p'$.

If $(q' \mapsto q'')\phi_0=\top$, for some proper $\xi:A_1$ $q''(\phi_0\xi)=\top$ so $\phi_0\xi=\top$ and $\phi_0=\top$; hence as $\top \prec \top$ in B_1, $\phi_1\xi=\top$ for some $\phi_1 \in Z$ and $\phi_1=\top$, giving $\bigsqcup\{(q' \mapsto q'')\phi \mid \phi \in Z\}=\top$.

Note also that if $(q' \mapsto q'')(cut\phi)=\bot$ for some $\phi:A_0 \mapsto B_0$, $q''(cut\phi\xi)=\bot$ for some $\xi:A_1$ so that $q''(\phi\xi)=\bot$ by hypothesis and $(q' \mapsto q'')\phi=\bot$.

Should $cut\eta_1$ be proper for all proper $\eta_1:B_1$, take $\eta_1=p''\eta_0$ for some proper $\eta_0:B_0$. As $q''\eta_1=\eta_0=\bot$, $q''(cut\eta_1)=\bot$ and

$$cut\eta_0 = \bigsqcap\{\eta_2 \mid \top = p''\eta_0 \sqsupseteq p''\eta_2 = \bot\}$$
$$= \bigsqcap\{q''(p''\eta_2) \mid \top = \eta_1 \sqsupseteq p''\eta_2 = \bot\}$$
$$\sqsupseteq \bigsqcap\{q''\eta_2 \mid \top = \eta_1 \sqsupseteq \eta_2 = \bot\}$$
$$\sqsupseteq q''(cut\eta_1)$$
$$= \bot.\triangleright$$

Thus the set of slit continuous lattices is closed under the creation of $A \twoheadrightarrow B$ from a flat lattice A and a slit lattice B, and any projection of B of the form above induces one of $A \twoheadrightarrow B$ which is also of this form. An analogous proof shows that the set is also closed under the construction of arbitrary products. Here we content ourselves with a more modest result for which we take $q' \times q''$ to be $\lambda\langle \xi,\eta\rangle .q'\xi{:}A_0 \wedge q''\eta{:}B_0 \rightarrow \langle q'\xi, q''\eta\rangle , \top$ when $q'{:}A_1 \rightarrow A_0$ and $q''{:}B_1 \rightarrow B_0$.

1.2.6. Proposition.

If A and B are continuous lattices with $\top \prec \top$ so is $A \times B$, whereas if A and B are slit so is $A \times B$. Moreover suppose that $q'{:}A_1 \rightarrow A_0$ and $q''{:}B_1 \rightarrow B_0$ are projections such that $q'\xi = \top$ implies $\xi = \top$, $q''\eta = \top$ implies $\eta = \top$, $q'(cut\,\xi) = \bot$ implies $q'\xi = \bot$ and $q''(cut\,\eta) = \bot$ implies $q'\eta = \bot$; then $q' \times q''$ is a projection with analogous properties.

⊲We can show by methods akin to those of 1.2.5 that $A \times B$ is a retraction of $\{\langle \xi,\eta\rangle \mid \xi \in A \wedge \eta \in B\}$. Hence if A and B are continuous so is $A \times B$, and when $\xi{:}A$ and $\eta{:}B$ $cut\langle \xi,\eta\rangle = cut\,\xi{:}A \wedge cut\,\eta{:}B \rightarrow \langle cut\,\xi, cut\,\eta\rangle , \top$.

Because $A_0 \times B_0$ is a retraction of $\{\langle \xi,\eta\rangle \mid \xi \in A_0 \wedge \eta \in B_0\}$ $q' \times q''$ is continuous; furthermore, if $(q' \times q'')\langle \xi,\eta\rangle = \langle \top,\top\rangle$ $q'\xi = \top$ or $q''\eta = \top$ so that $\langle \xi,\eta\rangle = \langle \top,\top\rangle$. If $(q' \times q'')(cut\langle \xi,\eta\rangle) = \langle \bot,\bot\rangle$ while A_0 and B_0 are slit $q'(cut\,\xi) = \bot$ or $q''(cut\,\eta) = \bot$ so that $(q' \times q'')\langle \xi,\eta\rangle = \langle \bot,\bot\rangle$.⊳

Henceforth when writing elements of domains we shall frequently leave implicit the effect of passing to the quotient space $A \twoheadrightarrow B$ or $A \times B$. Thus $\lambda\xi.\xi = \xi_0 \rightarrow \bot, \phi_0\xi$ and $\langle \xi_0, \bot\rangle$ may arise when $\lambda\xi.\bot$ and $\langle \bot,\bot\rangle$ are meant. Instances of this will be inherent in the definitions of 1.3.2, which will introduce functions acting on environments.

We shall now verify that a suitable class of slit continuous lattices is closed under the construction of inverse limits.

1.2.7. Proposition.

Suppose that $\{B_n \mid n\geq 0\}$ is a sequence of slit continuous lattices having for each $n\geq 0$ a projection $j_n: B_{n+1} \to B_n$ such that $j_n\eta=\top$ implies $\eta=\top$ and $j_n(cut\eta)=\bot$ implies $j_n\eta=\bot$. Then the resulting inverse limit B is a slit continuous lattice.

⊲Certainly B is a continuous lattice, so we need only verify that it is slit. Let p_n be the natural injection of B_n into B and let q_n be the reciprocal projection. We shall show that for any η:B $\{cut(q_n\eta) \mid n\geq 0\}$ is an element of B having $q_n(cut\eta)=cut(q_n\eta)$ for all $n\geq 0$.

For every $n\geq 0$ and $\eta: B_n$ $q_n\eta=j_n(q_{n+1}\eta)\sqsupseteq j_n(cut(q_{n+1}\eta))$, so if $q_n\eta=\bot$ $j_n(cut(q_{n+1}\eta))=\bot$. If $q_n\eta\sqsupset\bot$ on the other hand, $q_{n+1}\eta\sqsupset\bot$ and $j_n(cut(q_{n+1}\eta))\sqsupset\bot$ from the hypotheses, so that $j_n(cut(q_{n+1}\eta))\sqsupseteq cut(q_n\eta)$; as B_n is slit we have also $cut(q_n\eta)\sqsupset\bot$, giving $q_{n-1}(p_n(cut(q_n\eta)))\sqsupseteq cut(q_{n+1}\eta)$ and $cut(q_n\eta)\sqsupseteq j_n(cut(q_{n+1}\eta))$. Hence for all possible cases $cut(q_n\eta)=j_n(cut(q_{n+1}\eta))$ and $\{cut(q_n\eta) \mid n\geq 0\}$ is a member of B.

Suppose that η_0:B; then for any η_1:B with $\eta_0\sqsupseteq\eta_1\sqsupset\bot$ there is some least $m\geq 0$ with $q_n\eta_1=\bot$ when $n\geq m$ and thus with $q_n\eta_1\sqsupseteq cut(q_n\eta_0)$ when $n\geq m$. If $m>0$ $q_{m-1}\eta_1=j_m(q_m\eta_1)\sqsupseteq j_m(cut(q_m\eta_0))=cut(j_m(q_m\eta_0))=cut(q_{m-1}\eta_0)$ in contradiction to the minimal nature of m. Accordingly we can assume that $\bigsqcap\{q_n\eta_1 \mid \top\sqsupset\eta_0\sqsupseteq\eta_1\sqsupset\bot\}\sqsupseteq cut(q_n\eta_0)$ for every $n\geq 0$. However

$$\begin{aligned} q_n(cut\eta_0) &\sqsupseteq q_n(\bigsqcap\{p_n(q_n\eta_1) \mid \top\sqsupset\eta_0\sqsupseteq\eta_1\sqsupset\bot\}) \\ &\sqsupseteq q_n(p_n(\bigsqcap\{q_n\eta_1 \mid \top\sqsupset\eta_0\sqsupseteq\eta_1\sqsupset\bot\})) \\ &= \bigsqcap\{q_n\eta_1 \mid \top\sqsupset\eta_0\sqsupseteq\eta_1\sqsupset\bot\} \\ &\sqsupseteq q_n(cut\eta_0), \end{aligned}$$

so that $q_n(cut\eta_0)\sqsupseteq cut(q_n\eta_0)$. When $q_n\eta_0=\bot$ we have $q_n(cut\eta_0)=\bot$ also, whilst when $q_n\eta_0\sqsupset\bot$ we know that $\eta_0\sqsupseteq p_n(cut(q_n\eta_0))\sqsupset\bot$ and that $q_n\eta_0\sqsupseteq cut(q_n\eta_0)$. Hence for every n $q_n\circ cut=cut\circ q_n$, and in particular

when $cut\eta=\bot$ $cut(q_n\eta)=\bot$, so that as B_n is slit for every $n\geq 0$ η is such that $q_n\eta=\bot$.

When $Y\subseteq B$ is directed and $\bigsqcup Y=\top$, $q_0(\bigsqcup Y)=\top$ and $q_0\eta=\top$ for some $\eta\in Y$, since $\top\prec\top$ in B_0. If $q_n\eta=\top$ for this η, $j_n(q_{n+1}\eta)=\top$ so $q_{n+1}\eta=\top$ and we may deduce that $\eta=\top$ and that $\top\prec\top$ in B.⊳

Observe also that if for each $n\geq 0$ k_n is a projection of B_n on to some other slit continuous lattice and if $k_n\eta=\top$ only if $\eta=\top$ while $k_n(cut\eta)=\bot$ only if $k_n\eta=\bot$ then $\bigsqcup k_n\circ q_n$ is a projection of B on to this lattice having $\bigsqcup k_n(q_n\eta)=\top$ only if $\eta=\top$ and $\bigsqcup k_n(q_n(cut\eta))=\bot$ only if $\bigsqcup k_n(q_n\eta)=\bot$.

1.2.8. Methods of combining lattices.

For future reference some methods of constructing domains will now be described. In the course of this description A and B will be taken to be lattices (which may be subject to certain constraints), and q':A→A and q'':B→B will be any continuous mappings of A into A and of B into B having $q'=q'\circ q'\sqsubseteq\lambda\xi.\xi$ and $q''=q''\circ q''\sqsubseteq\lambda\eta.\eta$. Henceforth functions satisfying the conditions imposed on A and B will be termed 'projections' of A and B, but they are actually projections on to q'A and onto q''B having natural embeddings as the corresponding injections.

The set of continuous mappings of A into B, A→B, is itself a lattice on which can be defined a projection $q'\to q''$ by setting $q'\to q''=\lambda\phi.q''\circ\phi\circ q'$; furthermore if A and B are continuous lattices so is A→B. Of less obvious significance is A↛B, which comprises those functions ϕ in A→B, such that, unless $\phi\top=\bot$ or $\phi\bot=\top$, $\phi\xi$ is proper if and only if ξ is proper; $q'\nrightarrow q''$ is taken to be $\lambda\phi.\bigwedge\{q''(\phi(q'\xi)):B\,|\,\xi:A\}\to(\lambda\xi.\xi:A\to q''(\phi(q'\xi)),\top),\top$. By 1.2.3 when A is a flat lattice and B is a slit continuous lattice A↛B is a slit continuous lattice, and by 1.2.5 $q'\nrightarrow q''$ is a projection when

$q'=\lambda\xi.\xi$, $q''\tau$ is improper and for every η:B $q''(cut\eta)=\bot$ only if $q''\eta=\bot$.

The product of A and B, A×B, is not composed of the set of all pairs $\langle\xi,\eta\rangle$ with $\xi\in$A and $\eta\in$B; rather it is formed from the set of those $\langle\xi,\eta\rangle$ such that ξ and η are both proper, together with $\langle\bot,\bot\rangle$ and $\langle\tau,\tau\rangle$, and $q'\times q''$ is $\lambda\langle\xi,\eta\rangle.q'\xi$:A$\wedge q''\eta$:B$\rightarrow\langle q'\xi,q''\eta\rangle,\tau$. Nonetheless when A and B are continuous lattices having $\tau\prec\tau$ A×B is also continuous, whilst when A and B are slit continuous lattices A×B is a slit continuous lattice. In the latter situation the image of A×B under $q'\times q''$ is a slit continuous lattice, provided that $q'\tau$ and $q''\tau$ are improper, $q'(cut\xi)=\bot$ only if $q'\xi=\bot$, and $q''(cut\eta)=\bot$ only if $q''\eta=\bot$.

As intimated in 1.2.2, the sum of A and B, A+B, is the set $\{\xi|\xi:A\}\cup\{\eta|\eta:B\}\cup\{\bot,\tau\}$ under the ordering induced by amalgamating the orderings of A and B; thus when ζ_1:A+B and ζ_2:A+B $\zeta_1\sqsubseteq\zeta_2$ only if either ζ_1 and ζ_2 belong to A and $\zeta_1\sqsubseteq\zeta_2$ in A or ζ_1 and ζ_2 belong to B and $\zeta_1\sqsubseteq\zeta_2$ in B, and if B contains at most two elements A+B is isomorphic with A. The sum of two continuous lattices having $\tau\prec\tau$ is itself a continuous lattice, and the sum of two slit continuous lattices is a slit continuous lattice. Defining $q'+q''$ to be $\lambda\zeta.\zeta$:A$\rightarrow q'\zeta,\zeta$:B$\rightarrow q''\zeta,\tau$ gives a projection such that when A+B is a slit continuous lattice its image under $q'+q''$ is also slit and continuous so long as $q'\tau$ and $q''\tau$ are improper, $q'(cut\xi)=\bot$ only if $q'\xi=\bot$ and $q''(cut\eta)=\bot$ only if $q''\eta=\bot$. For any $\zeta\in$A+B the test ζ:A is deemed to result in *true* *false*, $\bot$ or τ, depending on whether ζ is a proper member of A, a proper member of B, $\bot$ or τ; in addition $\zeta|$A is ζ (regarded as a member of A) if ζ is a proper member of A, τ if ξ is τ and $\bot$ otherwise. Similar remarks apply to ζ:B and $\zeta|$B. The mapping which coerces an element of A or B into the corresponding element of A+B will be omitted from many equations; indeed even the

projections of A+B into A and B will often be left out, so that, if T is a summand of a domain E say, and if ε is a variable ranging over members of E, $\lambda\varepsilon.(\varepsilon|T)\rightarrow\theta_1,\theta_2$ will be written as $\lambda\varepsilon.\varepsilon\rightarrow\theta_1,\theta_2$. By the same token $q'+q''$ may be transmuted into $q'\sqcup q''$.

To form a product of A and B which separates $\langle\perp,\eta\rangle$ from $\perp$ when η is proper, or to form a sum of A and B which separates $\perp$ in A from $\perp$ in the sum, use can be made of A°. This is formed by adjoining new elements $\perp^\circ$ and τ° to A in such a way that when $\xi_1\in A$ and $\xi_2\in A$ $\perp^\circ=\xi_1\sqsubseteq\xi_2=\tau^\circ$ if and only if $\xi_1\sqsubseteq\xi_2$ in A. Thus $\xi_0\in A$ precisely when $\xi_0:A^\circ$, and a singleton $\{dummy\}$ gives rise to $\{dummy\}^\circ$, a lattice containing precisely three elements. The projection $q'^\circ:A^\circ\rightarrow A^\circ$ is the unique function such that $q'^\circ\perp^\circ=\perp^\circ$, $q'^\circ\tau^\circ=\tau^\circ$ and $q'^\circ\xi=q'\xi$ whenever $\xi\in A$.

Given lattices $A_1,\ldots,A_n$ it is possible to set up the product $A_1\times\ldots\times A_n$ and the sum $A_1+\ldots+A_n$ by analogy with A×B and A+B. When $\langle\xi_1,\ldots,\xi_n\rangle\in A_1\times\ldots\times A_n$ its components can be selected by letting $\langle\xi_1,\ldots,\xi_n\rangle\downarrow m=(1\leq m\leq n\rightarrow\xi_m,\tau)$, and $\langle\xi_1,\ldots,\xi_n\rangle\dagger m=(m+1\leq n\rightarrow\langle\xi_{(0\wedge m)+1},\ldots,\xi_n\rangle,\langle\rangle)$. For the lattice $\{\langle\rangle\}^\circ$, which consists of the vector of dimension 0 together with $\perp^\circ$ and τ°, $\langle\rangle\downarrow m=\tau$ and $\langle\rangle\dagger m=\langle\rangle$ for every proper m. Moreover if $\langle\xi_{n+1},\ldots,\xi_{n+m}\rangle\in A_{n+1}\times\ldots\times A_{n+m}$ the element $\langle\xi_1,\ldots,\xi_n\rangle\,\S\,\langle\xi_{n+1},\ldots,\xi_{n+m}\rangle$ of $A_1\times\ldots\times A_n\times A_{n+1}\times\ldots\times A_{n+m}$ is taken to be $\langle\xi_1,\ldots,\xi_n,\xi_{n+1},\ldots,\xi_{n+m}\rangle$; an obvious comparable convention covers the vector of dimension 0. The continuous operations of concatenation, §, slicing, †, selection, ↓, and function application will be given increasing degrees of binding power.

The lattice of finite lists of elements of A, A*, can therefore be defined to be $\{\langle\rangle\}^\circ+A+A\times A+A\times A\times A+\ldots$, which is slit and continuous when A is slit and continuous. Should ξ signify a typical member of A, ξ* will be a typical member of A* having length #ξ*, where $\#\xi^*=(\xi^*:\{\langle\rangle\}^\circ\rightarrow 0,1+\#(\xi^*\dagger 1))$. Concatenation,

slicing and selection will be applied to lists without any mention of the mappings between A* and the relevant summands; # will have lower precedence than these operations, so #ε*↑1 will be written instead of #(ξ*↑1). The predicate ξ:ξ* means $\bigvee\{1\leq\nu\leq\#\xi^*\rightarrow(\xi=\xi^*\downarrow\nu),\mathit{false}\mid\nu:N\}$, where N is the flat lattice of integers. Finally, the projection induced by q' on A*, q'^*, is $\lambda\xi^*.\xi^*:\{\langle\rangle\}^\circ\rightarrow\xi^*,\langle q'(\xi^*\downarrow1)\rangle\S q'^*(\xi^*\uparrow1)$. Although ξ* will not mean $\langle\xi,\ldots,\xi\rangle$, dummy^* will be understood to be $\langle\mathit{dummy},\ldots,\mathit{dummy}\rangle$, and a similar convention will govern 0*, 1*, 2* and 3*.

In 3.5.3 lists which are infinite in length will be required. Accordingly when A is an arbitrary slit continuous lattice A^∞ will be taken to be the least solution of the domain equation $A^\infty=A\times A^{\infty\circ}$ and A^ν will be a solution of $A^\nu=\{\langle\rangle\}^\circ+A\times A^\nu$; the elements of A^∞ are necessarily of infinite length, but some members of A^ν are finite. The nature of A^∞ illustrates one feature of our definition of A×B which would not arise were $\langle\xi,\perp\rangle$ not identified with ⊥; to ensure that certain domains subject to recursive equations are not trivial it is necessary to 'seed' the equation by using $A\times B^\circ$, say, rather than A×B. Though it would be stupid to make A^∞ satisfy the equation $A^\infty=A\times A^\infty$, A^ν can be presumed to satisfy $A^\nu=\{\langle\rangle\}^\circ+A\times A^\nu$ with impunity, since the presence of a sum of domains avoids the need to suppose that $A^\nu=\{\langle\rangle\}^\circ+A\times A^{\nu\circ}$; a similar phenomenon occurs when A↛B is adopted instead of A→B.

If B_1, say, is an arbitrary slit continuous mapping and $j_0:B_1\rightarrow B_0$ is a projection of B_1 on to a lattice B_0 then B_0 is slit and continuous provided that for all $\eta\in B_1$ $j_0\eta=\tau$ only if $\eta=\tau$ and $j_0(cut\eta)=\perp$ only if $j_0\eta=\perp$. The slit continuous lattices are therefore the objects of a category having as their morphisms the projections which satisfy the additional conditions imposed on j_0 in the preceding sentence. This category is closed under

the formation of products, sums and inverse limits; moreover if A and B are slit continuous lattices then $[A\to B]^\circ$ is slit and continuous, while if A is flat and B is slit and continuous then $A\to B$ is slit and continuous. Except in 2.4.4. the domains to be considered below can be obtained by subjecting objects in this category to combinations of the functors discussed above, so they can be assumed to be projections of a universal domain into itself.

Any functor which takes the category of slit continuous lattices into itself and which is finitely generated from the functors described above can be viewed as a transformation acting on projections of the lattices as follows. Let $\mathfrak{A}$ and $\mathfrak{B}$ be such functors for which any projection q of a slit continuous lattice V such that $q\top$ is improper and $q(cut\beta)=\bot$ only if $q\beta=\bot$ gives rise to projections $\mathfrak{A}q:\mathfrak{A}V\to\mathfrak{A}V$ and $\mathfrak{B}q:\mathfrak{B}V\to\mathfrak{B}V$ for which $\mathfrak{A}q\top$ and $\mathfrak{B}q\top$ are improper, $\mathfrak{A}q(cut\xi)=\bot$ only if $\mathfrak{A}q\xi=\bot$ and $\mathfrak{B}q(cut\eta)=\bot$ only if $\mathfrak{B}q\eta=\bot$; suppose that $\mathfrak{A}(q'\circ q'')=\mathfrak{A}q'\circ\mathfrak{A}q''\sqsubseteq\mathfrak{A}(\lambda\beta.\beta)=\lambda\xi.\xi$ and $\mathfrak{B}(q'\circ q'')=\mathfrak{B}q'\circ\mathfrak{B}q''\sqsubseteq\mathfrak{B}(\lambda\beta.\beta)=\lambda\eta.\eta$ for all projections $q':V\to V$ and $q'':V\to V$ satisfying the constraints imposed on q. When $\mathfrak{C}V$ is V, $\mathfrak{A}V\times\mathfrak{B}V$, $\mathfrak{A}V+\mathfrak{B}V$, $\mathfrak{A}V^\circ$, $\mathfrak{A}V^*$ or a lattice independent of V for any projection $q:V\to V$ $\mathfrak{C}q$ will automatically be taken to be q, $\mathfrak{A}q\times\mathfrak{B}q$, $\mathfrak{A}q+\mathfrak{B}q$, $\mathfrak{A}q^\circ$, $\mathfrak{A}q^*$ or $\lambda\theta.\theta$ respectively; when $\mathfrak{C}V$ is $\mathfrak{A}V\to\mathfrak{B}V$ $\mathfrak{C}q$ will be $\mathfrak{A}q\to\mathfrak{B}q$ so long as $\mathfrak{A}V$ is flat and $\mathfrak{A}q=\lambda\xi.\xi$. In all these cases $\mathfrak{C}V$ is a slit continuous lattice, since $\mathfrak{A}V$ and $\mathfrak{B}V$ are slit and continuous; moreover for all the appropriate projections of V $\mathfrak{C}q\top$ is improper, $\mathfrak{C}q(cut\theta)=\bot$ only if $\mathfrak{C}q\theta=\bot$ and $\mathfrak{C}(q'\circ q'')=\mathfrak{C}q'\circ\mathfrak{C}q''\sqsubseteq\mathfrak{C}(\lambda\beta.\beta)=\lambda\theta.\theta$. When $\mathfrak{A}V$ and $\mathfrak{B}V$ are only known to be continuous, writing $\mathfrak{C}V$ for $\mathfrak{A}V\to\mathfrak{B}V$ and $\mathfrak{C}q$ for $\mathfrak{A}q\to\mathfrak{B}q$ gives a continuous lattice $\mathfrak{C}V$ having $\mathfrak{C}(q'\circ q'')=\mathfrak{C}q'\circ\mathfrak{C}q''\sqsubseteq\mathfrak{C}(\lambda\beta.\beta)=\lambda\theta.\theta$; in addition $\mathfrak{C}V^\circ$ is slit and $\mathfrak{C}q^\circ$ (which is not $\mathfrak{C}(q^\circ)$ but $(\mathfrak{C}q)^\circ$) satisfies conditions analogous to those imposed on q.

1.3. The initial paradigm.

1.3.1. Inverse limit spaces.

Semantic equations can be based merely on the existence of reflexive domains and on mappings which preserve the joins of countable chains (in the sense suggested in 1.1.2). Equivalences between these equations, however, frequently also require that the domains be 'small enough'; certain lattices will therefore be viewed as the least fixed points of functions between projections, although explicit inverse limits will not be set up until 2.4.4. Accordingly the limiting process will be embodied in an assertion about the effect of applying *fix* to a particular function; in a formal system this assertion could be regarded as an induction rule.

If V is a putative space of stored values the corresponding domain of stores, $\mathfrak{S}V$, can be constructed from it and a few flat lattices like L and T; $\mathfrak{S}V$ might satisfy $\mathfrak{S}V=[L\to[T\times V]]\times V^*\times V^*$, for instance. The space comprising store transformations, $\mathfrak{C}V$, might be $\mathfrak{S}V\to\mathfrak{A}V$, where $\mathfrak{A}V$ is some other lattice which depends on V; whether or not a computation terminates is influenced by the store supplied as an argument, so $\mathfrak{S}V\to\mathfrak{A}V$ could not be used as $\mathfrak{C}V$. Most languages need further domains, such as those containing expressed values, $\mathfrak{E}V$, and denoted values, $\mathfrak{D}V$. From these domains can be built a new space of stored values, $\mathfrak{V}V$; for Pal this is $B+L^*+\mathfrak{J}V+\mathfrak{f}V$, in which $\mathfrak{J}V$ represents label entry points, $\mathfrak{f}V$ represents functions and B is a fixed flat lattice.

In accordance with 1.2.8, slit continuous lattices like V are the objects of a category having as morphisms the projections such as q which have $q\beta=\top$ only if $\beta=\top$ and $q(cut\beta)=\bot$ only if $q\beta=\bot$. Moreover, the other lattices are built from V in such a way that $\mathfrak{C}V$ and $\mathfrak{A}V$ are continuous while $\mathfrak{S}V$, $\mathfrak{E}V$ and $\mathfrak{D}V$ are both continuous and slit. In fact $\mathfrak{S}$, $\mathfrak{C}$ and $\mathfrak{D}$ are functors taking the category of slit continuous lattices into itself; since this implies that $\mathfrak{I}$ and $\mathfrak{f}$

are functors defined on the same category, in Pal $\mathfrak{V}$ is a functor such that $\mathfrak{V}q=q_0+\mathfrak{J}q+\mathfrak{J}q$, where q_0 is the identity function on B+L*.

When there is a natural isomorphism between V and $\mathfrak{V}$V they can effectively be equated and q_0 can be regarded as a projection of V into itself. Under the conventions of 1.2.8 for any projection q of V into itself for which $q\tau$ is improper and $q\beta=\perp$ whenever $q(out\beta)=\perp$ it is possible to set up a projection $\mathfrak{V}q$ of V into itself having $\mathfrak{V}q=q_0+\mathfrak{J}q+\mathfrak{J}q$; plainly $\mathfrak{V}q\tau$ is improper and $\mathfrak{V}q\beta=\perp$ whenever $\mathfrak{V}q(out\beta)=\perp$. Hence if $q_{n+1}=\mathfrak{V}q_n$ for all $n\geq 0$ then $q_{n+1}\sqsupseteq q_n$ and the minimal nature of V is expressed by the equality $\lambda\beta.\beta=\bigsqcup q_n$; by an elementary calculation this equality holds if and only if $\lambda\beta.\beta=fix(\mathfrak{V})$. For a few little languages the space of stored values reduces to B while the other domains continue to be reflexive, so different functors must be used in formulating the induction rule; one such is $\mathfrak{W}$, which will be mentioned in 2.4.2.

These remarks will now be elucidated by discussing the semantics of a computer language. The paradigm will be a syntactic variant of Pal [5] because its stored label entry points and functions reduce the scope for simple proofs about programs. Since only locations can be denoted even while $x>0$ do $x:=0$ and l: if $x>0$ then $x:=0$; goto l else dummy do not compute the same function according to the semantic equations, although 2.4.5 will present a notion of equivalence which is appropriate to them.

As hinted above, the space of stored values for Pal is built from a fixed flat lattice B, the form of which is relevant to the semantics of the language but not to the theorems to be proved below. In fact B={*dummy*}°+T+N+R+H*, where T={*true*}°+{*false*}°, N represents integers, R represents real numbers and H represents characters (so that H* is the domain of strings). The lattice of real numbers is a flat one, because that suggested by interval analysis is not slit and yields a discontinuous equality predicate.

Taking J to be a lattice of label entry points and F to be one of functions, V=B+L*+J+F, S=[L→[T×V]]×V*×V* and C=S→A. The additional components of S provide rudimentary input and output facilities which are not specified by the original language definition but which are intended to illustrate how real facilities would influence equivalence results; their use will be justified in 3.5.4. The semantic equations will presume nothing about the final domain A to which transformations map stores, but often it can be taken to be a retraction of S encapsulating the output.

Certain primitive functions will be common to all the kinds of semantics which will be described. They include

$empty=(\lambda\alpha.(false,dummy),(),())$;

$area=\lambda\alpha\sigma.(\sigma\downarrow 1)\alpha\downarrow 1$;

$hold=\lambda\alpha\sigma.(\sigma\downarrow 1)\alpha\downarrow 2$;

$update=\lambda\alpha\beta\sigma.\alpha:L\wedge\beta:V\wedge\sigma:S\rightarrow(\lambda\alpha'.\alpha'=\alpha\rightarrow(true,\beta),(\sigma\downarrow 1)\alpha')\,\S\,\sigma\dagger 1,\top$.

Thus every location is deemed to contain a value, but whether or not that value can be obtained when the location is not in the region of store currently in use depends on the language concerned. In 2.5.9 it will be shown that no such location is ever handled by a Pal program as the implementation cannot delete accessible storage.

A sequence of store accesses is performed by means of

$holds=\lambda\alpha^*\sigma.\alpha^*=()\rightarrow(),(hold(\alpha^*\downarrow 1)\sigma)\,\S\,holds(\alpha^*\dagger 1)\sigma$;

$updates=\lambda\alpha^*\beta^*\sigma.\alpha^*=()\rightarrow\sigma,update(\alpha^*\downarrow 1)(\beta^*\downarrow 1)(updates(\alpha^*\dagger 1)(\beta^*\dagger 1)\sigma)$.

Because S now contains few superfluous elements $new:S\rightarrow L$ may be taken to be any continuous function satisfying

$\lambda\sigma.area(new\sigma)\sigma=\lambda\sigma.\bigwedge\{area\alpha\sigma\mid\alpha:L\}\rightarrow\perp,false$.

As $\prod\{\beta\mid\beta:J\}=\perp$ and $\prod\{\beta\mid\beta:F\}=\perp$, when σ is proper the magnitude of any label entry point or function contained in it cannot influence the value of $new\sigma$.

This operation can also be iterated, giving $news:N\rightarrow S\rightarrow L^*$:

$news=\lambda\nu\sigma.\nu=0\rightarrow(),(\lambda\alpha.(\alpha)\,\S\,news(\nu-1)(update\alpha(hold\alpha\sigma)\sigma))(new\sigma)$.

Allowing *update* to adjoin locations to the area does not conform with computing practice, but it does economize on definitions of basic functions.

Another possible model for storage is $[L\to[\{flag\}^\circ+V]]\times V^*\times V^*$, where $\sigma\alpha=flag$ indicates that α is not in the area of store in use whilst $\sigma\alpha:V$ establishes that α is in this area and contains $\sigma\alpha$. Although this model provides the store transformations with precisely those parts of the store on which they truly depend, it is not physically realistic. Locations outside the area of accessible storage have contents which may affect the choice of new locations in a way which this model cannot reflect. Even if they do not influence that choice there remains the question of what new locations should contain when the store is extended. Since the methods to be introduced in 2.4.5 can be adapted to verify that in the earlier model the result of a computation is independent of the contents of locations outside the store area, the artifice described above is not needed to mirror reality well.

1.3.2. The structure of the environment.

The environment supplied as an argument to the semantic equations associates with every identifier all the values it has been made to denote 'up till the current program point'. Though it will be confirmed in 1.5.2 that the meaning of a Pal program has to depend on the values given to its free variables at their most recent declarations only, the entire environment will be necessary in later kinds of equation and will therefore be used here for consistency. For reasons which will be explained in 2.1.6 identifiers will be permitted to denote $\perp$ in an environment which is not $\perp$; consequently when Ide is the flat lattice of identifiers and D is the slit lattice of denoted values the relevant component of U, the domain of environments, will be taken to be $Ide\to D^\circ*$.

The scope of the return link associated with res depends upon the program text, so the environment must include the list of links set up on entry to val blocks. Since these links take both an expressed value from E and a store from S as arguments they belong to K=E→C. Consequently res can sometimes denote the continuation ⊥, which represents a non-terminating program, and, as the coalesced product is being used, the lattice K°*, not K*, must cope with val in U.

To assist with the detection of errors in programs Pal demands that when a new denotation is bound into an environment the erstwhile height of the environment be kept. This requirement will be ignored, however, because it entails only superficial changes in the definition of ρ[δ/I]. The domain of environments will therefore be given by U=[Ide→D°*]×K°*; under a convenient abuse of notation, if ρ is an environment and I is an identifier ρ⟦I⟧ and ρ⟦res⟧ will signify the entities more correctly represented by (ρ↓1)⟦I⟧ and ρ↓2 respectively. In order to extend the environment with δ:D° or κ:K° it is appropriate to set ρ[δ/I]=⟨(λI'.I'=I→⟨δ⟩§ρ⟦I⟧,ρ⟦I'⟧),ρ⟦res⟧⟩ and ρ[κ/res]=⟨λI.ρ⟦I⟧,⟨κ⟩§ρ⟦res⟧⟩. Furthermore, if δ*:D°* and I*:Ide* ρ[δ*/I*]=(I*=⟨⟩→ρ,(ρ[δ*†1/I*†1])[δ*↓1/I*↓1]); all references to ρ[*dummy**/I*] or *updates*α**dummy**σ (such as those to be given in 1.4.4) will tacitly assume that #*dummy** is #I* or #α*.

In 2.1.5 several functions for amalgamating and segregating environments will be needed, but here it suffices to introduce

arid=⟨λI.⟨⟩,⟨⟩⟩;

divert=$\lambda\rho_0\rho_1$.⟨λI.ρ_1⟦I⟧§ρ_0⟦I⟧,ρ_1⟦res⟧§ρ_0⟦res⟧⟩;

invert=$\lambda\rho_0\rho_1$.⟨λI.ρ_1⟦I⟧§(ρ_0⟦I⟧†#ρ_1⟦I⟧),ρ_1⟦res⟧§(ρ_0⟦res⟧†#ρ_1⟦res⟧)⟩;

revert=$\lambda\rho_0\rho_1$.⟨λI.ρ_1⟦I⟧†(#ρ_1⟦I⟧-#ρ_0⟦I⟧),ρ_1⟦res⟧†(#ρ_1⟦res⟧-#ρ_0⟦res⟧)⟩

conserve=λρ*.ρ*=⟨⟩→*arid*,*divert* (ρ*↓1)(*conserve* (ρ*†1)).

Declarations give environments as results so the corresponding semantic equations must be supplied with continuations taken from X=U→C. One pecularity of Pal is that an identifier, x say, may be given a meaning local to a declaration rather than an expression by a form such as x=0 within y=x. To prevent the value given to x from being entered in the environment returned by the complete text the result of a declaration is arranged to contain only those identifiers set up in it rather than the current environment. Thus if χ:X and α:L the declaration above might give rise to the store transformation $\chi(arid[\alpha/y])$.

1.3.3. Value domains.

An expression in a program supplies an argument drawn from a domain comprising the possible outcomes of expressions [23] to the next instruction. This domain of expressed values, E, is L+B+L*+J+F for Pal whereas the domain of denoted values, D, is simply L. In 2.1.6 we shall also be concerned with the domain of witnessed values, W, where ω:W if ω can potentially appear in the store, in the environment or as the answer given by an expression. Because the environment has a simple structure the Pal version of W is L+B+L*+J+F+K° while, anticipating 1.3.4, $\mathscr{I}[\![I]\!]$ is $\lambda\rho\kappa.\kappa(\rho[\![I]\!]\downarrow 1)$.

A label entry point is a store transformation taking only σ:S as an argument. The entry point $\perp$, however, corresponds to a non-terminating computation and on intuitive grounds should not be identified with the stored value $\perp$, which is weaker than some members of B in the lattice ordering. Similarly should τ in C correspond to an erroneous computation it must not be regarded as τ in V. Indeed making such identifications in our model for storage might cause a collapse of the store when an assignment was made to a label variable, and so we must take J to be C° rather than C.

An abstraction takes as its arguments a member of one domain of values and a return link, to which is supplied a member of another domain of values 'on completing the execution of the abstraction'. In Pal both these value domains are L, so the lattice of functions is L→[L→C]→C; however more use will be made of E→[E→C]→C, which allows for the discussion of language features such as those to be introduced in 1.4.5. Considerations of non-termination apply to functions as well as to label entry points, so in standard semantics F will be taken to be [E→K→C]°.

Later a lattice of procedures without parameters, G, will be needed; the structure adopted for this will be [K→C]°. Typical elements of J, F and G will be denoted by θ, ϕ and γ respectively; should they (or certain other) Greek minuscules appear without any explicit mention of the domains concerned they will signify the ones given in the relevant appendix. Because the episema cannot be typed some of the remaining letters will have to be used for more than one purpose, but the context will suffice to remove the ambiguity. Variables ranging over syntactic domains like Exp, the lattice of expressions, will be represented by Greek majuscules. The functors which construct lattices from V (and later from W) will be designated by Light Old English equivalents of the names of the lattices themselves, which will appear without serifs.

1.3.4. Evaluating continuations.

The semantic equations for Pal rely on the 'continuations' of Wadsworth [25], and their main novelty is the introduction of auxiliary valuations to extend the scopes of labels and mutually recursive declarations beyond their textual positions. The idea underlying such valuations will be explained here and applied to the semantics of Mal, of which Pal is a subset.

In many languages labels can be set by colon only when they occur in such a sequence as $I_1:\Gamma_1;\ldots;I_n:\Gamma_n$ where for $1\leq m\leq n$ I_m:Ide and Γ_m:Com (the lattice of annotated derivation trees for commands). Moreover the scope of a label is generally arranged to be the sequence in which it is declared. Under these circumstances there is one semantic equation which applies $\mathscr{C}$ to a command sequence and yields its effect as a labelled block. Labels in Pal, however, can occur anywhere in an expression and their scopes propagate beyond the expression when it is an arm of a conditional clause or the body of a loop. In principle there is no reason why a jump cannot be made into an arithmetic expression or the premise of a conditional clause without assigning a local label to a variable of greater scope; the arbitrary decision in Pal to let scopes propagate only through expressions of the above forms and sequences of expressions is therefore made manifest in our formalism. We introduce auxiliary valuations $\mathscr{J}$:Exp→Ide* and $\mathscr{P}$:Exp→U→K→J*, which collect up the labels and the corresponding entry points of an expression embedded in a particular environment and followed by a known continuation. The valuation $\mathscr{D}$:Exp→U→K→C provides the effect of an expression ignoring label declarations and is applied when the labels set within the expression may have scopes exceeding it, whereas $\mathscr{E}$:Exp→U→K→C regards its argument as a block which confines label scopes. For any E:Exp which is not a label setting, a conditional clause, a loop or a sequence the lists of labels and entry points are empty so $\mathscr{E}[\![E]\!]=\mathscr{D}[\![E]\!]$; in such cases $\mathscr{J}[\![E]\!]$ and $\mathscr{P}[\![E]\!]$ will generally be omitted from the equations in the appendices.

Mutually recursive declarations involve a similar situation, in that the variables being declared are in scope throughout all the declarations and the syntax governing Dec, the domain of declarations, is too complex to permit one semantic equation

to cover all the cases. Thus besides $\mathscr{D}$:Dec→U→X→C we require valuations $\mathscr{I}$:Dec→Ide* and $\mathscr{T}$:Dec→U→X→C, one to assemble the identifiers being declared and the other to evaluate the declarations when these identifiers are already in scope. Thus in Pal if Δ:Dec we take $\mathscr{D}[\![\text{rec } \Delta]\!]$ to be

$\lambda\rho\chi\sigma.(\lambda\alpha^*.\alpha^*:E\rightarrow\mathscr{T}[\![\Delta]\!]\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]]\chi(updates\alpha^*dummy^*\sigma),\tau)(news(\#\mathscr{I}[\![\Delta]\!])\sigma)$,

where the list of locations is tested merely to ensure that when the free store is exhausted execution does not carry on. In fact Pal tends to vacillate between two possible values for $\mathscr{T}[\![\text{I=E}]\!]$ which in terms of the functions to be described in 1.3.5 may be written as

$\lambda\rho\chi.\mathscr{R}[\![E]\!]\rho(\lambda\beta.\chi(arid[\rho[\![I]\!]\downarrow 1/I])\circ update(\rho[\![I]\!]\downarrow 1)\beta)$ and

$\lambda\rho\chi.\mathscr{R}[\![E]\!]\rho(\lambda\beta.lv(\lambda\alpha.\chi(arid[\alpha/I]))\beta\circ update(\rho[\![I]\!]\downarrow 1)\beta)$;

we shall adopt the former, but all that follows, including 2.7.5, remains valid if the latter is used. By contrast $\mathscr{D}[\![\text{I=E}]\!]$ is $\lambda\rho\chi.\mathscr{L}[\![E]\!]\rho(\lambda\alpha.\chi(arid[\alpha/I]))$, in which the location ultimately adjoined to the environment is not known on entry to the expression. The Pal equations governing Δ_0 within Δ_1 and Δ_1 and ... and Δ_n are identical in form with those of appendix 1, which are the ones for a language in which D is not merely L.

Certain kinds of expression have meanings before they are supplied with their continuations. In our present case these include the abstractions and the basic constants, which comprise flat lattices Abs and Bas respectively. For the first of these we introduce a valuation $\mathscr{F}$:Abs→U→F such that when Φ:Abs $\mathscr{F}[\![\Phi]\!]$ requires only an environment to make it into a closure; $\mathscr{F}[\![\text{fnI.E}]\!]$, for instance, is $\lambda\rho.\lambda\varepsilon\kappa.\mathscr{L}[\![E]\!]\rho[\varepsilon/I]\kappa$. We describe $\mathscr{B}$:Bas→E simply by stating that if N, P, and H are variables ranging over numerals, decimals and external representations of characters then $\mathscr{B}[\![N]\!]$, $\mathscr{B}[\![P]\!]$ and $\mathscr{B}[\![\text{"H}_1\text{...H}_n\text{"}]\!]$ signify appropriate elements in the summands N, R and H* of E. In addition $\mathscr{B}[\![\text{dummy}]\!]=dummy$,

$\mathscr{B}[\![\text{true}]\!]=true$, $\mathscr{B}[\![\text{false}]\!]=false$, $\mathscr{B}[\![\text{nil}]\!]=()\,|\,L^*$ and $\mathscr{B}[\![\text{""}]\!]=()\,|\,H^*$;
we shall take B to designate a typical member of Bas.

Because operators in Pal cannot be declared by the programmer there are valuations $\mathscr{O}$:Mon→E→B and $\mathscr{W}$:Dya→[E×E]→B defined on flat lattices Mon and Dya. Thus we introduce variables O:Mon ranging over the monadic operators of the language and Ω:Dya ranging over its dyadic operators, to which the usual meanings will be ascribed in examples such as 3.1.5. We shall not specify these meanings as they are irrelevant to all our theorems provided that we make the assumption that when ε_1:J and ε_2:J or ε_1:F and ε_2:F $\lambda O.\mathscr{O}[\![O]\!]\varepsilon_1=\lambda O.\mathscr{O}[\![O]\!]\varepsilon_2$, while
$\lambda\Omega\varepsilon.\mathscr{W}[\![\Omega]\!](\varepsilon_1,\varepsilon)=\lambda\Omega\varepsilon.\mathscr{W}[\![\Omega]\!](\varepsilon_2,\varepsilon)$ and $\lambda\Omega\varepsilon.\mathscr{W}[\![\Omega]\!](\varepsilon,\varepsilon_1)=\lambda\Omega\varepsilon.\mathscr{W}[\![\Omega]\!](\varepsilon,\varepsilon_2)$.
These equations are reasonable because in practice any operators taking label entry points or functions as arguments can only achieve non-trivial ends (like having a range with cardinality greater than 1) by using information about the machine representation of elements of E. Hence although such operators can be provided in the formalism of 2.1.1, which splits up closures into their constituents, they are not realistic in standard semantics, and this lack of realism is reflected in their absence from conventional programming languages. More precisely, unless we admit $\perp$ or $\top$ as possible values for $\mathscr{O}[\![O]\!]\varepsilon_1$ and $\mathscr{W}[\![\Omega]\!](\varepsilon_1,\varepsilon_2)$ when ε_1 and ε_2 are in J or F the continuity of the valuations and the orderings of B, C° and $[E\to[K\to C]]^\circ$ entail the assumption above.

1.3.5. Other primitive functions.

To obtain the members of L and V associated with an expressed value and a store we introduce
$lv=\lambda\kappa\varepsilon\sigma.\varepsilon:L\to\kappa\varepsilon\sigma,(\lambda\alpha.\alpha:L\to\kappa\alpha(update\,\alpha\varepsilon\sigma),\top)(new\sigma)$;
$rv=\lambda\kappa\varepsilon\sigma.\varepsilon:L\to(area\,\varepsilon\sigma\to\kappa(hold\varepsilon\sigma)\sigma,\top),\kappa\varepsilon\sigma$.
The continuation κ is supplied to these functions in order to

provide a more satisfactory treatment of errors than could be given merely by sending the arguments of κ to τ. Strictly speaking we should introduce functions which take remedial action when an error occurs, but this would add to our notation without enlarging its conceptual basis or altering the nature of our proofs. Only in a language having on conditions is it necessary to make matters so complex; we are content merely to distinguish faulty computations from those which do not finish by taking τ rather than $\perp$ to be an error stop.

Notice that both here and in 1.3.1 the meanings of the primitive functions are specified precisely, instead of being conveyed by sets of axioms. In some applications the latter approach would be adequate, but to establish 2.3.1, for instance, we must know the outcome of $rv\kappa a\sigma$ even when $area a\sigma = false$. Providing axioms sufficient to cope with the situations we consider is tantamount to defining the primitive functions except at certain improper values. Accordingly it is far less cumbersome to give complete definitions by fixing a few values arbitrarily than it is to postulate properties.

In this connection observe also that the presence of continuations in semantic equations can ensure that $\perp$ and τ are never passed on by one expression to the succeeding one, for failures to terminate correctly give rise to improper elements of A rather than of E or S. Consequently the meanings of the equations need be influenced neither by any particular choice of conditional function nor by whether $update\tau dummy$ and $update\perp dummy$ do or do not commute. A formal proof of this could be given for the semantics of appendix 1 by using the technique outlined in 2.2.7; essentially it would show that the ultimate member of A yielded by an expression applied to a continuation κ could not be affected by the values of $\kappa\perp$ and $\kappa\tau$.

Though the valuation $\mathscr{E}$ suffices to determine the outcome of a Pal expression it is convenient to adopt $\mathscr{L}$:Exp→U→K→C and $\mathscr{R}$:Exp→U→K→C which coerce that outcome into forms appropriate to left-hand and right-hand contexts. Thus for all E:Exp we write $\mathscr{L}[\![E]\!]=\lambda\rho\kappa.\mathscr{E}[\![E]\!]\rho(l\nu\kappa)$ and $\mathscr{R}[\![E]\!]=\lambda\rho\kappa.\mathscr{E}[\![E]\!]\rho(r\nu\kappa)$. Languages in which expressions are subject to a much wider range of contextual coercions require the more elaborate treatment mentioned in 3.6.1.

Often we wish to avoid imposing a particular order of evaluation on a list of expressions or declarations because two implementations may choose different ones. To eliminate pointless restrictions we make use of a function i:N→N→N such that for every $\nu>0$ $i\nu$ is a permutation of $\{1,\ldots,\nu\}$. This function gives rise to j:N→N→N, which is set up in such a way that $j\nu$ is the inverse of $i\nu$ whenever $\nu>0$. For any lattice A we define k:N→A*→A* by $k=\lambda\nu\xi^*.\nu\leq\#\xi^*\to\langle\xi^*\downarrow j\nu 1,\ldots,\xi^*\downarrow j\nu\nu\rangle,\top$; hence if for some lattice B we have ϕ:[[A→B]→B]°* and ψ:A*→B we may set

$$run=\lambda\phi\psi.(\lambda n.(\phi\downarrow i(n)1)(\lambda\xi_1.(\phi\downarrow i(n)2)(\lambda\xi_2.(\phi\downarrow i(n)3)(\lambda\xi_3.$$
$$\ldots(\phi\downarrow i(n)n)(\lambda\xi_n.\psi(k(n)\langle\xi_1,\ldots,\xi_n\rangle))\ldots)))(\#\phi).$$

In particular we require run:[K→C]°*→[E*→C]→C and run:[X→C]°*→[U*→C]→C; a minor modification to their definitions would allow their orders of evaluation to depend upon the store. These functions do not allow us to optimize a program by evaluating only one member of a pair of identical expressions, since run acts upon code, not upon text. To model a compiler which optimized $x1+x1$, say, we would need to introduce additional valuations which would gather up the occurrences of $x1$ and the other expressions; an analogue of run which would apply to portions of text could then be used to evaluate each expression only once. Such a function usually has no place in a formal definition of a language, although it is relevant to the correctness of particular

compilers.

Implementations are judged by the extent to which they measure up to equations having the form of those in appendix 1; indeed it is the normative role of these equations that is responsible for the name 'standard semantics'. Furthermore, from these equations we can derive a class of rules intended to describe particular language constructs [8], but its utility for Pal is limited by the possibility of sharing, the absence of denoted functions and the existence of label variables. To prove programs correct is is frequently necessary to resort to the formal semantics; by this means one can, for instance, validate programs for copying and reversing Pal graphs of locations. The proofs are no more complex than conventional ones dealing solely with acyclic trees [2], but they are boring. More interesting is the following example, which adapts a method due to Park [14] in order to show that, according to our intuitive beliefs about $\perp$, certain programs fail to terminate.

1.3.6. Example.

Let Δ_0 be f=fn$z$.z within f=fn$z$.($f)$z$ and E_0 be rec Δ_0 inside ($f)0; when ρ_0 and σ_0 are proper $\lambda\kappa.\mathscr{E}[\![E_0]\!]\rho_0\kappa\sigma_0$ is $\perp$.

⊲Suppose that ρ_0 and $news4\sigma_0$ are proper, and let $\alpha_0=new\sigma_0$, $\rho_1=\rho_0[\alpha_0/f]$, $\sigma_1=update\alpha_0 dummy\sigma_0$, $\alpha_1=new\sigma_1$, $\sigma_2=update\alpha_1(\mathscr{F}[\![$ fn$z.z]\!]\rho_1)\sigma_1$, $\rho_2=\rho_1[\alpha_1/f]$ and $\sigma_3=update\alpha_1(\mathscr{F}[\![$ fn$z.($\$f)z]\!]\rho_2)\sigma_2$. For any χ

$$\begin{aligned}\mathscr{T}[\![\Delta_0]\!]\rho_1\chi\sigma_1 &= \mathscr{D}[\![f\text{=fn}z.z]\!]\rho_1(\lambda\rho.\mathscr{T}[\![f\text{=fn}z.(\$f)z]\!](divert\rho_1\rho)\chi)\sigma_1\\ &= \mathscr{L}[\![\text{fn}z.z]\!]\rho_1(\lambda\alpha.\mathscr{T}[\![f\text{=fn}z.(\$f)z]\!](divert\rho_1(arid[\alpha/f]))\chi)\sigma_1\\ &= \mathscr{T}[\![f\text{=fn}z.(\$f)z]\!]\rho_2\chi\sigma_2\\ &= \mathscr{R}[\![\text{fn}z.(\$f)z]\!]\rho_2(\lambda\beta.\chi(arid[\alpha_1/f])\circ update\alpha_1\beta)\sigma_2\\ &= \chi(arid[\alpha_1/f])\sigma_3.\end{aligned}$$

Now set $\rho_3=\rho_0[\alpha_1/f]$, $\langle\alpha_2,\alpha_3\rangle=news2\sigma_3$ and $\sigma_4=updates\langle\alpha_2,\alpha_3\rangle\langle hold\alpha_1\sigma_3,0\rangle\sigma_3$. For any κ

$$\begin{aligned}\mathscr{S}[\![E_0]\!]\rho_0\kappa\sigma_0&=\mathscr{D}[\![\text{rec }\Delta_0]\!]\rho_0(\lambda\rho.\mathscr{L}[\![(\$f)0]\!](divert\rho_0\rho)\kappa)\sigma_0\\&=\mathscr{S}[\![\Delta_0]\!]\rho_1(\lambda\rho.\mathscr{L}[\![(\$f)0]\!](divert\rho_0\rho)\kappa)\sigma_1\\&=\mathscr{L}[\![(\$f)0]\!]\rho_3\kappa\sigma_3\\&=\mathscr{S}[\![\text{fn}z.(\$f)z]\!]\rho_2\alpha_3(lv\kappa)\sigma_4\\&=\mathscr{L}[\![(\$f)z]\!]\rho_2[\alpha_3/z](lv\kappa)\sigma_4.\end{aligned}$$

Take σ_5 to be any store such that $hold\alpha_1\sigma_5\sqsubseteq hold\alpha_1\sigma_3$; then, writing α_4 for $new\sigma_5$ and σ_6 for $update\alpha_4(hold\alpha_1\sigma_5)\sigma_5$, $\alpha_4=\top$ and $hold\alpha_1\sigma_6\sqsubseteq hold\alpha_1\sigma_5$. In terms of the projections of 1.3.1, however,

$$\begin{aligned}\mathscr{L}[\![(\$f)z]\!]\rho_2[\alpha_3/z](lv\kappa)(\textcircled{\$}q_{n+1}\sigma_5)&=rv(lv(rv(\lambda\beta.\beta\alpha_3(lv\kappa))))\alpha_1(\textcircled{\$}q_{n+1}\sigma_5)\\&\sqsubseteq(q_{n+1}(hold\alpha_1\sigma_5)\,|\,\mathsf{F})\alpha_3(lv\kappa)(\textcircled{\$}q_{n+1}\sigma_6)\\&\sqsubseteq\mathscr{S}q_n(hold\alpha_1\sigma_5\,|\,\mathsf{F})\alpha_3(lv\kappa)\sigma_6\\&\sqsubseteq(hold\alpha_1\sigma_5\,|\,\mathsf{F})\alpha_3(lv\kappa)(\textcircled{\$}q_n\sigma_6)\\&\sqsubseteq(hold\alpha_1\sigma_3\,|\,\mathsf{F})\alpha_3(lv\kappa)(\textcircled{\$}q_n\sigma_6)\\&=\mathscr{S}[\![\text{fn}z.(\$f)z]\!]\rho_2\alpha_3(lv\kappa)(\textcircled{\$}q_n\sigma_6)\\&=\mathscr{L}[\![(\$f)z]\!]\rho_2[\alpha_3/z](lv\kappa)(\textcircled{\$}q_n\sigma_6),\end{aligned}$$

while $\mathscr{L}[\![(\$f)z]\!]\rho_2[\alpha_3/z](lv\kappa)(\textcircled{\$}q_0\sigma_5)=\bot$. As σ_6 obeys the sole constraint imposed on σ_5, by induction $\mathscr{L}[\![(\$f)z]\!]\rho_2[\alpha_3/z](lv\kappa)(\textcircled{\$}q_n\sigma_5)=\bot$ for all n and in particular $\mathscr{S}[\![E_0]\!]\rho_0\kappa\sigma_0=\mathscr{L}[\![(\$f)z]\!]\rho_2[\alpha_3/z](lv\kappa)(\bigsqcup\textcircled{\$}q_n\sigma_4)=\bot$.⊁

The presence of $\$f$ instead of f in Δ_0 and E_0 is irrelevant to the argument above but will be necessary in 2.7.7, where the elegance of within declarations will be shown to be somewhat meretricious. Notwithstanding the intention that the identifiers declared by Δ_1 in Δ_1 within Δ_2 should have scopes limited to Δ_2 this example demonstrates that this need not happen even in Pal.

The validity of this result does not require the hypothesis that $new\sigma=\bot$ if $\bigwedge\{area\alpha\sigma\,|\,\alpha:\mathsf{L}\}=true$, provided that L is taken to be infinite and all the areas of storage are assumed to be finite.

1.4. Incidence and reference.

1.4.1. Declarations having side effects.

The exotic flavour of Pal owes much to the fact that only locations can be denoted. When designing languages with more efficient implementations and equally elegant semantics it is natural to remove this restriction. This task is not trivial, however: although the semantic equation for stored labels can readily be converted into one for denoted labels which uses a fixed point this is not so for recursive declarations. If the right hand expressions in such declarations were necessarily constants, identifiers or functional abstractions our intuitive view of recursion would be captured by a simple semantic equation, but the arbitrary expressions permitted by Pal may influence (and be influenced by) the store in a way which this would not reflect. Here we seek an equation which models declarations with side effects and which corresponds with Pal recursion in a way to be clarified in 2.7.6. To illuminate the search we use Mal, an enhancement of Pal wherein I==E and $I_1,\ldots,I_n$==E indicate that the identifiers declared thereby signify values which in Pal can be only be stored. In contrast to the declarations by reference of 1.3.4 such declarations by incidence require a large space of denoted values, which tentatively we take to be L+B+L*+J+F; accordingly now we can safely assume $\mathscr{D}[\![\,\text{I==E}\,]\!]$ to be $\lambda\rho\chi.\mathscr{R}[\![\,\text{E}\,]\!]\rho(\lambda\varepsilon.\chi(\mathit{arid}[\varepsilon/\text{I}]))$.

Were the equation for rec I==E to mimic that for rec I=E, probably like the latter it would simplify sensibly when E signifies an abstraction. We must not be led solely by this criterion, however, as rec I=E may give a silly answer. If I is accessed in the body of E and outside an abstraction before being updated, the result may reflect the capricious choice of initial content of the corresponding location; *dummy* is not adopted for this in

the manual [5]. Here the main importance of equating $\mathscr{D}[\![\text{rec } I=E]\!]$ with $\lambda\rho\chi.lv(\lambda\alpha.\mathscr{R}[\![E]\!]\rho[\alpha/I](\lambda\beta.\chi(arid[\alpha/I])\circ update\alpha\beta))(dummy)$ lies in the suggestion that the side effects of E be carried out once, at the time of declaration, instead of whenever I is looked up in the environment or during both declaration and inspection.

If Φ:Abs a natural candidate for the value of $\mathscr{D}[\![\text{rec } I==\Phi]\!]\rho\chi\sigma$ is $\mathscr{R}[\![\Phi]\!]\rho[fix(\lambda\varepsilon.\mathscr{F}[\![\Phi]\!]\rho[\varepsilon/I])/I]\kappa\sigma$ or $\kappa(fix(\lambda\rho'.\rho[\mathscr{F}[\![\Phi]\!]\rho'/I])[\![I]\!]\downarrow 1)\sigma$ where κ is $\lambda\varepsilon.\chi(arid[\varepsilon/I])$. This, however, gives little guidance about $\mathscr{D}[\![\text{rec } I==E]\!]$ even when E is if I_0 then Φ_1 else Φ_2 where what I finally denotes may depend on the store in a way which is not known when the environment above is formed. We cannot give I a random initial value as this will then appear in the environment attached to the resulting closure; rather we should supply it with something reflecting the whole of E. Accordingly we adjoin G=[K→C]° to D (so that D becomes E+G), attempt to use $\mathscr{R}[\![E]\!]\rho[fix(\lambda\delta.\mathscr{R}[\![E]\!]\rho[\delta/I])/I]\kappa\sigma$ as the recursion operator and take $\mathscr{I}[\![I]\!]$ to be $\lambda\rho'\kappa'.\rho'[\![I]\!]\downarrow 1:G\rightarrow(\rho'[\![I]\!]\downarrow 1)\kappa',\kappa'(\rho'[\![I]\!]\downarrow 1)$.

Unfortunately this form of equation does not satisfy the demand that the side effects of E happen exactly once. It can readily be verified that under it

z=1 inside rec *f*==(*z*:=-*z*; fn*x*.if *x*>0 then *f*0 else 0) inside *f*1

leaves 1 in the location denoted by *z* whereas the corresponding program in which *f* denotes a location leaves -1.

To eliminate the side effects during the inspection of *f* in the environment we try out $\mathscr{R}[\![E]\!]\rho[fix(\lambda\gamma\kappa'\sigma'.\mathscr{R}[\![E]\!]\rho[\gamma/I](\lambda\varepsilon\sigma''.\kappa'\varepsilon\sigma')\sigma')/I]\kappa\sigma$. Here whatever effect the procedure denoted by I has on the store at the time of inspection is thrown away, only that part of the result in E

being preserved. Now the outcome of the program above is to leave -1 in the location denoted by z.

Even this operator is inaccurate, however, as the store is not sealed into the procedure denoted by I, so this returns a result which depends on the store at the time of activation (not that at the time of declaration) in much the same way as fluid variables depend on the environment. For instance,

z=1 inside (rec f==if z>0 then fn$x$.if x>0 then f0 else 0 else fn$x$.1
inside (z:=0; f1))

switches from one branch of the conditional clause to the other when the application f1 invokes in turn f0 (with f signifying a member of G). Because the content of the location denoted by z changes between the definition of f and its application, this member of G selects the second arm of the conditional so that the program returns a location containing 1. The Pal version of the program, however, fixes the function assigned to f during the declaration and is not affected by the change thereafter in the value of z. Accordingly it returns a location containing 0 as result.

To seal in the store properly we therefore take the value of $\mathscr{D}[\![$ rec I==E$]\!]\rho\chi\sigma$ to be

$\mathscr{R}[\![E]\!]\rho[fix(\lambda\gamma\kappa'\sigma'.\mathscr{R}[\![E]\!]\rho[\gamma/I](\lambda\varepsilon\sigma''.\kappa'\varepsilon\sigma')\sigma)/I]\kappa\sigma$ or

$\mathscr{R}[\![E]\!](fix(\lambda\rho'.\rho[\lambda\kappa'\sigma'.\mathscr{R}[\![E]\!]\rho'(\lambda\varepsilon\sigma''.\kappa'\varepsilon\sigma')\sigma/I]))\kappa\sigma$ where κ is $\lambda\varepsilon.\chi(arid[\varepsilon/I])$. This deals with the programs above in the same manner as $\mathscr{D}[\![$ rec I=E$]\!]$; yet there remain programs which do not access I in the body of E but for which the semantic equations give different answers. Though below we give some examples of these, they are too pathological to nullify our conviction that the equations are equivalent in all sensible cases. When they disagree the equation for rec I=E is sometimes preferable; rec x=nil aug x typifies this situation.

1.4.2. Example.

Let E_0 be res fn$z$.f, E_1 be (val (rec f==E_0 inside 0))1 and E_2 be (val (rec f=E_0 inside 0))1. When ρ_0 and σ_0 are proper $\mathscr{G}[\![E_1]\!]\rho_0\kappa_0\sigma_0$ is inevitably $\perp$ whereas $\mathscr{G}[\![E_2]\!]\rho_0\kappa_0\sigma_0$ is not.

⊲Set $\alpha_1=new\sigma_0$, $\sigma_1=update\alpha_1 1\sigma_0$, $\alpha_2=new\sigma_1$, $\kappa_1=rv(\lambda\beta\sigma.\beta:\mathsf{F}\to\beta\alpha_1\kappa_0\sigma, 1\le hold\alpha_1\sigma\,|\,\mathsf{N}\le\#\beta\,|\,\mathsf{L}^*\to\kappa_0(\beta\downarrow hold\alpha_1\sigma)\sigma,\tau)$, $fun=\lambda\nu.\nu=0\to\perp,\rho_0[\kappa_1/\mathrm{res}][\lambda\kappa\sigma.\mathscr{R}[\![E_0]\!](fun(\nu-1))(\lambda\varepsilon\sigma'.\kappa\varepsilon\sigma)\sigma_1/f]$, $\rho_1=\bigsqcup\{fun\nu\,|\,\nu:\mathsf{N}\}$ and $\sigma_2=update\alpha_2(\lambda\varepsilon.\mathscr{L}[\![f]\!]\rho_1[\varepsilon/z])\sigma_1$.

$$\begin{aligned}\mathscr{G}[\![E_1]\!]\rho_0\kappa_0\sigma_0&=\mathscr{L}[\![\text{rec } f\text{==}E_0 \text{ inside } 0]\!]\rho_0[\kappa_1/\mathrm{res}]\kappa_1\sigma_1\\&=\mathscr{R}[\![E_0]\!]\rho_1(\lambda\beta.\mathscr{L}[\![0]\!]\rho_0[\kappa_1/\mathrm{res}][\beta/f]\kappa_1)\sigma_1\\&=\mathscr{L}[\![\mathrm{fn}z.f]\!]\rho_1\kappa_1\sigma_1.\end{aligned}$$

For any $\nu\ge 0$ there is some κ_2 such that

$$\begin{aligned}\mathscr{L}[\![\mathrm{fn}z.f]\!](fun(\nu+1))\kappa_1\sigma_1&=lv\kappa_1(\lambda\varepsilon.\mathscr{L}[\![f]\!](fun(\nu+1))[\varepsilon/z])\sigma_1\\&=\mathscr{L}[\![f]\!](fun(\nu+1))[\alpha_1/z]\kappa_0\sigma_2\\&=\mathscr{R}[\![E_0]\!](fun\nu)\kappa_2\sigma_2\\&=\mathscr{L}[\![\mathrm{fn}z.f]\!](fun\nu)\kappa_1\sigma_2\end{aligned}$$

while

$$\begin{aligned}\mathscr{L}[\![\mathrm{fn}z.f]\!]\perp\kappa_1\sigma_1&=lv\kappa_1(\lambda\varepsilon.\mathscr{L}[\![f]\!]\perp[\varepsilon/z])\sigma_1\\&=\mathscr{L}[\![f]\!]\perp[\alpha_1/z]\kappa_0\sigma_2\\&=\perp.\end{aligned}$$

Moreover these conclusions hold for all proper σ_1 and the σ_2 induced by it because if $\lambda\alpha.area\alpha\sigma=true$ then $new\sigma=\perp$; in particular $\mathscr{L}[\![\mathrm{fn}z.f]\!]\rho_1\kappa_1\sigma_1=\perp$ for our original choice of ρ_1 and σ_1.

Contrariwise, defining $\sigma_3=update\alpha_2 dummy\sigma_1$, $\rho_2=\rho_0[\kappa_1/\mathrm{res}][\alpha_2/f]$ and $\phi_0=\lambda\varepsilon.\mathscr{L}[\![f]\!]\rho_2[\varepsilon/z]$ gives

$$\begin{aligned}\mathscr{G}[\![E_2]\!]\rho_0\kappa_0\sigma_0&=\mathscr{L}[\![\text{rec } f\text{=}E_0 \text{ inside } 0]\!]\rho_0[\kappa_1/\mathrm{res}]\kappa_1\sigma_1\\&=\mathscr{R}[\![E_0]\!]\rho_2(\lambda\beta.\mathscr{L}[\![0]\!]\rho_2\kappa_1\circ update\alpha_2\beta)\sigma_3\\&=\mathscr{L}[\![\mathrm{fn}z.f]\!]\rho_2\kappa_1\sigma_3\\&=lv\kappa_1\phi_0\sigma_3\\&=\mathscr{L}[\![f]\!]\rho_2[\alpha_1/z]\kappa_0(update(new\sigma_3)\phi_0\sigma_3)\\&=\kappa_0\alpha_2(update(new\sigma_3)\phi_0\sigma_3).\end{aligned}$$

⊳

To reduce the possibility of jumping out of a declaration into an expression within its scope Pal arranges that in, say, rec g=goto l inside l: g a jump is made not to the l set by colon here but to one in an outer block. Other languages are not so prescient: an Algol 68 version of this fragment, for instance, would cause the same sort of chaos as arose above. On the other hand the context conditions of Algol 68 prohibit a form of 1.4.3 in which the label is assigned to a variable of type ref proc void. Similar prohibitions will also be incorporated in the proof of the equivalence of rec I=E and rec I==E to be given in 2.7.7.

1.4.3. Example.

Let E_0 be l: m:=l; x:=-x; E_3,
E_1 be m,x=1,1 inside rec f==E_0 inside E_4,
E_2 be m,x=1,1 inside rec f=E_0 inside E_4,
E_3 be if x>0 then Φ_0 else Φ_1 and E_4 be if x>0 then fx else goto m,
where Φ_0 is fn$z$.if z>0 then f0 else 0 and Φ_1 is
fn$z$.if z>0 then f0 else 1. Then E_1 and E_2 yield different answers.

Set $\langle\alpha_0,\alpha_1\rangle$=$news2\sigma_0$, σ_1=$updates\langle\alpha_0,\alpha_1\rangle\langle 1,1\rangle\sigma_0$, α_2=$new\sigma_1$ and ρ_1=$\rho_0[\alpha_0/m][\alpha_1/f]$ for some proper ρ_0 and σ_0; let κ_0 be arbitrary.

If the recursion operator used is that introduced in 1.4.1 set $\kappa_1=\lambda\varepsilon.\mathscr{L}[\![E_4]\!]\rho_1[\varepsilon/f]\kappa_0$, $\theta_0=\mathscr{G}[\![E_0]\!]\rho_2[\alpha_2/l](rv\kappa_1)$, $\rho_2=fix(\lambda\rho.\rho_1[\lambda\kappa\sigma.\mathscr{R}[\![E_0]\!]\rho(\lambda\varepsilon\sigma'.\kappa\varepsilon\sigma)\sigma_1/f])$, $\phi_0=\mathscr{F}[\![\Phi_0]\!]\rho_2$, $\phi_1=\mathscr{F}[\![\Phi_1]\!]\rho_2$, $\sigma_2=updates\langle\alpha_0,\alpha_2\rangle\langle\theta_0,\theta_0\rangle\sigma_1$ and $\sigma_3=update\alpha_1(-1)\sigma_2$.

$$\begin{aligned}
\mathscr{G}[\![E_1]\!]\rho_0\kappa_0\sigma_0 &= \mathscr{R}[\![E_0]\!]\rho_2\kappa_1\sigma_1\\
&= \theta_0(update\alpha_2\theta_0\sigma_1)\\
&= \mathscr{G}[\![E_3]\!]\rho_2[\alpha_2/l]\kappa_1\sigma_3\\
&= \mathscr{L}[\![\text{goto } m]\!]\rho_2[\phi_1/f]\kappa_0\sigma_3\\
&= \mathscr{G}[\![E_3]\!]\rho_2[\alpha_2/l]\kappa_1\sigma_2\\
&= \mathscr{L}[\![fx]\!]\rho_1[\phi_0/f]\kappa_0\sigma_2\\
&= \mathscr{L}[\![f0]\!]\rho_2[\alpha_1/z]\kappa_0\sigma_2.
\end{aligned}$$

Define $\kappa_2=\lambda\varepsilon\sigma.lv(\lambda\alpha.\varepsilon\alpha\kappa_0)0\sigma_2$ and $\theta_1=\mathscr{E}[\![E_0]\!]\rho_2[\alpha_2/l](rv\kappa_2)$, for which

$$\begin{aligned}
\mathscr{E}[\![E_1]\!]\rho_0\kappa_0\sigma_0&=\mathscr{L}[\![f0]\!]\rho_2[\alpha_1/z]\kappa_0\sigma_2\\
&=\mathscr{R}[\![E_0]\!]\rho_2\kappa_2\sigma_1\\
&=\theta_1(update\,\alpha_2\theta_1\sigma_1)\\
&=\mathscr{E}[\![E_3]\!]\rho_2[\alpha_2/l]\kappa_2(update\,\alpha_2\theta_1\sigma_3)\\
&=\kappa_2\phi_1(update\,\alpha_2\theta_1\sigma_3)\\
&=lv(\lambda\alpha.\phi_1\alpha\kappa_0)0\sigma_2\\
&=lv\kappa_0 1(update(new\sigma_2)0\sigma_2).
\end{aligned}$$

On the other hand, writing $\sigma_4=update\,\alpha_2\,dummy\sigma_1$, $\alpha_3=new\sigma_4$, $\rho_3=\rho_1[\alpha_2/f]$, $\kappa_3=rv(\mathscr{L}[\![E_4]\!]\rho_3\kappa_0\circ update\,\alpha_2)$, $\phi_2=\mathscr{F}[\![\Phi_0]\!]\rho_3$, $\phi_3=\mathscr{F}[\![\Phi_1]\!]\rho_3$, $\theta_2=\mathscr{E}[\![E_0]\!]\rho_3[\alpha_3/l]\kappa_3$, $\sigma_5=updates\langle\alpha_0,\alpha_2,\alpha_3\rangle\langle\theta_2,\phi_3,\theta_2\rangle\sigma_3$ and $\sigma_6=updates\langle\alpha_1,\alpha_2\rangle\langle 1,\phi_2\rangle\sigma_5$ gives the outcome of the declaration by reference thus:

$$\begin{aligned}
\mathscr{E}[\![E_2]\!]\rho_0\kappa_0\sigma_0&=\mathscr{R}[\![E_0]\!]\rho_3\kappa_3\sigma_4\\
&=\theta_2(update\,\alpha_3\theta_2\sigma_4)\\
&=\mathscr{E}[\![E_3]\!]\rho_3[\alpha_2/l]\kappa_3(updates\langle\alpha_0,\alpha_1,\alpha_3\rangle\langle\theta_2,-1,\theta_2\rangle\sigma_4)\\
&=\kappa_3\phi_3(updates\langle\alpha_0,\alpha_1,\alpha_3\rangle\langle\theta_2,-1,\theta_2\rangle\sigma_4)\\
&=\mathscr{T}[\![\text{goto } m]\!]\rho_3\kappa_0\sigma_5\\
&=\theta_2\sigma_5\\
&=\mathscr{E}[\![E_3]\!]\rho_3[\alpha_3/l]\kappa_3(update\,\alpha_1 1\sigma_5)\\
&=\kappa_3\phi_2(update\,\alpha_1 1\sigma_5)\\
&=\mathscr{T}[\![fx]\!]\rho_3\kappa_0\sigma_6\\
&=\phi_2\alpha_1\kappa_0\sigma_6\\
&=\mathscr{T}[\![f0]\!]\rho_3[\alpha_1/z]\kappa_0\sigma_6\\
&=lv(\lambda\alpha.\phi_2\alpha\kappa_0)0\sigma_6\\
&=lv\kappa_0 0(update(new\sigma_6)0\sigma_6).\blacktriangleright
\end{aligned}$$

Hence under some circumstances the function set up by the assignment method of performing recursion can change irrevocably as a result of jumping back into the declaration, whereas that set up by using a fixed point can switch back and forth between two distinct forms.

1.4.4. Mutual recursion.

The recursion operator of 1.4.1 can be extended to multiple declarations by sealing the store into each of the new denoted values either when the constituent declaration is reached (and the side effects of preceding declarations are accounted for) or when evaluation of the sequence starts. These alternatives give y in rec x=0 and y==x the values 0 and *dummy* respectively; as rec x= 0 and y=x makes y contain 0 it is plain that the former option is correct. Accordingly we still take $\mathscr{T}$:Dec→U→X→C as a valuation on declarations, defining $\mathscr{T}[\![I==E]\!]$ to be $\lambda\rho\chi.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon.\chi(arid[\varepsilon/I]))$. Now, however, $\mathscr{D}[\![$ rec $\Delta]\!]$ is

$\lambda\rho\chi\sigma.(\lambda\alpha^*.(\lambda\sigma'.\alpha^*:E\to\mathscr{T}[\![\Delta]\!](fix(\lambda\rho'.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!][\mathscr{T}[\![\Delta]\!]\rho'\sigma'/\mathscr{K}[\![\Delta]\!]]))\chi\sigma',\tau)$

$(updates\alpha^*dummy^*\sigma))(news(\#\mathscr{I}[\![\Delta]\!])\sigma)$,

where $\mathscr{K}$:Dec→Ide* collects up the identifiers to be given meanings which are in V+G and $\mathscr{S}$:Dec→U→S→G* is such that when $1\le\nu\le\#\mathscr{K}[\![\Delta]\!]$ $\mathscr{S}[\![\Delta]\!]\rho\sigma\downarrow\nu$ is $\lambda\kappa'\sigma'.\mathscr{T}[\![\Delta]\!]\rho(\lambda\rho''\sigma''.\kappa'(\rho''[\![\mathscr{K}[\![\Delta]\!]\downarrow\nu]\!]\downarrow 1\,|\,E)\sigma')\sigma$.

Whereas Pal offers only one possible reading of $\mathscr{T}[\![\Delta_1$ and...and $\Delta_n]\!]$, Mal offers two because the environment returned by a recursive declaration may not be a portion of that which is used for the evaluation. Thus although we could use $\lambda\rho\chi.run\langle\mathscr{T}[\![\Delta_1]\!]\rho,\ldots,\mathscr{T}[\![\Delta_n]\!]\rho\rangle(\chi\circ conserve)$ as the requisite value, we actually adopt

$\lambda\rho\chi.\mathscr{T}[\![\Delta_1]\!]\rho(\lambda\rho_1.\mathscr{T}[\![\Delta_2]\!](divert\rho\rho_1)(\lambda\rho_2.\mathscr{T}[\![\Delta_3]\!](divert\rho(conserve\langle\rho_1,\rho_2\rangle))$

$\ldots(\lambda\rho_n.\chi(conserve\langle\rho_1,\ldots,\rho_n\rangle))\ldots)))$,

(which could be generalized to permit any order of evaluation). The methods of 2.7.6 can be used to show that in fact these alternatives yield essentially similar theories under reasonable circumstances which include the omission of labels and goto statements from declarations.

The proof of 1.5.9 will justify the belief that this analysis of recursion coincides with the view of recursive

function abstraction given in 1.4.1. This result can also be established for the alternative version of $\mathscr{T}[\![\Delta_1$ and...and $\Delta_n]\!]$ (albeit at the expense of extra complexity), so nothing is lost by selecting one equation rather than the other.

1.4.5. Further features.

The concomitant of declaration by incidence is abstraction by incidence: in Ma1 fnI..E and fn$I_1,\ldots,I_n$..E take parameters which are not locations, although fnI.E and fn$I_1,\ldots,I_n$.E inherit the abstraction by reference mechanism of Pa1. Accordingly we extend $\mathscr{F}$:Abs→U→F to the new cases by taking $\mathscr{F}[\![$ fnI..E$]\!]$ to be $\lambda\rho.\lambda\varepsilon\kappa.rv(\lambda\beta.\mathscr{P}[\![E]\!]\rho[\beta/I]\kappa)\varepsilon$ and $\mathscr{F}[\![$ fn$I_1,\ldots,I_n$..E$]\!]$ to be $\lambda\rho.\lambda\varepsilon\kappa.rv(\lambda\beta.\#\beta\,|\,L^*=n\rightarrow\lambda\sigma.\mathscr{P}[\![E]\!]\rho[holds\,\beta\sigma/\langle I_1,\ldots,I_n\rangle]\kappa\sigma,\tau)\varepsilon$. For simplicity we do not introduce calls by incidence but retain our earlier conception of functional application, so that giving fnI..E an expression returning a value in V will cause that value to be copied into a new location and then extracted again. Though this is inefficient it is as satisfactory as a more conventional approach, for the location used ceases to be accessible.

A disadvantage of Pa1 is that calls by reference [22] allow the locations denoted by the parameters to be assigned to in the body of the function unless they are protected by using $E_0(\$E_1)$ instead of E_0E_1. To avoid the need to include \$ at the time the function is applied, in Ma1 we provide $\mathscr{G}[\![E\$]\!]$, which is $\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon.\varepsilon:F\rightarrow\kappa(\lambda\varepsilon'\kappa'.rv(\lambda\beta.\varepsilon\beta\kappa')\varepsilon'),\tau)$. The effect of $(E_0\$)E_1$ may not be that of $E_0(\$E_1)$ because only in the latter is a member of L used to complete the environment of the function closure; for instance, (((fn$u$.fn$v$.(u:=0; put u))\$)1)1 prints 1 whereas (fn$u$.fn$v$.(u:=0; put u))(\$1)1 prints 0. On the other hand, $\mathscr{G}[\![$ fnI..E$]\!]$ coincides with $\mathscr{G}[\![$(fnI.E)\$$]\!]$.

Given an expression E designating a member of L* we can

obtain a new list of locations having the contents of the old by evaluating $\mathscr{E}[\![\pounds E]\!]$, which is

$\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\sigma.(\lambda\alpha^*.\alpha^*:E\to\kappa\alpha^*(updates\alpha^*(holds\varepsilon\sigma)\sigma),\tau)(news(\#\varepsilon|L^*)\sigma))$.

The use of £ in $E_0(\pounds E)$ or in $(E_0\pounds)E_1$ enables us to protect simultaneously all the arguments of a function having several parameters.

Because in Mal any stored value may be denoted it is possible to declare an identifier which signifies a label entry point. Nevertheless we introduce in addition a way of requiring a label to be created as a member of D rather than of V. Such labels are declared by I::E and are given the scopes of their stored counterparts set by reference. To incorporate them in the semantics we have only to define $\mathscr{K}$:Exp→Ide* and $\mathscr{Q}$:Exp→U→K→J*, which list the labels and their entry points. As labels can now be set by incidence $\mathscr{E}[\![E]\!]$ becomes

$\lambda\rho\kappa\sigma.(\lambda\alpha^*.(\lambda\rho'.\alpha^*:E\to\mathscr{G}[\![E]\!]\rho'\kappa(updates\alpha^*(\mathscr{P}[\![E]\!]\rho'\kappa)\sigma),\tau)$

$(fix(\lambda\rho''.\rho[\alpha^*/\mathscr{J}[\![E]\!]][\mathscr{Q}[\![E]\!]\rho''\kappa/\mathscr{K}[\![E]\!]]))(news(\#\mathscr{J}[\![E]\!])\sigma)$.

From henceforth we shall assume that any expression E:Exp and any declaration Δ:Dec in a correct Mal program are such that in the lists $\mathscr{J}[\![E]\!]\S\mathscr{K}[\![E]\!]$ and $\mathscr{J}[\![\Delta]\!]\S\mathscr{K}[\![\Delta]\!]$ no identifier occurs twice. We could of course incorporate this restriction in the semantic equations by making them yield the answer τ when repetitions arose; avoiding this complication (which is conceptually trivial) is tantamount to viewing the tests on the relevant lists as a feature of parsing rather than program execution.

1.4.6. Syntactic transformations preserving meaning.

To illustrate the sense in which declarations by incidence mimic those by reference we now draw up a set of rules for converting a program written in Mal into one which could almost be written in Pal. As will be shown in 2.5.9 these rules preserve

the meaning of the program because they simply induce a correspondence between denoted values and locations containing comparable stored values. Thus, typically, we substitute I:E for certain occurrences of I::E and apply analogous translations to certain declarations and abstractions. Unfortunately we cannot simply change I==E into I=E, as the latter introduces an identifier which may share the location it denotes with something that is then assigned to; we do not, for instance, wish to try to prove the equivalence of u=0 inside v==u inside (u:=1; v) and u=0 inside v=u inside (u:=1; v). Fortunately, built into Pal is a mechanism for avoiding hidden assignments so we can replace I==E by I=$E and, using the additions of 1.4.5, fnI..E by (fnI.E)$.

A similar expedient is adopted with declarations of more than one variable, which are complicated, however, by the fact that L* is a summand of E while V* is not (so that a vector of stored values must be handled as one of locations). We exchange $I_1,\ldots,I_n$==E and fn$I_1,\ldots,I_n$..E for $I_1,\ldots,I_n$=£E and (fn$I_1,\ldots,I_n$.E)£ respectively, the apposite equations for which are given in appendix 1.

The coercions inherent in Mal permit nominal assignments to be made to any identifier, as when the identifier does not signify a member of L a new location is given the stored value instead. Consequently w==0 inside (w:=1; w) and w=$0 inside ($w$:=1; w) are not equivalent although both are legitimate. We therefore arrange that if I switches from being declared by incidence to being declared by reference then all occurrences of it in the relevant scope are replaced by $I; to do so we introduce predicates in Ide↦B* which indicate when a variable is to endure this change. The primitive functions set up in 1.3.2 for dealing with environments will be carried across to these predicates, which will be represented by the letter ψ.

If $\psi[\![I]\!]\downarrow 1$ is *true* we shall presume that I has been converted from denoting a member of V or G to denoting a member of L; to bring about this switch we introduce opt:Ide→[Ide↦B*]→T and its iterative extension to lists, $opts$:Ide*→[Ide↦B*]→T*. Only convenience dictates that the parameters of opt be limited to I and ψ, for all that is actually required is some means, no matter how capricious, of making a choice.

The translation from one program into another is carried out by mappings $\mathscr{e}$:Exp→[Ide↦B*]→Exp, $\mathscr{g}$:Exp→[Ide↦B*]→Exp, $\mathscr{d}$:Dec→[Ide↦B*]→Dec and $\mathscr{l}$:Dec→[Ide↦B*]→Dec which are built up thus:

$\mathscr{e}[\![E]\!]=\lambda\psi.\mathscr{g}[\![E]\!]\psi[false*/\mathscr{f}[\![E]\!]][opts(\mathscr{x}[\![E]\!])\psi/\mathscr{x}[\![E]\!]];$

$\mathscr{g}[\![I]\!]=\lambda\psi.\#\psi[\![I]\!]>0\rightarrow(\psi[\![I]\!]\downarrow 1=true\rightarrow\$I,I),dummy;$

$\mathscr{g}[\![B]\!]=\lambda\psi.B;$

$\mathscr{g}[\![fn()E]\!]=\lambda\psi.fn()\mathscr{e}[\![E]\!]\psi;$

$\mathscr{g}[\![fnI.E]\!]=\lambda\psi.fnI.\mathscr{e}[\![E]\!]\psi[false/I];$

$\mathscr{g}[\![fnI_1,\ldots,I_n.E]\!]=\lambda\psi.fnI_1,\ldots,I_n.\mathscr{e}[\![E]\!]\psi[false*/\langle I_1,\ldots,I_n\rangle];$

$\mathscr{g}[\![fnI..E]\!]=\lambda\psi.opt[\![I]\!]\psi=true\rightarrow\mathscr{e}[\![fnI.E]\!]\psi\$,fnI..\mathscr{e}[\![E]\!]\psi[false/I];$

$\mathscr{g}[\![fnI_1,\ldots,I_n..E]\!]=\lambda\psi.\bigwedge\{opt[\![I_m]\!]\psi\mid 1\leq m\leq n\}=true\rightarrow\mathscr{g}[\![fnI_1,\ldots,I_n.E]\!]\psi\mathfrak{L},$
$fnI_1,\ldots,I_n..\mathscr{e}[\![E]\!]\psi[false*/\langle I_1,\ldots,I_n\rangle];$

$\mathscr{g}[\![OE]\!]=\lambda\psi.O\mathscr{e}[\![E]\!]\psi;$

$\mathscr{g}[\![E_0\Omega E_1]\!]=\lambda\psi.\mathscr{e}[\![E_0]\!]\psi\Omega\mathscr{e}[\![E_1]\!]\psi;$

$\mathscr{g}[\![E_0:=E_1]\!]=\lambda\psi.\mathscr{e}[\![E_0]\!]\psi:=\mathscr{e}[\![E_1]\!]\psi;$

$\mathscr{g}[\![E_1,\ldots,E_n:=E_0]\!]=\lambda\psi.\mathscr{e}[\![E_1]\!]\psi,\ldots,\mathscr{e}[\![E_n]\!]\psi:=\mathscr{e}[\![E_0]\!]\psi;$

$\mathscr{g}[\![\text{get } E]\!]=\lambda\psi.\text{get }\mathscr{e}[\![E]\!]\psi;$

$\mathscr{g}[\![\text{put } E]\!]=\lambda\psi.\text{put }\mathscr{e}[\![E]\!]\psi;$

$\mathscr{g}[\![E_0 \text{ aug } E_1]\!]=\lambda\psi.\mathscr{e}[\![E_0]\!]\psi \text{ aug } \mathscr{e}[\![E_1]\!]\psi;$

$\mathscr{g}[\![E_1,\ldots,E_n]\!]=\lambda\psi.\mathscr{e}[\![E_1]\!]\psi,\ldots,\mathscr{e}[\![E_n]\!]\psi;$

$\mathscr{g}[\![\$E]\!]=\lambda\psi.\$\mathscr{e}[\![E]\!]\psi;$

$\mathscr{g}[\![E\$]\!]=\lambda\psi.\mathscr{e}[\![E]\!]\$\psi;$

$\mathscr{g}[\![\mathfrak{L}E]\!]=\lambda\psi.\mathfrak{L}\mathscr{e}[\![E]\!]\psi;$

$\mathscr{g}[\![E\mathfrak{L}]\!]=\lambda\psi.\mathscr{e}[\![E]\!]\mathfrak{L}\psi;$

$$g[\![E_0 E_1]\!] = \lambda\psi.\, e[\![E_0]\!]\psi\, e[\![E_1]\!]\psi;$$

$$g[\![\texttt{val } E]\!] = \lambda\psi.\, \texttt{val } e[\![E]\!]\psi;$$

$$g[\![\texttt{res } E]\!] = \lambda\psi.\, \texttt{res } e[\![E]\!]\psi;$$

$$g[\![\texttt{goto } E]\!] = \lambda\psi.\, \texttt{goto } e[\![E]\!]\psi;$$

$$g[\![\Delta \texttt{ inside } E]\!] = \lambda\psi.\, d[\![\Delta]\!]\psi \texttt{ inside } e[\![E]\!]\psi[false^*/\mathcal{I}[\![\Delta]\!]][opts(\mathcal{H}[\![\Delta]\!])\psi/\mathcal{H}[\![\Delta]\!]];$$

$$g[\![E_0 ; E_1]\!] = \lambda\psi.\, g[\![E_0]\!]\psi;\ g[\![E_1]\!]\psi;$$

$$g[\![\texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2]\!] = \lambda\psi.\, \texttt{if } e[\![E_0]\!]\psi \texttt{ then } g[\![E_1]\!]\psi \texttt{ else } g[\![E_2]\!]\psi;$$

$$g[\![\texttt{while } E_0 \texttt{ do } E_1]\!] = \lambda\psi.\, \texttt{while } e[\![E_0]\!]\psi \texttt{ do } g[\![E_1]\!]\psi;$$

$$g[\![I{:}E]\!] = \lambda\psi.\, I{:}g[\![E]\!]\psi;$$

$$g[\![I{::}E]\!] = \lambda\psi.\, \psi[\![I]\!]\downarrow 1 = true \rightarrow I{:}g[\![E]\!]\psi, I{::}g[\![E]\!]\psi;$$

$$g[\![(E)]\!] = \lambda\psi.\, (g[\![E]\!]\psi);$$

$$d[\![I{=}E]\!] = \lambda\psi.\, I{=}e[\![E]\!]\psi;$$

$$\mathcal{E}[\![I{=}E]\!] = \lambda\psi.\, I{=}e[\![E]\!]\psi;$$

$$d[\![I_1,\ldots,I_n{=}E]\!] = \lambda\psi.\, I_1,\ldots,I_n{=}e[\![E]\!]\psi;$$

$$\mathcal{E}[\![I_1,\ldots,I_n{=}E]\!] = \lambda\psi.\, I_1,\ldots,I_n{=}e[\![E]\!]\psi;$$

$$d[\![I{==}E]\!] = \lambda\psi.\, opt[\![I]\!]\psi = true \rightarrow I{=}\$e[\![E]\!]\psi, I{==}e[\![E]\!]\psi;$$

$$t[\![I{==}E]\!] = \lambda\psi.\, \psi[\![I]\!]\downarrow 1 = true \rightarrow I{=}e[\![E]\!]\psi, I{==}e[\![E]\!]\psi;$$

$$d[\![I_1,\ldots,I_n{==}E]\!] = \lambda\psi.\, \bigwedge\{opt[\![I_m]\!]\psi \mid 1 \le m \le n\} = true \rightarrow I_1,\ldots,I_n{=}\pounds e[\![E]\!]\psi,$$
$$I_1,\ldots,I_n{==}e[\![E]\!]\psi;$$

$$\mathcal{E}[\![I_1,\ldots,I_n{==}E]\!] = \lambda\psi.\, \bigwedge\{\psi[\![I_m]\!]\downarrow 1 \mid 1 \le m \le n\} = true \rightarrow I_1,\ldots,I_n{=}e[\![E]\!]\psi,$$
$$I_1,\ldots,I_n{==}e[\![E]\!]\psi;$$

$$d[\![\Delta_0 \texttt{ within } \Delta_1]\!] = \lambda\psi.\, d[\![\Delta_0]\!]\psi$$
$$\texttt{within } d[\![\Delta_1]\!]\psi[false^*/\mathcal{I}[\![\Delta_0]\!]][opts(\mathcal{H}[\![\Delta_0]\!])\psi/\mathcal{H}[\![\Delta_0]\!]];$$

$$\mathcal{E}[\![\Delta_0 \texttt{ within } \Delta_1]\!] = \lambda\psi.\, d[\![\Delta_0]\!]\psi$$
$$\texttt{within } \mathcal{E}[\![\Delta_1]\!]\psi[false^*/\mathcal{I}[\![\Delta_0]\!]][opts(\mathcal{H}[\![\Delta_0]\!])\psi/\mathcal{H}[\![\Delta_0]\!]];$$

$$d[\![\Delta_1 \texttt{ and...and } \Delta_n]\!] = \lambda\psi.\, d[\![\Delta_1]\!]\psi \texttt{ and...and } d[\![\Delta_n]\!]\psi;$$

$$\mathcal{E}[\![\Delta_1 \texttt{ and...and } \Delta_n]\!] = \lambda\psi.\, \mathcal{E}[\![\Delta_1]\!]\psi \texttt{ and...and } \mathcal{E}[\![\Delta_n]\!]\psi;$$

$$d[\![\texttt{rec } \Delta]\!] = \lambda\psi.\, \texttt{rec } \mathcal{E}[\![\Delta]\!]\psi[false^*/\mathcal{I}[\![\Delta]\!]][opts(\mathcal{H}[\![\Delta]\!])\psi/\mathcal{H}[\![\Delta]\!]];$$

$$\mathcal{E}[\![\texttt{rec } \Delta]\!] = \lambda\psi.\, \texttt{rec } \mathcal{E}[\![\Delta]\!]\psi;$$

$$d[\![(\Delta)]\!] = \lambda\psi.\, (d[\![\Delta]\!]\psi);$$

$$\mathcal{E}[\![(\Delta)]\!] = \lambda\psi.\, (\mathcal{E}[\![\Delta]\!]\psi).$$

In fact the meaning of a program would be conserved by a transformation less wasteful of storage than that suggested here. Instead of updating a new location with the content of that denoted by I, as we do in $I, we could simply extract the content, leaving it to the context to determine whether another location must be provided. Since Pal does not possess a construct which simply obtains a right hand value without introducing an extra location we have adopted the formulation above. An alternative would be to use $I only when the syntax shows that I appears in a left hand context, but this would complicate our proofs without shedding light on their outcome.

Suppose that when compiled a program is given δ as its environment whereas its transform under the ψ rules above is given the environment δ'. To express the fact that δ and δ' are related by ψ we define

$$apt=\lambda\psi\langle\delta,\delta'\rangle.\bigwedge\{\#\delta'[\![I]\!]=0\vee(\#\delta'[\![I]\!]>0\wedge\#\psi[\![I]\!]=0)\rightarrow true,$$
$$\delta[\![I]\!]\downarrow 1:L\rightarrow\delta'[\![I]\!]\downarrow 1:L\wedge(\psi[\![I]\!]\downarrow 1=false),$$
$$\delta[\![I]\!]\downarrow 1:G\rightarrow\delta'[\![I]\!]\downarrow 1:G\wedge(\psi[\![I]\!]\downarrow 1=false),$$
$$\delta'[\![I]\!]\downarrow 1:V\vee(\delta'[\![I]\!]\downarrow 1:L\wedge(\psi[\![I]\!]\downarrow 1=true))\mid I:Ide\}.$$

As the rules permit rec I==E to be turned into rec I=E one might expect *apt* to allow for the possibility that $\langle\delta[\![I]\!]\downarrow 1,\delta'[\![I]\!]\downarrow 1\rangle:G\times V$. We shall, however, eliminate correspondences of this kind by a further pass through the program which takes place 'after compilation' in the manner to be discussed in 2.7.3.

Applying the transformation to a labelled block moves some members of $\mathscr{K}[\![E]\!]$ into $\mathscr{J}[\![\mathscr{g}[\![E]\!]\psi]\!]$ and thus permutes $\mathscr{J}[\![E]\!]\S\mathscr{K}[\![E]\!]$. The elements of $\mathscr{P}[\![\mathscr{g}[\![E]\!]\psi]\!]\delta'\kappa'\S\mathscr{Q}[\![\mathscr{g}[\![E]\!]\psi]\!]\delta'\kappa'$ have therefore to be ordered before they can be made to tally with $\mathscr{P}[\![E]\!]\delta\kappa\S\mathscr{Q}[\![E]\!]\delta\kappa$; it is this that accounts for the complexity of the definitions in 2.4.5. We carry out this permutation by means of

$$swap=\lambda I^*_0 I^*_1\varepsilon^*.I^*_0=\langle\rangle\rightarrow\langle\rangle,\langle\varepsilon^*\downarrow(\bigsqcup\{\nu\mid I^*_0\downarrow 1=I^*_1\downarrow\nu\})\rangle\S swap(I^*_0\uparrow 1)I^*_1\varepsilon^*.$$

1.5. Conjugate valuations.

1.5.1. Free variables.

One of the most evident connections between programming languages and λ-calculus lies in their use of 'free variables'. We could formalize our intuitive understanding of what is meant by a free occurrence of an identifier by introducing a predicate *free*:[Exp+Dec]→Ide→T with an inductive definition on the constructs of Mal; thus we would let *free*⟦I_0⟧⟦I_1⟧, for instance, be the relation $I_0=I_1$ while *free*⟦Δ inside E⟧⟦I⟧ would be *free*⟦Δ⟧⟦I⟧∨(*free*⟦E⟧⟦I⟧∧~I:𝒟⟦Δ⟧𝒟⟦Δ⟧). However as 1.4.6 has already provided one group of mutually recursive definitions involving the syntax we can avoid setting up another group to govern *free*. Making the temporary assumption that the *opt* function of 1.4.6 is λI.*false*, given some E:Exp and Δ:Dec we simply write *free*⟦E⟧=λI.~(E=ℰ⟦E⟧(λI'.I'=I→⟨⟩,⟨*false*⟩)) and *free*⟦Δ⟧=λI.~(Δ=𝒟⟦Δ⟧(λI'.I'=I→⟨⟩,⟨*false*⟩)). For this *opt* function ℰ⟦E⟧ψ differs from E if and only if ψ⟦I⟧=⟨⟩ for some I:Ide which occurs free in E; this remark, and the analogous one about 𝒟⟦Δ⟧ψ and Δ, can be validated by a structural induction using a more conventional description of *free*. Moreover, when the equality relation adopted by E=ℰ⟦E⟧ψ is the continuous one yielding a proper truth value only for E and ℰ⟦E⟧ψ without constituents which are ⊥ or T, *free* is also continuous.

We require *free*⟦E⟧⟦res⟧ and *free*⟦Δ⟧⟦res⟧, which are *true* if and only if res occurs outside the outermost val block of E:Exp and Δ:Dec. In particular, for any expression E we stipulate that *free*⟦val E⟧⟦res⟧ be *false* and that *free*⟦res E⟧⟦res⟧ be *true*; the remainder of the definition of *free*⟦E⟧⟦res⟧ is obvious.

In 2.1.4 we shall wish to reduce the environment attached to a declaration by discarding all the denotations which do not

correspond to free variables. Since the domain U to which this process will apply has a third component besides those for Ide and res we truncate its members by means of

rend⟦E⟧=λρ.⟨(λI.*free*⟦E⟧⟦I⟧∧#ρ⟦I⟧>0→⟨ρ⟦I⟧↓1⟩,⟨⟩),
(*free*⟦E⟧⟦res⟧∧#ρ⟦res⟧>0→⟨ρ⟦res⟧↓1⟩,⟨⟩),⟨⟩⟩;

rend⟦Δ⟧=λρ.⟨(λI.*free*⟦Δ⟧⟦I⟧∧#ρ⟦I⟧>0→⟨ρ⟦I⟧↓1⟩,⟨⟩),
(*free*⟦Δ⟧⟦res⟧∧#ρ⟦res⟧>0→⟨ρ⟦res⟧↓1⟩,⟨⟩),⟨⟩⟩;

tear⟦E⟧=λρ.⟨(λI.(*free*⟦E⟧⟦I⟧∨I:𝒥⟦E⟧§𝒳⟦E⟧)∧#ρ⟦I⟧>0→⟨ρ⟦I⟧↓1⟩,⟨⟩),
(*free*⟦E⟧⟦res⟧∧#ρ⟦res⟧>0→⟨ρ⟦res⟧↓1⟩,⟨⟩),⟨⟩⟩;

tear⟦Δ⟧=λρ.⟨(λI.(*free*⟦Δ⟧⟦I⟧∨I:𝒥⟦Δ⟧§𝒳⟦Δ⟧)∧#ρ⟦I⟧>0→⟨ρ⟦I⟧↓1⟩,⟨⟩),
(*free*⟦Δ⟧⟦res⟧∧#ρ⟦res⟧>0→⟨ρ⟦res⟧↓1⟩,⟨⟩),⟨⟩⟩.

These functions undergo trivial modifications when the domain of environments is [Ide↦D°*]×K°* as it is in 1.3.2.

To express the belief that an environment is large enough to give meaning to a program we define also

rent⟦E⟧=λρ.⋀{~*free*⟦E⟧⟦I⟧∨#ρ⟦I⟧>0 | I:Ide}∧(~*free*⟦E⟧⟦res⟧∨#ρ⟦res⟧>0);

rent⟦Δ⟧=λρ.⋀{~*free*⟦Δ⟧⟦I⟧∨#ρ⟦I⟧>0 | I:Ide}∧(~*free*⟦Δ⟧⟦res⟧∨#ρ⟦res⟧>0);

torn⟦E⟧=λρ.⋀{~I:𝒥⟦E⟧§𝒳⟦E⟧∨#ρ⟦I⟧>0 | I:Ide}∧*rent*⟦E⟧ρ;

torn⟦Δ⟧=λρ.⋀{~I:𝒥⟦Δ⟧§𝒳⟦Δ⟧∨#ρ⟦I⟧>0 | I:Ide}∧*rent*⟦Δ⟧ρ.

With the aid of these definitions we can now verify that when evaluating an expression or a declaration in standard semantics the only significant values in the environment are those representing the most recent incarnations of the free variables. This property is not shared with the stack semantics to be used to describe implementations of Mal in 3.1.1, for which it is essential that the entire environment appear in the evaluation procedure. To establish that the environment can be cut back to a free variable list when the equations of appendix 1 are being applied we introduce a method of pruning declared environments, namely

snip=λΔρ.⟨(λI.I:𝒥⟦Δ⟧§𝒳⟦Δ⟧→⟨ρ⟦I⟧↓1⟩,⟨⟩),⟨⟩⟩.

1.5.2. Proposition.

In standard semantics all E:Exp satisfy $\mathscr{E}[\![E]\!]=\mathscr{E}[\![E]\!]\circ rend[\![E]\!]$, $\mathscr{L}[\![E]\!]=\mathscr{L}[\![E]\!]\circ rend[\![E]\!]$ and $\mathscr{R}[\![E]\!]=\mathscr{R}[\![E]\!]\circ rend[\![E]\!]$, together with $\mathscr{G}[\![E]\!]=\mathscr{G}[\![E]\!]\circ tear[\![E]\!]$, $\mathscr{P}[\![E]\!]=\mathscr{P}[\![E]\!]\circ tear[\![E]\!]$ and $\mathscr{Q}[\![E]\!]=\mathscr{Q}[\![E]\!]\circ tear[\![E]\!]$; moreover all Δ:Dec satisfy $\mathscr{D}[\![\Delta]\!]=\mathscr{D}[\![\Delta]\!]\circ rend[\![\Delta]\!]=\lambda\rho\chi.\mathscr{D}[\![\Delta]\!]\rho(\chi\circ snip[\![\Delta]\!])$ and $\mathscr{T}[\![\Delta]\!]=\mathscr{T}[\![\Delta]\!]\circ tear[\![\Delta]\!]=\lambda\rho\chi.\mathscr{T}[\![\Delta]\!]\rho(\chi\circ snip[\![\Delta]\!])$.

⊲The proof of this result involves a structural induction on the constructs of Mal. Since it is merely an elementary precursor of the technique required by such proofs as that of 1.5.5 we shall not embark on a discussion of its details.⊳

1.5.3. Expression exits.

In order to list the possible expressed values returned by part of a program we introduce $exit$:[Exp+Dec]→[Exp*+Dec*] with the intention that $exit[\![E]\!]$ reveal which expressions may be encountered last during the evaluation of E. Though it may not be decidable whether this evaluation will ever end we can nevertheless collect up the exits thus:

$exit[\![I]\!]=\langle I\rangle$;

$exit[\![B]\!]=\langle B\rangle$;

$exit[\![\Phi]\!]=\langle \Phi\rangle$;

$exit[\![OE]\!]=\langle OE\rangle$;

$exit[\![E_0\Omega E_1]\!]=\langle E_0\Omega E_1\rangle$;

$exit[\![E_1,\ldots,E_n]\!]=\langle (E_1,\ldots,E_n)\rangle$;

$exit[\![E_0E_1]\!]=\langle E_0E_1\rangle$;

$exit[\![$ val E$]\!]=\langle$ val E$\rangle$;

$exit[\![$ res E$]\!]=\langle$ res E$\rangle$;

$exit[\![$ goto E$]\!]=\langle$ goto E$\rangle$;

$exit[\![\Delta$ inside E$]\!]=\langle\Delta$ inside E$\rangle$;

$exit[\![E_0;\ E_1]\!]=exit[\![E_1]\!]$;

$exit[\![$ if E_0 then E_1 else $E_2]\!] = exit[\![E_1]\!] \S exit[\![E_2]\!]$;

$exit[\![$ while E_0 do $E_1]\!] = \langle$ while E_0 do $E_1 \rangle$;

$exit[\![I:E]\!] = exit[\![E]\!]$;

$exit[\![I::E]\!] = exit[\![E]\!]$;

$exit[\![(E)]\!] = exit[\![E]\!]$.

No other forms of expression have obvious interpretations as commands followed by more expressions, so $exit[\![E]\!]$ will be taken to be $\langle E \rangle$ except in some of the cases above; $exit[\![E_0 := E_1]\!]$, for instance, will be $\langle E_0 := E_1 \rangle$. For a declaration Δ $exit[\![\Delta]\!]$ will split up Δ into its constituent simple declarations by means of:

$exit[\![I=E]\!] = \langle I=E \rangle$;

$exit[\![I_1,\ldots,I_n=E]\!] = \langle I_1,\ldots,I_n=E \rangle$;

$exit[\![I==E]\!] = \langle I==E \rangle$;

$exit[\![I_1,\ldots,I_n==E]\!] = \langle I_1,\ldots,I_n==E \rangle$;

$exit[\![\Delta_0$ within $\Delta_1]\!] = \langle \Delta_0$ within $\Delta_1 \rangle$;

$exit[\![\Delta_1$ and...and $\Delta_n]\!] = exit[\![\Delta_1]\!] \S \ldots \S exit[\![\Delta_n]\!]$;

$exit[\![$rec $\Delta]\!] = exit[\![\Delta]\!]$;

$exit[\![(\Delta)]\!] = exit[\![\Delta]\!]$.

Any member of the list $exit[\![\Delta]\!]$ will be termed an 'exit' of the declaration Δ, and similar nomenclature will apply to elements of $exit[\![E]\!]$.

For ease of implementation it is frequently wise to restrict programs so that references to locations are not passed out of the blocks in which they are created. Accordingly we provide a syntactic constraint sufficient to ensure that this situation exists. Loosely speaking, we allow an identifier to be returned as the result of an expression only when it denotes a member of V, while we allow an abstraction to be returned only when all its free variables are global to the expression. To show that this condition is fulfilled we introduce some ψ:Ide$\rightarrow$B* such

that $\psi[\![I]\!]\downarrow 1$ is 0, 1 or 2 when I denotes a member of L, V or G respectively unless I is local to the block, when $\psi[\![I]\!]\downarrow 1$ should be 3. Following the precedent set by 1.4.6 we extend the notation provided in 1.3.2 to cover this new element of Ide↦B*; the definitions in 1.5.1 will also be regarded as being applicable to this domain. We now set up *cramped*:[Exp+Dec]→[Ide↦B*]→T recursively as follows: *cramped*$[\![E]\!]\psi$ is to be *true* if and only if any exit I of E in Ide satisfies $\psi[\![I]\!]\downarrow 1=1$, any exit Φ of E in Abs satisfies $\vee\{(free[\![\Phi]\!][\![I]\!]\wedge\psi[\![I]\!]\downarrow 1=3)\vee free[\![\Phi]\!][\![res]\!]\,|\,I:Ide\}=false$, no exit of E has the form get E_0, put E_0, E_0 aug E_1, $(E_1,\ldots,E_n)$, £E_0, E_0E_1 or res E_0, any exit of the form \$$E_0$, E_0\$, E_0£ or val E_0 is subject to $cramped[\![E_0]\!]\psi[3^*/\mathscr{I}[\![E_0]\!]\,\S\mathscr{K}[\![E_0]\!]]=true$, and any exit of the form Δ_0 inside E_0 is subject to $cramped[\![E_0]\!]\psi[3^*/\mathscr{I}[\![\Delta_0]\!]\,\S\mathscr{K}[\![\Delta_0]\!]\,\S\mathscr{I}[\![E_0]\!]\,\S\mathscr{K}[\![E_0]\!]]=true$. These limitations are appropriate only for the particular languages we consider, in which OE_0, $E_0\Omega E_1$, $E_0:=E_1$ and while E_0 do E_1 return as their results expressed values that are inevitably in B, but variants of the stipulations above can be given for other algorithmic languages. For reasons that will become apparent in 2.6.6 it is convenient to let $cramped[\![\Delta]\!]\psi$ be *true* if and only if any exit of the form I==E is such that $cramped[\![E]\!]\psi[3^*/\mathscr{I}[\![E]\!]\,\S\mathscr{K}[\![E]\!]]=true$, any exit of the form $I_1,\ldots,I_n$==E is such that E is $E_1,\ldots,E_n$ or £$(E_1,\ldots,E_n)$ where $cramped[\![E_m]\!]\psi[3^*/\mathscr{I}[\![E_m]\!]\,\S\mathscr{K}[\![E_m]\!]]=true$ whenever $1\leq m\leq n$, and any exit of the form Δ_0 within Δ_1 has $cramped[\![\Delta_1]\!]\psi[3^*/\mathscr{I}[\![\Delta_0]\!]\,\S\mathscr{K}[\![\Delta_0]\!]]=true$.

We shall also presume that for every ψ, E and Δ $torn[\![E]\!]\psi=true$ when $cramped[\![E]\!]\psi=true$ and $torn[\![\Delta]\!]\psi=true$ when $cramped[\![\Delta]\!]\psi=true$. A more sophisticated set of restrictions than that imposed by *cramped* will be introduced in 3.1.4, but for the purposes of 2.6.5 *cramped* will be perfectly adequate.

1.5.4. Programs without jumps.

Continuations are required by the equations of appendix 1 merely to provide an understanding of jumps and of certain aspects of unending computations [16]. It is therefore natural to expect that a programming language which has been emasculated by removing its imperative parts may be defined without reference to continuations. Here we shall carry out this operation on Mal in order to prove that the resulting equations are equivalent to those introduced above.

Applying a function may occasion a jump out of the code for the function, so it is necessary to restrict those abstractions which can be applied to those devoid of goto statements; for simplicity we choose to ban function application entirely. Likewise we must exclude the possibility that an identifier could denote a member of G under ρ:U by testing some ψ:Ide→B* such that $\psi[\![I]\!]\downarrow 1=2$ whenever $\rho[\![I]\!]\downarrow 1$:G. We now define *crushed*:[Exp+Dec]→[Ide→B*]→T somewhat informally thus: *crushed*$[\![E]\!]\psi$ is *true* if and only if no expressions of the form E_0E_1, val E_0, res E_0, goto E_0, I:E_0 or I::E_0 occur in E except within an abstraction and any identifier I appearing in E outside an abstraction and having *free*$[\![E]\!][\![I]\!]$=*true* satisfies $\sim\psi[\![I]\!]\downarrow 1=2$. For any Δ *crushed*$[\![\Delta]\!]\psi$ will be *true* if and only if any exit of the form I=E, $I_1,\ldots,I_n$=E, I==E or $I_1,\ldots,I_n$==E satisfies *crushed*$[\![E]\!]\psi$=*true*, any exit of the form Δ_0 within Δ_1 satisfies both *crushed*$[\![\Delta_0]\!]\psi$=*true* and *crushed*$[\![\Delta_1]\!]\psi$=*true* and, when Δ is given the form rec Δ_2, *crushed*$[\![\Delta_2]\!]\psi[0*/\mathcal{L}[\![\Delta_2]\!]][2*/\mathcal{K}[\![\Delta_2]\!]]$=*true*.

Although the connection between the standard kind of semantics with continuations and that without is of interest in its own right we are investigating it mainly because of the role it will play in 2.7.6. Accordingly we have not endeavoured to reduce the syntactic constraints to the least possible, as the

expressions permitted above encompass all those needed for our ultimate purpose. In particular we could adapt the techniques of 2.2.5 to allow function applications like E_0E_1 to appear amid the expressions free from jumps so long as semantic stipulations are made about those closures which are stored or denoted, but to do so would detract from the elegance of the ensuing propositions.

Our intention is to set up for every valuation a conjugate which when given an abstract program, an environment and a store as arguments returns a result which when supplied to any continuation yields precisely the same answer as would have been obtained from the original valuation by applying it to the same program, environment, continuation and store. Strictly speaking the conjugates cannot be made to elucidate the meaning of a program since they use environments which are created in a framework where closures must take continuations as parameters. Equations which are entirely free from the influence of continuations can be constructed with ease, and the methods of 2.2.8 can be applied to proving them equivalent to the standard kind, but this is beside the point; here we are content to take the conjugate of $\mathscr{F}$, $\P\mathscr{F}$, to be $\mathscr{F}$ itself, just as the conjugate of $\mathscr{B}$, $\P\mathscr{B}$, is $\mathscr{B}$. There being no label declarations in the expressions we allow the conjugates of $\mathscr{E}$ and $\mathscr{G}$ to coincide; thus we may write $\P\mathscr{E}$ for $\P\mathscr{G}$.

The remaining conjugate valuations will emerge in the course of the following proposition. Suffice it to say here that those acting on expressions are members of Exp→U→S→[E×S] whereas $\P\mathscr{D}$ and $\P\mathscr{T}$, which act on declarations, are members of Dec→U→S→[U×S]. More generally, given a lattice Syn (comprising the terminal symbols derived from a suitable system of syntactic production rules) and a valuation $\mathscr{X}$:Syn→R_1→...→R_n→[H_1→...→H_m→A]→[I_1→...→I_l→A], say, $\P\mathscr{X}$, the conjugate of $\mathscr{X}$, satisfies $\P\mathscr{X}$:Syn→R_1→...→R_n→[I_1×...×I_l]→[H_1×...×H_m].

1.5.5. Proposition.

Let ψ and ρ be such that for all I:Ide $\psi[\![I]\!]\downarrow 1=2$ if $\rho[\![I]\!]\downarrow 1$:G. For every proper σ and every E:Exp satisfying $crushed[\![E]\!]\psi=true$ either $\lambda\kappa.\mathscr{E}[\![E]\!]\rho\kappa\sigma$ is improper or $\neg\mathscr{E}[\![E]\!]\rho\sigma$ is proper and $\lambda\kappa.\mathscr{E}[\![E]\!]\rho\kappa\sigma=\lambda\kappa.(\kappa\star\neg\mathscr{E}[\![E]\!]\rho)\sigma$; analogous conclusions hold for $\mathscr{L}$, $\mathscr{R}$ and $\mathscr{G}$ also. For every proper σ and every Δ:Dec satisfying $crushed[\![\Delta]\!]\psi=true$ either $\lambda\chi.\mathscr{D}[\![\Delta]\!]\rho\chi\sigma$ is improper or $\neg\mathscr{D}[\![\Delta]\!]\rho\sigma$ is proper, $\lambda\chi.\mathscr{D}[\![\Delta]\!]\rho\chi\sigma=\lambda\chi.(\chi\star\neg\mathscr{D}[\![\Delta]\!]\rho)\sigma$ and, when I:Ide is such that $\#\rho'[\![I]\!]>0$ for $\rho'=\mathscr{D}[\![\Delta]\!]\rho\sigma\downarrow 1$, $\rho'[\![I]\!]\downarrow 1$:E and I:$\mathscr{I}[\![\Delta]\!]\S\mathscr{K}[\![\Delta]\!]$; analogous conclusions hold for $\mathscr{T}$.

◁Observe first that we can set up $\neg\mathscr{L}$ and $\neg\mathscr{R}$ as follows. Take any expression E such that if ρ and σ are proper and if $crushed[\![E]\!]\psi=true$ for some ψ having $\psi[\![I]\!]\downarrow 1=2$ when $\rho[\![I]\!]\downarrow 1$:G for all I:Ide then $\lambda\kappa.\mathscr{G}[\![E]\!]\rho\kappa\sigma$ is improper or $\lambda\kappa.\mathscr{G}[\![E]\!]\rho\kappa\sigma=\lambda\kappa.(\kappa\star\neg\mathscr{G}[\![E]\!]\rho)\sigma$, where $\neg\mathscr{G}[\![E]\!]\rho\sigma$ is proper; as noted above we may take $\neg\mathscr{E}$ to be $\neg\mathscr{G}$. Define $\neg\mathscr{L}[\![E]\!]\rho$ and $\neg\mathscr{R}[\![E]\!]\rho$ to be

$(\lambda\varepsilon\sigma.\varepsilon$:L$\rightarrow\langle\varepsilon,\sigma\rangle,\langle new\sigma,update(new\sigma)\varepsilon\sigma\rangle)\star\neg\mathscr{E}[\![E]\!]\rho$ and

$(\lambda\varepsilon\sigma.\varepsilon$:L$\rightarrow(area\varepsilon\sigma\rightarrow\langle hold\varepsilon\sigma,\sigma\rangle,\langle\tau,\tau\rangle),\langle\varepsilon,\sigma\rangle)\star\neg\mathscr{E}[\![E]\!]\rho$ respectively.

Suppose that for some proper σ $\lambda\kappa.\mathscr{E}[\![E]\!]\rho\kappa\sigma$ is neither $\perp$ nor τ; then $\lambda\kappa.\mathscr{L}[\![E]\!]\rho\kappa\sigma=\lambda\kappa.\mathscr{E}[\![E]\!]\rho(lv\kappa)\sigma=\lambda\kappa.(lv\kappa\star\neg\mathscr{E}[\![E]\!]\rho)\sigma$ and $\lambda\kappa.\mathscr{R}[\![E]\!]\rho\kappa\sigma=\lambda\kappa.\mathscr{E}[\![E]\!]\rho(rv\kappa)\sigma=\lambda\kappa.(rv\kappa\star\neg\mathscr{E}[\![E]\!]\rho)\sigma$. Let $\langle\varepsilon_0,\sigma_0\rangle$ be $\neg\mathscr{E}[\![E]\!]\rho\sigma$ which, being $\neg\mathscr{G}[\![E]\!]\rho\sigma$, is proper. If ε_0 is a location, $\lambda\kappa.lv\kappa\varepsilon_0\sigma_0=\lambda\kappa.(\kappa\star\neg\mathscr{L}[\![E]\!]\rho)\sigma$ and $\lambda\kappa.rv\kappa\varepsilon_0\sigma_0=\lambda\kappa.area\varepsilon_0\sigma_0\rightarrow(\kappa\star\neg\mathscr{R}[\![E]\!]\rho)\sigma,\tau$ from the definitions of 1.3.5; furthermore $hold\varepsilon_0\sigma_0$ is proper. If ε_0 is a stored value, $\lambda\kappa.lv\kappa\varepsilon_0\sigma_0=\lambda\kappa.new\sigma_0$:L$\rightarrow(\kappa\star\neg\mathscr{L}[\![E]\!]\rho)\sigma,\tau$ and $\lambda\kappa.rv\kappa\varepsilon_0\sigma_0=\lambda\kappa.(\kappa\star\neg\mathscr{R}[\![E]\!]\rho)\sigma$. Thus unless $\lambda\kappa.\mathscr{L}[\![E]\!]\rho\kappa\sigma$ is improper it coincides with $\lambda\kappa.(\kappa\star\neg\mathscr{L}[\![E]\!]\rho)\sigma$ and $\neg\mathscr{L}[\![E]\!]\rho\sigma$ is proper; likewise if $\lambda\kappa.\mathscr{R}[\![E]\!]\rho\kappa\sigma$ is proper it is $\lambda\kappa.(\kappa\star\neg\mathscr{R}[\![E]\!]\rho)\sigma$ and $\neg\mathscr{R}[\![E]\!]\rho\sigma$ cannot be improper.

The proof now proceeds by structural induction on the

expressions E and declarations Δ having $crushed[\![E]\!]\psi=true$ and $crushed[\![\Delta]\!]\psi=true$ for some suitable ψ.

Suppose that E is an identifier I; then if $\rho[\![I]\!]\downarrow 1$ is improper $\lambda\kappa.\mathscr{G}[\![I]\!]\rho\kappa\sigma$ is improper for all σ whilst otherwise $\rho[\![I]\!]\downarrow 1$ cannot be in G ($crushed[\![I]\!]\psi$ being *true*) and $\lambda\kappa.\mathscr{G}[\![I]\!]\rho\kappa\sigma=\lambda\kappa.(\kappa*\P\mathscr{G}[\![I]\!]\rho)\sigma$ where $\P\mathscr{G}[\![I]\!]$ is defined to be $\lambda\rho\sigma.\langle\rho[\![I]\!]\downarrow 1|E,\sigma\rangle$.

Because $\P\mathscr{F}$ is $\mathscr{F}$ and $\P\mathscr{B}$ is $\mathscr{B}$ it is obvious that the result holds when E is an abstraction Φ or a constant B if we take $\P\mathscr{G}[\![\Phi]\!]$ and $\P\mathscr{G}[\![B]\!]$ to be $\lambda\rho\sigma.\langle\mathscr{F}[\![\Phi]\!]\rho,\sigma\rangle$ and $\lambda\rho\sigma.\langle\mathscr{B}[\![B]\!],\sigma\rangle$ respectively.

We shall omit all consideration of the other types of expression save three, which will be enough to indicate what procedures are adopted for the remainder. Suppose that E is of the form $E_0:=E_1$ and that $crushed[\![E]\!]\psi=true$; assume also that expressions are evaluated from left to right, the proof being almost identical if this is not so. For any proper σ_0 $\mathscr{L}[\![E_0]\!]\rho\kappa_1\sigma_0$ is $\perp$ or $\top$ for all κ_1 or $\mathscr{L}[\![E_0]\!]\rho\kappa_1\sigma_0=(\kappa_1*\P\mathscr{L}[\![E_0]\!]\rho)\sigma_0$ for all κ_1 as $crushed[\![E_0]\!]\psi=true$. In particular, writing $\lambda\varepsilon_0.\mathscr{R}[\![E_1]\!]\rho(\lambda\varepsilon_1.\kappa_0(dummy)\circ update\varepsilon_0\varepsilon_1)$ as κ_1 for any κ_0, either $\mathscr{G}[\![E_0:=E_1]\!]\rho\kappa_0\sigma_0$ is $\perp$ or $\top$ for all κ_0 or $\mathscr{G}[\![E_0:=E_1]\!]\rho\kappa_0\sigma_0=(\kappa_1*\P\mathscr{L}[\![E_0]\!]\rho)\sigma_0$ for all κ_0. If the latter holds write $\P\mathscr{L}[\![E_0]\!]\rho\sigma_0$ as $\langle\varepsilon_0,\sigma_1\rangle$ and $\lambda\varepsilon_1.\kappa_0(dummy)\circ update\varepsilon_0\varepsilon_1$ as κ_2 for all κ_0; then either $\mathscr{R}[\![E_1]\!]\rho\kappa_2\sigma_1$ is $\perp$ or $\top$ for all κ_0 or $\mathscr{R}[\![E_1]\!]\rho\kappa_2\sigma_1=(\kappa_2*\P\mathscr{R}[\![E_1]\!]\rho)\sigma_1$ for all κ_2, as $crushed[\![E_1]\!]\psi=true$ also. Consequently either $\mathscr{G}[\![E_0:=E_1]\!]\rho\kappa_0\sigma_0$ is $\perp$ or $\top$ for all κ_0 or $\mathscr{G}[\![E_0:=E_1]\!]\rho\kappa_0\sigma_0=(\kappa_0*\P\mathscr{G}[\![E_0:=E_1]\!]\rho)\sigma_0$ for all κ_0, where $\P\mathscr{G}[\![E_0:=E_1]\!]$ is $\lambda\rho.(\lambda\varepsilon_0.(\lambda\varepsilon_1\sigma.\langle dummy,update\varepsilon_0\varepsilon_1\sigma\rangle)*\P\mathscr{R}[\![E_1]\!]\rho)*\P\mathscr{L}[\![E_0]\!]\rho$.

Suppose that E is of the form Δ_0 inside E_0 and that $crushed[\![E]\!]\psi=true$ for some ψ having $\psi[\![I]\!]\downarrow 1=2$ whenever $\rho[\![I]\!]\downarrow 1:G$, so that $crushed[\![\Delta_0]\!]\psi=true$ and $crushed[\![E_0]\!]\psi=true$. For every proper

σ_0 either $\lambda\chi.\mathscr{D}[\![\Delta_0]\!]\rho\chi\sigma_0$ is improper or there are ρ_0 and σ_1 with $\lambda\chi.\mathscr{D}[\![\Delta_0]\!]\rho\chi\sigma_0=\lambda\chi.\chi\rho_0\sigma_1$ and $\langle\rho_0,\sigma_1\rangle=\P\mathscr{D}[\![\Delta_0]\!]\rho\sigma_0$; furthermore $\rho_0[\![I]\!]\downarrow1$:E when I:Ide and $\#\rho_0[\![I]\!]>0$. In the latter case $\psi[\![I]\!]\downarrow1=2$ whenever $divert\rho\rho_0[\![I]\!]\downarrow1$:G and σ_1 is proper so either $\lambda\kappa.\mathscr{L}[\![E_0]\!](divert\rho\rho_0)\kappa\sigma_1$ is improper or $\lambda\kappa.\mathscr{L}[\![E_0]\!](divert\rho\rho_0)\kappa\sigma_1=\lambda\kappa.(\kappa*\P\mathscr{L}[\![E_0]\!](divert\rho\rho_0))\sigma_1$. Hence taking $\P\mathscr{G}[\![\Delta_0$ inside $E_0]\!]$ to be $\lambda\rho.(\lambda\rho_0.\P\mathscr{L}[\![E_0]\!](divert\rho\rho_0))*\P\mathscr{D}[\![\Delta_0]\!]\rho$, for all proper σ $\lambda\kappa.\mathscr{G}[\![\Delta_0$ inside $E_0]\!]\rho\kappa\sigma$ is improper if it is not $\lambda\kappa.(\kappa*\P\mathscr{G}[\![\Delta_0$ inside $E_0]\!]\rho)\sigma$.

⊲A more complex approach is essential when E is while E_0 do E_1. If $crushed[\![E]\!]\psi=true$ and $\psi[\![I]\!]\downarrow1=2$ whenever $\rho[\![I]\!]\downarrow1$:G, we define for every $n\geq0$ γ_n:G and ϕ_n:[S→[E×S]]° by $\gamma_0=\perp$, $\phi_0=\perp$, $\gamma_{n+1}=\lambda\kappa.\mathscr{R}[\![E_0]\!]\rho(\lambda\varepsilon.\varepsilon\rightarrow\mathscr{G}[\![E_1]\!]\rho(\lambda\varepsilon.\gamma_n\kappa),\kappa(dummy))$ $\phi_{n+1}=(\lambda\varepsilon\sigma.\varepsilon\rightarrow\phi_n(\P\mathscr{G}[\![E_1]\!]\rho\sigma\downarrow2),\langle dummy,\sigma\rangle)*\P\mathscr{R}[\![E_0]\!]\rho$; then by induction $\gamma_{n+1}\sqsupseteq\gamma_n$ and $\phi_{n+1}\sqsupseteq\phi_n$ for all $n\geq0$, while $\mathscr{G}[\![$while E_0 do $E_1]\!]\rho=\bigsqcup\gamma_n$.

⊲Assume that for some $n\geq0$ and every proper σ unless $\lambda\kappa.\gamma_n\kappa\sigma$ is improper it equals $\lambda\kappa.(\kappa*\phi_n)\sigma$ and $\phi_n\sigma$ is proper; to establish the analogous contention for $n+1$ take any σ_0 such that $\lambda\kappa.\gamma_{n+1}\kappa\sigma_0$ is proper.

It is clear from the definition of γ_{n+1} that $\mathscr{R}[\![E_0]\!]\rho\kappa\sigma_0$ is proper for some κ and thus that $\lambda\kappa.\gamma_{n+1}\kappa\sigma_0=\lambda\kappa.\varepsilon_0\rightarrow\mathscr{G}[\![E_1]\!]\rho(\lambda\varepsilon.\gamma_n\kappa)\sigma_1,\kappa(dummy)\sigma_1$, where $\langle\varepsilon_0,\sigma_1\rangle$ is the proper pair $\P\mathscr{R}[\![E_0]\!]\rho\sigma_0$; moreover ε_0 must be *true* or *false* as $\lambda\kappa.\gamma_{n+1}\kappa\sigma_0$ is proper. If $\varepsilon_0=true$, $\mathscr{G}[\![E_1]\!]\rho(\lambda\varepsilon.\gamma_n\kappa)\sigma_1$ must be proper for some κ and $\lambda\kappa.\mathscr{G}[\![E_1]\!]\rho(\lambda\varepsilon.\gamma_n\kappa)\sigma_1$ must be $\lambda\kappa.(\lambda\varepsilon.\gamma_n\kappa)\varepsilon_1\sigma_2$, where $\langle\varepsilon_1,\sigma_2\rangle$ is the proper pair $\P\mathscr{G}[\![E_1]\!]\rho\sigma_1$; by the induction hypothesis $\lambda\kappa.(\lambda\varepsilon.\gamma_n\kappa)\varepsilon_1\sigma_2=\lambda\kappa.(\kappa*\phi_n)\sigma_2$ and $\phi_n\sigma_2$ is proper. If $\varepsilon_0=false$, on the other hand,

$\lambda\kappa.\gamma_{n+1}\kappa\sigma_0 = \lambda\kappa.(\kappa*(\lambda\sigma.\langle dummy,\sigma\rangle))\sigma_1$

$= \lambda\kappa.(\kappa*(\lambda\varepsilon\sigma.\varepsilon\rightarrow\phi_n(\mathcal{C}[\![E_1]\!]\rho\sigma\downarrow 2),\langle dummy,\sigma\rangle)\varepsilon_0)\sigma_1$

$= \lambda\kappa.(\kappa*\phi_{n+1})\sigma_0.$

Hence in general

$\lambda\kappa.\gamma_{n+1}\kappa\sigma_0 = \lambda\kappa.\varepsilon_0\rightarrow(\lambda\varepsilon.\gamma_n\kappa)\varepsilon_1\sigma_2,(\kappa*\phi_{n+1})\sigma_0$

$= \lambda\kappa.\varepsilon_0\rightarrow(\kappa*\phi_{n+1})\sigma_0,(\kappa*\phi_{n+1})\sigma_0$

$= \lambda\kappa.(\kappa*\phi_{n+1})\sigma_0$

and $\phi_{n+1}\sigma_0$ is proper.

Accordingly for all proper σ either $\lambda\kappa.\gamma_{n+1}\kappa\sigma$ is improper or it equals $\lambda\kappa.(\kappa*\phi_{n+1})\sigma$ and $\phi_{n+1}\sigma$ is proper, so long as the corresponding remarks apply to γ_n. In addition $\lambda\kappa.\gamma_0\kappa\sigma=\perp$ for all proper σ by definition, so we may deduce that for all $n\geq 0$ and for all proper σ unless $\lambda\kappa.\gamma_n\kappa\sigma$ is improper $\phi_n\sigma$ is proper and $\lambda\kappa.\gamma_n\kappa\sigma=\lambda\kappa.(\kappa*\phi_n)\sigma$.▷

Since $\gamma_{n+1}\sqsupseteq\gamma_n$ for every $n\geq 0$, for all proper σ either $\lambda\kappa.\bigsqcup\gamma_n\kappa\sigma$ is improper or there is some $m\geq 0$ such that when $n\geq m$ $\phi_n\sigma$ is proper and $\lambda\kappa.\gamma_n\kappa\sigma=\lambda\kappa.(\kappa*\phi_n)\sigma$; under the latter circumstances $\bigsqcup\phi_n\sigma$ is proper as $\tau\prec\tau$ in $E\times S$ and $\lambda\kappa.\bigsqcup\gamma_n\kappa\sigma=\lambda\kappa.(\kappa*\bigsqcup\phi_n)\sigma$ by continuity. Defining $\mathcal{C}[\![\textbf{while } E_0 \textbf{ do } E_1]\!]\rho$ to be $\bigsqcup\phi_n$ or $fix(\lambda\phi.(\lambda\varepsilon\sigma.\varepsilon\rightarrow\phi(\mathcal{C}[\![E_1]\!]\rho\sigma\downarrow 2),\langle dummy,\sigma\rangle)*\mathcal{R}[\![E_0]\!]\rho)$, unless $\lambda\kappa.\mathcal{S}[\![\textbf{while } E_0 \textbf{ do } E_1]\!]\rho\kappa\sigma$ is improper it equals $\lambda\kappa.(\kappa*\mathcal{C}[\![\textbf{while } E_0 \textbf{ do } E_1]\!]\rho)\sigma$ and $\mathcal{C}[\![\textbf{while } E_0 \textbf{ do } E_1]\!]\rho\sigma$ is proper.▷

The induction on declarations is very similar, as we can take $\mathcal{D}[\![I=E]\!]$ to be $\lambda\rho.(\lambda\varepsilon\sigma.\langle arid[\varepsilon/I],\sigma\rangle)*\mathcal{E}[\![E]\!]\rho$, $\mathcal{T}[\![I=E]\!]$ to be $\lambda\rho.(\lambda\varepsilon\sigma.\langle arid[\rho[\![I]\!]\downarrow 1/I],update(\rho[\![I]\!]\downarrow 1)\varepsilon\sigma\rangle)*\mathcal{R}[\![E]\!]\rho$ and $\mathcal{D}[\![I==E]\!]$ and $\mathcal{T}[\![I==E]\!]$ to be $\lambda\rho.(\lambda\varepsilon\sigma.\langle arid[\varepsilon/I],\sigma\rangle)*\mathcal{R}[\![E]\!]\rho$. Analogous remarks apply to $I_1,\ldots,I_n=E$ and $I_1,\ldots,I_n==E$, and we can incorporate multiple and other declarations in the scheme by such equations as $\mathcal{D}[\![\Delta_0 \textbf{ within } \Delta_1]\!]=\lambda\rho.(\lambda\rho_0.\mathcal{D}[\![\Delta_1]\!](divert\rho\rho_0))*\mathcal{D}[\![\Delta_0]\!]\rho$ and $\mathcal{T}[\![\Delta_0 \textbf{ within } \Delta_1]\!]=\lambda\rho.(\lambda\rho_0.\mathcal{T}[\![\Delta_1]\!](divert\rho\rho_0))*\mathcal{D}[\![\Delta_0]\!]\rho$. To illus-

trate the technique we examine rec Δ, taking ψ and ρ to be any entities such that $crushed[\![\text{rec } \Delta]\!]\psi = true$ and $\psi[\![I]\!]\downarrow 1 = 2$ whenever $\rho[\![I]\!]\downarrow 1 : G$. Given any proper σ we take α^* to be $news(\#\mathscr{I}[\![\Delta]\!])\sigma$; if α^* is improper $\lambda\chi.\mathscr{D}[\![\text{rec } \Delta]\!]\rho\chi\sigma$ is improper whilst otherwise $\lambda\chi.\mathscr{D}[\![\text{rec } \Delta]\!]\rho\chi\sigma = \lambda\chi.\mathscr{T}[\![\Delta]\!]\rho_0\chi\sigma_0$ where $\rho_0 = fix(\lambda\rho'.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho'\sigma_0/\mathscr{H}[\![\Delta]\!]])$ and $\sigma_0 = updates\alpha^* dummy^*\sigma$. Because ρ_0 and σ_0 are proper and $crushed[\![\Delta]\!]\psi[0^*/\mathscr{I}[\![\Delta]\!]][2^*/\mathscr{H}[\![\Delta]\!]] = true$, either $\lambda\chi.\mathscr{T}[\![\Delta]\!]\rho_0\chi\sigma_0$ is improper or $\lambda\chi.\mathscr{T}[\![\Delta]\!]\rho_0\chi\sigma_0 = \lambda\chi.(\chi^*\neg\mathscr{T}[\![\Delta]\!]\rho_0)\sigma_0$. Hence either $\lambda\chi.\mathscr{D}[\![\text{rec } \Delta]\!]\rho\chi\sigma$ is improper or $\lambda\chi.\mathscr{D}[\![\text{rec } \Delta]\!]\rho\chi\sigma = \lambda\chi.(\chi^*\neg\mathscr{D}[\![\text{rec } \Delta]\!]\rho)\sigma$ where $\neg\mathscr{D}[\![\text{rec } \Delta]\!]\rho\sigma$ is defined to be

$$(\lambda\alpha^*.(\lambda\sigma'.\alpha^*:E\to\neg\mathscr{T}[\![\Delta]\!](fix(\lambda\rho'.\rho'[\alpha^*/\mathscr{I}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho'\sigma'/\mathscr{H}[\![\Delta]\!]]))\sigma',\tau)$$
$$(updates\alpha^* dummy^*\sigma))(news(\#\mathscr{I}[\![\Delta]\!])\sigma).$$

Similarly we can take $\neg\mathscr{T}[\![\text{rec } \Delta]\!]\rho\sigma$ to be $\neg\mathscr{T}[\![\Delta]\!]\rho\sigma$.▷

In fact there is even a minor gloss on this result: unless $\lambda\kappa.\mathscr{E}[\![E]\!]\rho\kappa\sigma$ is τ it must be $\lambda\kappa.(\kappa*\neg\mathscr{E}[\![E]\!]\rho)\sigma$ (if $crushed[\![E]\!]\psi = true$ and $\psi[\![I]\!]\downarrow 1 = 2$ whenever $\rho[\![I]\!]\downarrow 1 : G$) whether or not σ is proper. Should while E_0 do E_1 be omitted from the list of expressions permitted by *crushed* there would be a further extension, as then for every suitable ρ and σ (and for all E in this abbreviated list) $\lambda\kappa.\mathscr{E}[\![E]\!]\rho\kappa\sigma = \neg\mathscr{E}[\![E]\!]\rho\sigma:[E\times S]\to\lambda\kappa.(\kappa*\neg\mathscr{E}[\![E]\!]\rho)\sigma,\tau$. This extension requires the omission of while loops because standard semantics and its conjugate deal differently with unending programs: in standard semantics a potentially infinite program which encounters an error may yield τ for an answer whereas its conjugate equations result in $\perp$, as in effect it 'continues after going wrong'. Thus if, say, $\mathscr{W}[\![\Omega]\!] = \lambda(\beta_0,\beta_1).-2^{32}<\beta_0+\beta_1<2^{32}\to\beta_0+\beta_1,\tau$ for some Ω:Dya, then *n*=0 inside while true do *n*:=*n*+1 will provide the answer τ when evaluated using $\mathscr{D}$ and the answer $\perp$ when evaluated using $\neg\mathscr{D}$ unless $\perp:[S\to[E\times S]]^\circ$ is presumed to satisfy $\perp\tau = \tau$.

1.5.6. Proposition.

For every ψ, ρ, σ and E:Exp such that σ is proper and $crushed[\![E]\!]\psi=true$ $\lambda\kappa.\mathscr{E}[\![E]\!]\perp\kappa\sigma\sqsubseteq\lambda\kappa.(\kappa\star\neg\mathscr{E}[\![E]\!]\rho)\sigma$ unless $\lambda\kappa.\mathscr{E}[\![E]\!]\perp\kappa\sigma=\tau$; similar results apply to the other valuations, so for every ψ, ρ, σ and Δ:Dec such that σ is proper and $crushed[\![\Delta]\!]\psi=true$ $\lambda\chi.\mathscr{T}[\![\Delta]\!]\perp\chi\sigma\sqsubseteq\lambda\chi.(\chi\star\neg\mathscr{T}[\![\Delta]\!]\rho)\sigma$ unless $\lambda\chi.\mathscr{T}[\![\Delta]\!]\perp\chi\sigma=\tau$.

⊲The proof of this follows the lines drawn up in 1.5.5 by using the fact that

$\lambda\kappa.\mathscr{G}[\![I]\!]\perp\kappa\sigma=\perp\sqsubseteq\lambda\kappa.\kappa\perp\perp=\lambda\kappa.(\kappa\star\neg\mathscr{G}[\![I]\!]\perp)\sigma\sqsubseteq\lambda\kappa.(\kappa\star\neg\mathscr{G}[\![I]\!]\rho)\sigma$

as the basis of a structural induction.⊳

As conjugate equations give rise to elements in $[S\to[E\times S]]^\circ$ rather than G it is natural to expect that the version of recursion involved in them can be rewritten to obviate the need for the latter kind of element. As a preliminary to this we show that members of E can sometimes be substituted in the environment for members of G.

1.5.7. Lemma.

Suppose that for some I*:Ide*, some ε*:E* with $\#\varepsilon^*=\#I^*$ and some ρ_0:U we define ρ_1 and ρ_2 to be $\rho_0[\varepsilon^*/I^*]$ and $\rho_1[(\ldots,\lambda\kappa.\kappa(\varepsilon^*\downarrow\nu),\ldots)/I^*]$ respectively. For every E:Exp $\mathscr{E}[\![E]\!]\rho_1=\mathscr{E}[\![E]\!]\rho_2$ and, if $crushed[\![E]\!]\psi=true$ for some ψ having $\psi[\![I]\!]\downarrow1=2$ whenever $\rho_2[\![I]\!]\downarrow1$:G, $\neg\mathscr{E}[\![E]\!]\rho_1=\neg\mathscr{E}[\![E]\!]\rho_2$; analogous conclusions hold for $\mathscr{L},\mathscr{R}$ and $\mathscr{G}$ also. For every Δ:Dec $\mathscr{D}[\![\Delta]\!]\rho_1=\mathscr{D}[\![\Delta]\!]\rho_2$ and, if $crushed[\![\Delta]\!]\psi=true$ for some ψ having $\psi[\![I]\!]\downarrow1=2$ whenever $\rho_2[\![I]\!]\downarrow1$:G, $\neg\mathscr{D}[\![\Delta]\!]\rho_1=\neg\mathscr{D}[\![\Delta]\!]\rho_2$; analogous conclusions hold for $\mathscr{T}$ also.

⊲When $1\le\nu\le\#I^*$ $\mathscr{G}[\![I^*\downarrow\nu]\!]\rho_1=\lambda\kappa.\kappa(\varepsilon^*\downarrow\nu')=\mathscr{G}[\![I^*\downarrow\nu]\!]\rho_2$ for some ν'. Moreover, when I is not a member of I* $\mathscr{G}[\![I]\!]\rho_1=\mathscr{G}[\![I]\!]\rho_0=\mathscr{G}[\![I]\!]\rho_2$ and $\neg\mathscr{G}[\![I]\!]\rho_1=\neg\mathscr{G}[\![I]\!]\rho_0=\neg\mathscr{G}[\![I]\!]\rho_2$. Now we use

structural induction to complete the proof; for instance if $\mathscr{L}[\![E]\!]\rho_1=\mathscr{L}[\![E]\!]\rho_2$ for all such ρ_1 and ρ_2, $\mathscr{F}[\![\mathrm{fn}\ I.E]\!]\rho_1=\lambda\varepsilon.\mathscr{L}[\![E]\!]\rho_1[\varepsilon/I]=\lambda\varepsilon.\mathscr{L}[\![E]\!]\rho_2[\varepsilon/I]=\mathscr{F}[\![\mathrm{fn}\ I.E]\!]\rho_2$, $\mathscr{G}[\![\mathrm{fn}\ I.E]\!]\rho_1=\mathscr{G}[\![\mathrm{fn}\ I.E]\!]\rho_2$ and $\P\mathscr{G}[\![\mathrm{fn}\ I.E]\!]\rho_1=\P\mathscr{G}[\![\mathrm{fn}\ I.E]\!]\rho_2$.⊳

For any Δ:Dec satisfying $crushed[\![\Delta]\!]\psi=true$ for some ψ:Ide↦B* we now take $\P\mathscr{S}[\![\Delta]\!]$ to be $\lambda\rho\sigma.fix(\lambda\phi I^*.I^*=\langle\rangle\to\langle\rangle,\langle(\P\mathscr{T}[\![\Delta]\!]\rho\sigma\downarrow 1)[\![I^*\downarrow 1]\!]\downarrow 1\rangle\S\phi(I^*\dagger 1))(\mathscr{K}[\![\Delta]\!])$, so that $\P\mathscr{S}$ is a member of Dec→U→S→E* which lists the effects of the elements in G set up by a recursive declaration. If ρ and σ are such that σ is proper and $\psi[\![I]\!]\downarrow 1=2$ whenever $\rho[\![I]\!]\downarrow 1$:G either $\lambda\chi.\mathscr{T}[\![\Delta]\!]\rho\chi\sigma$ is improper or $\lambda\chi.\mathscr{T}[\![\Delta]\!]\rho\chi\sigma=\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!]\rho)\sigma$ and

$\mathscr{S}[\![\Delta]\!]\rho\sigma\downarrow\nu=\lambda\kappa\sigma''.\mathscr{T}[\![\Delta]\!]\rho(\lambda\rho'\sigma'.\kappa(\rho'[\![\mathscr{K}[\![\Delta]\!]\downarrow\nu]\!]\downarrow 1|E)\sigma'')\sigma$

$=\lambda\kappa\sigma''.((\lambda\rho'\sigma'.\kappa(\rho'[\![\mathscr{K}[\![\Delta]\!]\downarrow\nu]\!]\downarrow 1|E)\sigma'')*\P\mathscr{T}[\![\Delta]\!]\rho)\sigma$

$=\lambda\kappa.\kappa(\P\mathscr{S}[\![\Delta]\!]\rho\sigma\downarrow\nu)$

when $1\le\nu\le\#\mathscr{K}[\![\Delta]\!]$.

1.5.8. Proposition.

Let ψ, ρ, σ and Δ:Dec be such that σ is proper, $crushed[\![\mathrm{rec}\ \Delta]\!]\psi=true$ and for all I:Ide $\psi[\![I]\!]\downarrow 1=2$ if $\rho[\![I]\!]\downarrow 1$:G. Unless $\lambda\chi.\mathscr{D}[\![\mathrm{rec}\ \Delta]\!]\rho\chi\sigma$ is improper, $\P\mathscr{T}[\![\Delta]\!](fix(\lambda\rho'.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho'\sigma'/\mathscr{K}[\![\Delta]\!]]))$ is equal to $\P\mathscr{T}[\![\Delta]\!](fix(\lambda\rho'.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]][\P\mathscr{S}[\![\Delta]\!]\rho'\sigma'/\mathscr{K}[\![\Delta]\!]]))$ while $\lambda\chi.\mathscr{D}[\![\mathrm{rec}\ \Delta]\!]\rho\chi\sigma$ is equal to $\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!](fix(\lambda\rho'.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]][\P\mathscr{S}[\![\Delta]\!]\rho'\sigma'/\mathscr{K}[\![\Delta]\!]])))\sigma'$ where $\alpha^*=news(\#\mathscr{I}[\![\Delta]\!])\sigma$ and $\sigma'=updates\alpha^*dummy^*\sigma$.

⊲Take any proper ψ, ρ and σ such that $crushed[\![\mathrm{rec}\ \Delta]\!]\psi=true$ and $\psi[\![I]\!]\downarrow 1=2$ whenever $\rho[\![I]\!]\downarrow 1$:G. Suppose that $\lambda\chi.\mathscr{D}[\![\mathrm{rec}\ \Delta]\!]\rho\chi\sigma$ is proper and that $\#\mathscr{K}[\![\Delta]\!]>0$, the outcome being obvious otherwise. Define $\alpha^*=news(\#\mathscr{I}[\![\Delta]\!])\sigma$, $\sigma_0=updates\alpha^*dummy^*\sigma$ and $\rho_0=\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]]$; then α^* is proper, and if we set $\rho_1=fix(\lambda\rho'.\rho_0[\mathscr{S}[\![\Delta]\!]\rho'\sigma_0/\mathscr{K}[\![\Delta]\!]])$

$\lambda\chi.\mathscr{D}[\![\,\mathbf{rec}\ \Delta]\!]\rho\chi\sigma=\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!]\rho_1)\sigma_0$. Writing $fun=\lambda\nu.\nu=0\to\perp,\rho_0[\mathscr{S}[\![\Delta]\!](fun(\nu-1))\sigma_0/\mathscr{H}[\![\Delta]\!]]$ and $joy=\lambda\nu.\nu=0\to\perp,\rho_0[\P\mathscr{S}[\![\Delta]\!](joy(\nu-1))\sigma_0/\mathscr{H}[\![\Delta]\!]]$, we have $\rho_1=\bigsqcup\{fun\nu\mid\nu:\mathbf{N}\}$, and we need show only that $\P\mathscr{T}[\![\Delta]\!]\rho_1=\P\mathscr{T}[\![\Delta]\!]\rho_2$ where $\rho_2=\bigsqcup\{joy\nu\mid\nu:\mathbf{N}\}$, as then $\lambda\chi.\mathscr{D}[\![\,\mathbf{rec}\ \Delta]\!]\rho\chi\sigma=\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!]\rho_1)\sigma_0=\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!]\rho_2)\sigma_0$.

Observe that as $\lambda\chi.\mathscr{T}[\![\Delta]\!]\rho_1\chi\sigma_0$ is proper the proof of 1.5.5 allows us to infer that $\P\mathscr{T}[\![\Delta]\!]\rho'\sigma_0$ is proper and that $\lambda\chi.\mathscr{T}[\![\Delta]\!]\rho'\chi\sigma_0=\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!]\rho')\sigma_0$ whenever ρ' is proper and included in ρ_1; essentially this is so because *true* and *false* are incomparable and the choice of branch in **if** E_0 **then** E_1 **else** E_2 cannot be influenced by whether the environment is ρ' or ρ_1. In particular, for all $\nu\geq1$ $\lambda\chi.\mathscr{T}[\![\Delta]\!](fun\nu)\chi\sigma_0=\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!](fun\nu))\sigma_0$ and $\mathscr{S}[\![\Delta]\!](fun\nu)\sigma_0\downarrow\nu'=\lambda\kappa.\kappa(\P\mathscr{S}[\![\Delta]\!](fun\nu)\sigma_0\downarrow\nu')$ when $1\leq\nu'\leq\#\mathscr{H}[\![\Delta]\!]$. It also transpires from 1.5.6 that $\lambda\chi\mathscr{T}[\![\Delta]\!]\perp\chi\sigma_0\sqsubseteq\lambda\chi.(\chi*\P\mathscr{T}[\![\Delta]\!]\perp)\sigma_0$ and that $\mathscr{S}[\![\Delta]\!]\perp\sigma_0\downarrow\nu'\sqsubseteq\lambda\kappa.\kappa(\P\mathscr{S}[\![\Delta]\!]\perp\sigma_0\downarrow\nu')$ when $1\leq\nu'\leq\#\mathscr{H}[\![\Delta]\!]$; by 1.5.7, therefore, $\P\mathscr{T}[\![\Delta]\!](joy0)\sqsubseteq\P\mathscr{T}[\![\Delta]\!](fun1)\sqsubseteq\P\mathscr{T}[\![\Delta]\!](joy1)$.

Assume that for some $\nu\geq0$

$$\P\mathscr{T}[\![\Delta]\!](joy\nu)\sqsubseteq\P\mathscr{T}[\![\Delta]\!](fun(\nu+1))\sqsubseteq\P\mathscr{T}[\![\Delta]\!](joy(\nu+1)).$$

Then by 1.5.7 again

$$\begin{aligned}\P\mathscr{T}[\![\Delta]\!](joy(\nu+1))&=\P\mathscr{T}[\![\Delta]\!]\rho_0[\P\mathscr{S}[\![\Delta]\!](joy\nu)\sigma_0/\mathscr{H}[\![\Delta]\!]]\\&\sqsubseteq\P\mathscr{T}[\![\Delta]\!]\rho_0[\P\mathscr{S}[\![\Delta]\!](fun(\nu+1))\sigma_0/\mathscr{H}[\![\Delta]\!]]\\&=\P\mathscr{T}[\![\Delta]\!]\rho_0[\langle\ldots,\lambda\kappa.\kappa(\P\mathscr{S}[\![\Delta]\!](fun(\nu+1))\sigma_0\downarrow\nu'),\ldots\rangle/\mathscr{H}[\![\Delta]\!]]\\&=\P\mathscr{T}[\![\Delta]\!]\rho_0[\mathscr{S}[\![\Delta]\!](fun(\nu+1))\sigma_0/\mathscr{H}[\![\Delta]\!]]\\&=\P\mathscr{T}[\![\Delta]\!](fun(\nu+2))\end{aligned}$$

whilst from this

$$\begin{aligned}\P\mathscr{T}[\![\Delta]\!](fun(\nu+2))&=\P\mathscr{T}[\![\Delta]\!]\rho_0[\P\mathscr{S}[\![\Delta]\!](fun(\nu+1))\sigma_0/\mathscr{H}[\![\Delta]\!]]\\&\sqsubseteq\P\mathscr{T}[\![\Delta]\!]\rho_0[\P\mathscr{S}[\![\Delta]\!](joy(\nu+1))\sigma_0/\mathscr{H}[\![\Delta]\!]]\\&=\P\mathscr{T}[\![\Delta]\!](joy(\nu+2)).\end{aligned}$$

Hence for all $\nu \geq 0$

$\mathcal{T}[\![\Delta]\!](joy\nu) \sqsubseteq \mathcal{T}[\![\Delta]\!](fun(\nu+1)) \sqsubseteq \mathcal{T}[\![\Delta]\!](joy(\nu+1))$ and by continuity $\mathcal{T}[\![\Delta]\!]\rho_2 \sqsubseteq \mathcal{T}[\![\Delta]\!]\rho_1 \sqsubseteq \mathcal{T}[\![\Delta]\!]\rho_2$, which establishes the result.⊳

We can now confirm that the recursion operator of 1.4.4 has the effect one would expect when applied to abstractions.

1.5.9. Corollary.

For any abstractions $\Phi_1,\ldots,\Phi_n$, any distinct identifiers $I_1,\ldots,I_n$, any $\rho:U$ and any $\sigma:S$

$\lambda\chi.(\lambda\rho'.\chi(arid[\mathcal{F}[\![\Phi_1]\!]\rho'/I_1]\ldots[\mathcal{F}[\![\Phi_n]\!]\rho'/I_n])\sigma)$

$\quad(fix(\lambda\rho''.\rho[\mathcal{F}[\![\Phi_1]\!]\rho''/I_1]\ldots[\mathcal{F}[\![\Phi_n]\!]\rho''/I_n]))$

is $\lambda\chi.\mathcal{D}[\![\text{rec } \Delta]\!]\rho\chi\sigma$ where Δ is $I_1 == \Phi_1$ and...and $I_n == \Phi_n$.

⊲Observe that for any ρ'

$\mathcal{T}[\![\Delta]\!]\rho'\sigma = \langle arid[\mathcal{F}[\![\Phi_1]\!]\rho'/I_1]\ldots[\mathcal{F}[\![\Phi_n]\!]\rho'/I_n], \sigma\rangle$, so, σ being proper, $\mathcal{S}[\![\Delta]\!]\rho'\sigma = \langle \mathcal{F}[\![\Phi_1]\!]\rho', \ldots, \mathcal{F}[\![\Phi_n]\!]\rho'\rangle$. As $crushed[\![\Delta]\!]\psi = true$ we may apply 1.5.8 immediately to give the result. A direct proof using the techniques above even shows that the result holds whether or not ρ and σ are proper.⊳

It is to this equation for recursive procedures that methods for validating algorithms [1] most usually apply. However we wish to establish not that particular programs accord with what their writers intend but that once written a program will be executed properly. Consequently we must descend from these empyrean heights of abstraction and discuss mathematical models for implementations.

CHAPTER TWO

STORE SEMANTICS

2.1. State vectors.

2.1.1. Abstract closures.

The equations for rec I==Φ and rec I=Φ which are entailed by appendix 1 cannot yield the same results for all values of ρ, χ and σ, as only one involves adding a location to the available storage. However unless this location is assigned to it should not affect the outcome of applying the function Φ except by providing a means whereby the corresponding closure can refer to itself. To dispel the haze surrounding the link between these equations the location and its content must therefore be related to a closure kept in an environment; doing this should also clarify the sense in which I::E and I:E are equivalent.

Thus in order to connect $\kappa(fix(\lambda\phi.\mathscr{F}[\![\Phi]\!]\rho[\phi/I]))\sigma$ with $(\lambda\alpha.\kappa\alpha(update\alpha(\mathscr{F}[\![\Phi]\!]\rho[\alpha/I])\sigma))(new\sigma)$ we are obliged to compare a closure having itself in its free variable list with one having a location instead. Whilom closures were regarded as members of $[E\to K\to C]^\circ$, so that their free variable lists did not explicitly appear in the formal semantics; now we split them up to enable us to tell which locations may be referred to during the application of a function. Because these lists are essentially little environments we take the domain of function closures, F, to be $0^\circ\times U$, where certain pairs $\langle \xi,\rho\rangle : 0^\circ\times U$ are such that the entity $\xi\rho$ performs the same task as some $\phi:[E\to K\to C]^\circ$. The lattice 0 is separated from U in the product to avoid identifying $\perp$ in the domain V with an unending computation; it does not matter whether or not the component U is separated (as 2.2.7 will confirm) but since the environment $\perp$ arises only when forming a fixed point $0^\circ\times U$ is perhaps intuitively more satisfactory than $0^\circ\times U^\circ$.

Label entry points must be dissected similarly before the expressions I::E and I:E can be related. By analogy with closures we might expect an entry point to be in the domain $Z^{\circ}\times U$, where as no environment is subsumed under the first component $\zeta:Z$ consists of code after compilation and before loading. This model is accurate for badly-designed languages but it is inadequate for Pal, in which the standard continuation conceals more of the state than this reveals. Because labels can be assigned, control can return to an expression in the program some time after leaving it. If this happens not merely are the identifiers given their original meanings but the anonymous local variables reappear: jumping back into 1+(*l*:*m*:=*l*; 2), for instance, provides the answer 3. Thus as part of the value of an entry point we must keep all the anonymous quantities which could be needed on returning to the expression. In principle all the quantities created since the program began can be required, so if Y is the lattice of stacks (lists of unnamed variables) the transformation representing a portion of code must take a member of Y as an argument. Hence Z is $U\to Y\to S\to A$, J is $Z^{\circ}\times U\times Y$ and, as every nameless variable results from an expression, Y is E^{*}.

As hinted above, proving the equivalence of semantic equations can involve examining triples of the form $\langle\rho,\upsilon,\sigma\rangle$ having $\rho:U$, $\upsilon:Y$ and $\sigma:S$ wherein denoted closures can be made to tally with stored ones. To aid us we allow ρ, υ and σ to be passed as arguments by the program to the continuation $\zeta:Z$; now the formal description language resembles the interpreter of Landin [10] reconstituted as a compiler, having such valuations as $\mathscr{E}:Exp\to Z\to U\to Y\to S\to A$ and $\mathscr{E}:Exp\to Z\to U\to Y\to S\to A$. From these the values of $\mathscr{P}[\![I:E]\!]$ and $\mathscr{Q}[\![I::E]\!]$ may be derived as $\lambda\zeta\rho\upsilon.\langle\langle\mathscr{E}[\![E]\!],\rho,\upsilon\rangle\rangle\,\S\mathscr{P}[\![E]\!]\zeta\rho\upsilon$ and $\lambda\zeta\rho\upsilon.\langle\langle\mathscr{E}[\![E]\!],\rho,\upsilon\rangle\rangle\,\S\mathscr{Q}[\![E]\!]\zeta\rho\upsilon$ respectively. The other valuations are also given their earlier significance but different arguments, so

that $\mathscr{F}[\![\text{fn}()E]\!]$, for instance, is

$\lambda\rho.\langle sv\circ(\lambda\zeta'\rho'\upsilon'\sigma'.\upsilon'\downarrow 1\,|\,L^*=\langle\rangle\rightarrow\mathscr{G}[\![E]\!]\zeta'\rho'(\upsilon'\dagger 1)\sigma',\tau),rend[\![\text{fn}()E]\!]\rho\rangle$.

Removing the environment and the stack from the continuation enables Z to be substituted for K and X as well as C, so that now $\rho[\![\text{res}]\!]$ is in J*. Because the equations set up by this process simulate an implementation in which complex parts of the state vector can be stored by assignment the formulation of Mal in terms of them will be called its 'store semantics'.

To record all the locations which may be referred to while computing we must subject G to the same scrutiny as F and J. One possible reconstruction of it is $0^\circ\times U\times Y\times S$, for which $\mathscr{S}[\![\Delta]\!]\rho'\upsilon'\sigma'\downarrow\nu$ is $\langle\mathscr{F}[\![\Delta]\!],\rho',\upsilon',\sigma'\rangle$ and $\mathscr{G}[\![I]\!]\zeta\rho\upsilon\sigma$ is

$\xi(\lambda\rho''\upsilon''\sigma''.\zeta\rho((\rho''[\![I]\!]\downarrow 1\,|\,E)\,\S\upsilon)\sigma)\rho'\upsilon'\sigma'$ when $\rho[\![I]\!]\downarrow 1$ is $\langle\xi,\rho',\upsilon',\sigma'\rangle$, but this involves an unnecessary degree of dependence on the entire state at the time of the recursive declaration. The sole semantic equation to require a knowledge of the stack supplied as an argument is that for labelled blocks which does so because jumps need it. In our case, however, the continuation is bestowed when I is looked up in the environment, not when I is declared, and it carries with it the existing stack. Hence the third component of G above is as redundant as it would be in our domain of function closures, and we can take G to be $0^\circ\times U\times S$, $\mathscr{S}[\![\Delta]\!]\rho'\sigma'\downarrow\nu$ to be $\langle\mathscr{F}[\![\Delta]\!],\rho',\sigma'\rangle$ and $\mathscr{G}[\![I]\!]\zeta\rho\upsilon\sigma$ to be $\xi(\lambda\rho''\upsilon''\sigma''.\zeta\rho((\rho''[\![I]\!]\downarrow 1\,|\,E)\,\S\upsilon)\sigma)\rho'\langle\rangle\sigma'$ when $\rho[\![I]\!]\downarrow 1$ is $\langle\xi,\rho',\sigma'\rangle$. It will be established in 2.3.8 that these prescriptions together with variants of those in appendix 2 are indeed precisely equivalent to the standard ones.

The equations given in appendix 1 can imitate a storage allocation scheme which releases the storage requisitioned by a block on exit from it but not one which collects the currently inaccessible locations. Now that the environment and the stack have been separated from the code we can define a function

site:L→U→Y→S→T such that *site*αρυσ is *true* if and only if α is the final link in a chain of witnessed values from ⟨ρ,υ,σ⟩ each having the next as a constituent. A formalization of this must be left to 2.1.6; here all we need is the existence of a monotonic function, *novel*:U→Y→S→L, which is constrained by λρυσ.*site*(*novel*ρυσ)ρυσ=λρυσ.⋀{*site*αρυσ|α:L}→⊥,*false*. Keeping the entire environment instead of the current one ensures that choosing a fresh location in this way will not overwrite one required on returning to an outer block.

Garbage collection is merely a way of arranging to obtain locations which cannot be reached from the present state vector. Its essence is independent of particular marking and compaction algorithms, being captured in our notation by *novels*:N→U→Y→S→L*:

novels=λνρυσ.ν=0→⟨⟩,

(λα.⟨α⟩ §*novels*(ν-1)ρ(⟨α⟩ §υ)(*update*α*dummy*σ))(*novel*ρυσ).

We require also *mv*:0 and *sv*:0, which are given by

mv=λζρυσ.υ↓1:L→ζρυσ,

(λα.α:L→ζρ(⟨α⟩ §υ↑1)(*update*α(υ↓1)σ),τ)(*novel*ρυσ);

sv=λζρυσ.υ↓1:L→(*area*(υ↓1)σ→ζρ(⟨*hold*(υ↓1)σ⟩ §υ↑1)σ,τ),ζρυσ.

As the construction of S from V does not depend on choosing standard semantics instead of the present variety we shall take across the definitions in 1.3.1 to our later equations.

Unfortunately if we use *novel* rather than *new* in the semantics of Mal the recursion operator above will cause trouble. The evaluation of 𝒮⟦I⟧ζρυσ when ρ⟦I⟧↓1=⟨ξ,ρ′,σ′⟩ might require adding an inaccessible location to the region of store; for this *novel*ρ′⟨⟩σ′ would be used. At the time of declaration, however, the stack υ′ might not have been empty and so *novel*ρ′υ′σ′ would have been used instead. Since there is no reason to suppose that these locations are identical in such circumstances as those of 2.1.2 this operator differs in its effect from that of 1.3.1.

2.1.2. Example.

Let E_0 be x=1 inside rec f==E_1 inside E_2, E_1 be nil aug fn$z$.fz and E_2 be nil aug $(fx)x$=x. Then the location returned by E_0 may contain either *true* or *false* if the recursion operator above is used in conjunction with *novel*.

This example makes use of the confusion in Pal between selecting components from lists in L* and applying functions: fx represents the former and $(fx)x$ the latter. It also assumes that members of L* can be tested for equality by a program.

Take any proper ρ_0, υ_0 and σ_0, together with some ζ_0:Z°. Define α_0=*novel*$\rho_0(\langle 1\rangle \S\upsilon_0)\sigma_0$, σ_1=*update*$\alpha_0 1\sigma_0$, $\rho_1=\rho_0[\alpha_0/x]$, $\xi_0=\mathscr{R}[\![E_1]\!]\circ(\lambda\zeta\rho\upsilon.\zeta(invert\rho(arid[\upsilon\downarrow 1/f]))(\upsilon\uparrow 1))$, $\zeta_1=\zeta_0\circ revert\rho_0$, $\rho_2=fix(\lambda\rho.\rho_1[\langle\xi_0,\rho,\sigma_1\rangle/f])$ and $\xi_1=\lambda\zeta\rho\upsilon.\mathscr{L}[\![fz]\!]\zeta\rho[\upsilon\downarrow 1/z](\upsilon\uparrow 1)$.

$$\begin{aligned}
\mathscr{G}[\![E_0]\!]\zeta_0\rho_0\upsilon_0\sigma_0 &= \mathscr{L}[\![\text{rec } f==E_1 \text{ inside } E_2]\!]\zeta_1\rho_1\upsilon_0\sigma_1\\
&= \xi_0(\mathscr{L}[\![E_2]\!]\zeta_1)\rho_2\upsilon_0\sigma_1\\
&= mv(\lambda\rho\upsilon.\mathscr{L}[\![E_2]\!]\zeta_1\rho_1[\langle\upsilon\downarrow 1\rangle/f](\upsilon\uparrow 1))\rho_2(\langle\langle\xi_1,\rho_2\rangle\rangle\S\upsilon_0)\sigma_1\\
&= \mathscr{L}[\![E_2]\!]\zeta_1\rho_1[\langle\alpha_1\rangle/f]\upsilon_0\sigma_2.
\end{aligned}$$

Here α_1=*novel*$\rho_2(\langle\langle\xi_1,\rho_2\rangle\rangle\S\upsilon_0)\sigma_1$ and σ_2=*update*$\alpha_1\langle\xi_1,\rho_2\rangle\sigma_1$. Now set $\rho_3=\rho_1[\langle\alpha_1\rangle/f]$, α_2=*novel*$\rho_2\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_1$, σ_3=*update*$\alpha_2\langle\xi_1,\rho_2\rangle\sigma_1$, $\zeta_2=\lambda\rho\upsilon.\mathscr{R}[\![f]\!](\lambda\rho\upsilon.mv\zeta_1\rho(\langle\upsilon\downarrow 1=\upsilon\downarrow 2\rangle\S\upsilon\uparrow 2))\rho(\langle\langle\upsilon\downarrow 1|L^*\rangle\rangle\S\upsilon\uparrow 1)$ and $\zeta_3=\lambda\rho\upsilon\sigma.\zeta_2\rho_3(\langle\rho[\![f]\!]\downarrow 1|L^*\downarrow 1\rangle\S\upsilon_0)\sigma_2$.

$$\begin{aligned}
\mathscr{G}[\![E_0]\!]\zeta_0\rho_0\upsilon_0\sigma_0 &= \mathscr{L}[\![(fx)x]\!]\zeta_2\rho_3\upsilon_0\sigma_2\\
&= \xi_1(\zeta_2\circ revert\rho_3)(divert\rho_3\rho_1)(\langle\alpha_0\rangle\S\upsilon_0)\sigma_2\\
&= \mathscr{L}[\![fz]\!](\zeta_2\circ revert\rho_3)(divert\rho_3\rho_1[\alpha_0/z])\upsilon_0\sigma_2\\
&= \xi_0\zeta_3\rho_2\langle\rangle\sigma_1\\
&= mv(\lambda\rho\upsilon.\zeta_3\rho_1[\langle\upsilon\downarrow 1\rangle/f](\upsilon\uparrow 1))\rho_2\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_1\\
&= \zeta_3\rho_1[\langle\alpha_2\rangle/f]\upsilon_0\sigma_3\\
&= \zeta_2\rho_3(\langle\alpha_2\rangle\S\upsilon_0)\sigma_2\\
&= \mathscr{R}[\![f]\!](\lambda\rho\upsilon.mv\zeta_1\rho(\langle\upsilon\downarrow 1=\upsilon\downarrow 2\rangle\rangle\S\upsilon\uparrow 2))\rho_3(\langle\langle\alpha_2\rangle\rangle\S\upsilon_0)\sigma_2\\
&= mv\zeta_1\rho_3(\langle\langle\alpha_1\rangle=\langle\alpha_2\rangle\rangle\S\upsilon_0)\sigma_2.
\end{aligned}$$

Although $novel\rho_2(\langle\langle\xi_1,\rho_2\rangle\rangle \S \upsilon_0)\sigma_1$ may coincide with $novel\rho_2\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_1$ it need not do so if υ_0 is not $\langle\rangle$. Consequently $\langle\alpha_1\rangle=\langle\alpha_2\rangle$ may be *true* or *false*; when G is $0°\times U\times Y\times S$ or $[K\to C]°$, however, the program invariably produces *true* as its result.➢

We might choose to obviate this situation by keeping the stack at the time of declaration as part of the value of a recursively defined entity. In practice to keep stacks simply for this purpose would be inefficient, so as we are endeavouring to model a potential implementation we shall not do this. Indeed an implementation would not even retain the full environment but would content itself with a free variable list, thereby making matters worse. The next example demonstrates that taking G to be $0°\times U\times Y\times S$ would still leave us in thrall to the vagaries of *novel* because a location required by a declaration could be used elsewhere as well.

2.1.3. Example.

Let E_0 be $g,x=0,1$ inside E_2; E_3, E_1 be nil aug fn$z$.fz, E_2 be nil aug 0=nil aug gx and E_3 be rec f==E_1 inside g:=fx. Then the location returned by E_0 may contain either *true* or *false* if the recursion operator above is used in conjunction with *novel*.

➢The arguments involved in this example are very similar to those of 2.1.2, so we shall only outline them; furthermore we shall take ξ_0, ξ_1 and ζ_1 to be as before. Now, however, we let $\upsilon_0=\langle\rangle$ and $\rho_0=arid$ so that the effect of E_0 does not depend on which of the recursion operators of 2.1.1 is adopted.

Set $\alpha_0=novel\rho_0\langle 0\rangle\sigma_0$, $\alpha_1=novel\rho_0\langle 1\rangle(update\alpha_0 0\sigma_0)$,
$\sigma_1=updates\langle\alpha_0,\alpha_1\rangle\langle 0,1\rangle\sigma_0$, $\rho_1=\rho_0[\alpha_0/g][\alpha_1/x]$,
$\rho_2=fix(\lambda\rho.\rho_1[\langle\xi_0,\rho,\sigma_1\rangle/f]$, $\alpha_2=novel\rho_2\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_1$,
$\sigma_2=update\alpha_2\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_1$, $\rho_3=\rho_2[\langle\alpha_2\rangle/f]$,
$\sigma_3=update\alpha_0\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_2$, $\sigma_4=update(novel\rho_3\langle dummy\rangle\sigma_3)(dummy)\sigma_3$,

$\alpha_3 = novel\rho_1\langle 0\rangle\sigma_4$ and $\sigma_5 = update\alpha_3 0\sigma_4$. As noted above, given any $\zeta_0 : Z^\circ$ we write $\zeta_1 = \zeta_0 \circ revert\rho_0$; moreover we define $\zeta_2 = \lambda\rho\upsilon.mv\zeta_1\rho((\langle\upsilon\downarrow 1\rangle = \upsilon\downarrow 2)\,\S\,\upsilon\uparrow 2)$, $\zeta_3 = \lambda\rho\upsilon\sigma.\zeta_2\rho_1(\langle\rho[\![f]\!]\downarrow 1 \mid L^*\downarrow 1, \langle\alpha_4\rangle\rangle)\sigma_4$ and $\zeta_4 = \lambda\rho\upsilon.\mathscr{A}[\![E_2]\!]\zeta_1(revert\rho_1\rho)(\upsilon\uparrow 1)$.

$$\begin{aligned}
\mathscr{R}[\![E_0]\!]\zeta_0\rho_0\upsilon_0\sigma_0 &= \mathscr{A}[\![E_3;\ E_2]\!]\zeta_1\rho_1\upsilon_0\sigma_1 \\
&= \xi_0(\mathscr{A}[\![g := fx]\!]\zeta_4)\rho_2\upsilon_0\sigma_1 \\
&= mv(\lambda\rho\upsilon.\mathscr{L}[\![g := fx]\!]\zeta_4\rho_1[\langle\upsilon\downarrow 1\rangle/f](\upsilon\uparrow 1))\rho_2\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_1 \\
&= \mathscr{A}[\![g := fx]\!]\zeta_4\rho_3\upsilon_0\sigma_2 \\
&= mv\zeta_4\rho_3\langle dummy\rangle\sigma_3 \\
&= \mathscr{A}[\![E_2]\!]\zeta_1\rho_3\upsilon_0\sigma_4 \\
&= mv(\lambda\rho\upsilon.\mathscr{R}[\![gx]\!]\zeta_2\rho(\langle\upsilon\downarrow 1\rangle \mid L^*)\,\S\,\upsilon\uparrow 1))\rho_1\langle 0\rangle\sigma_4 \\
&= \mathscr{R}[\![gx]\!]\zeta_2\rho_1\langle\langle\alpha_3\rangle\rangle\sigma_5.
\end{aligned}$$

Because $site\alpha_0\rho_1\langle 0\rangle\sigma_4 \wedge site\alpha_1\rho_1\langle 0\rangle\sigma_4 = true$, we know that α_3 is neither α_0 nor α_1 and thus that assigning to it does not affect the properties of g and x.

$$\begin{aligned}
\mathscr{R}[\![E_0]\!]\zeta_0\rho_0\upsilon_0\sigma_0 &= \xi_1(\zeta_2\circ revert\rho_1)(divert\rho_1\rho_1)\langle\alpha_1,\langle\alpha_3\rangle\rangle\sigma_5 \\
&= \mathscr{A}[\![fz]\!](\zeta_2\circ revert\rho_1)(divert\rho_1\rho_1[\alpha_1/z])\langle\langle\alpha_3\rangle\rangle\sigma_5 \\
&= \xi_0\zeta_3\rho_2\langle\rangle\sigma_1 \\
&= mv(\lambda\rho\upsilon.\zeta_3\rho_1[\langle\upsilon\downarrow 1\rangle/f](\upsilon\uparrow 1))\rho_2\langle\langle\xi_1,\rho_2\rangle\rangle\sigma_1 \\
&= \zeta_3\rho_3\upsilon_0\sigma_1 \\
&= \zeta_2\rho_1(\langle\rho_3[\![f]\!]\downarrow 1 \mid L^*\downarrow 1, \langle\alpha_4\rangle\rangle)\sigma_4 \\
&= \zeta_2\rho_1\langle\alpha_2,\langle\alpha_3\rangle\rangle\sigma_4 \\
&= mv\zeta_1\rho_1\langle\langle\alpha_2\rangle = \langle\alpha_3\rangle\rangle\sigma_4.
\end{aligned}$$

Since $site\alpha_2\rho_1\langle 0\rangle\sigma_4 = false$ a possible candidate for $novel\rho_1\langle 0\rangle\sigma_4$ is α_2; hence some *novel* functions will give $\langle\alpha_2\rangle = \langle\alpha_3\rangle$ the value *true* while others will give it the value *false*. Had standard semantics been used to evaluate the program α_3 would have been $new\sigma_4$ and $area\alpha_2\sigma_4$ would inevitably have been *true*, so the locations could not have coincided and the program would have returned a location containing *false* as its result.▷

2.1.4. Recurrent program states.

The fact exemplified above, that the novel location chosen during a recursive evaluation may already 'really' be in use, suggests that the semantics of $\mathscr{I}[\![I]\!]\zeta\rho\upsilon\sigma$ be modified so that during a recursive evaluation the state vector $\langle\rho,\upsilon,\sigma\rangle$ to which we finally return an answer is kept as part of the state set up for the evaluation. Thus we take P and U to be $U\times Y\times S$ and $[Ide\to D^{\circ *}]\times J^*\times P^*$ respectively, denote the third component of $\rho:U$ by $\rho[\![rec]\!]$ and extend the conventions and functions of 1.3.2 to $\rho[\![rec]\!]$ by dealing with it in the same way as we dealt with $\rho[\![res]\!]$. By 1.2.7 all the lattices we require remain continuous and all except Z and O are slit.

We might suppose that, when $\rho[\![I]\!]\downarrow 1=\langle\xi,\rho',\sigma'\rangle$ and $\zeta'=\lambda\rho''\upsilon''\sigma''.(\lambda\pi.\zeta(\pi\downarrow 1)(\langle\rho''[\![I]\!]\downarrow 1|E\rangle\S\pi\downarrow 2)(\pi\downarrow 3))(\rho''[\![rec]\!]\downarrow 1)$, $\mathscr{I}[\![I]\!]\zeta\rho\upsilon\sigma$ would be $\xi\zeta'\rho'[\langle\rho,\upsilon,\sigma\rangle/rec]\langle\rangle\sigma'$. Then if *site* were defined as in 2.1.6 any location selected by *novel* during a recursive evaluation could not already be in use and therefore might well not be the location chosen at the time of declaration. For the reasons of efficiency outlined after 2.1.2 we are grudgingly prepared to put up with this; what is intolerable, however, is that should this location be returned as part of $\rho''[\![I]\!]\downarrow 1$ there is no guarantee that it will be in the area of $\rho''[\![rec]\!]\downarrow 1$. In fact this point causes difficulties even in examples akin to 1.4.2, in which we cannot ensure that $area\alpha_2\sigma_2$ is *true*. Accordingly we define

$replace=\lambda\rho\upsilon\sigma_0\sigma_1.\langle\lambda\alpha.plot\alpha\rho\upsilon\sigma_1\to(\sigma_1\downarrow 1)\alpha,(\sigma_0\downarrow 1)\alpha\rangle\S\sigma_1\dagger 1;$

$recur=\lambda\zeta\rho\upsilon\sigma.(\lambda\pi'.\zeta(\pi'\downarrow 1)(\upsilon\S\pi'\downarrow 2)(replace(\pi'\downarrow 1)(\pi'\downarrow 2)(\pi'\downarrow 3)\sigma))$

$(\rho[\![rec]\!]\downarrow 1).$

Here $plot:L\to U\to Y\to S\to T$ indicates which locations are directly accessible through the environment and the stack 'without passing through' members of P (so that if $plot\alpha\rho\upsilon\sigma=true$, $site\alpha\rho\upsilon\sigma=true$

also); its precise definition will be given in 2.1.6. Now $\mathscr{S}[\![\Delta]\!]\rho'\sigma'\downarrow\nu$ becomes $(\mathscr{T}[\![\Delta]\!]\circ(\lambda\zeta''\rho''\upsilon''.recur\zeta''\rho''(\rho''[\![\mathscr{H}[\![\Delta]\!]\downarrow\nu]\!]\downarrow 1\,|\,E),\rho',\sigma')$ and $\mathscr{G}[\![I]\!]\zeta\rho\upsilon\sigma$ reduces to

$(\lambda\delta.\delta:G\rightarrow(\delta\downarrow 1)\zeta((\delta\downarrow 2)[(\rho,\upsilon,\sigma)/\mathbf{rec}])()(\delta\downarrow 3),\zeta\rho((\delta)\S\upsilon)\sigma)(\rho[\![I]\!]\downarrow 1).$

The *novel* store definition of Mal given in appendix 2 is that suggested by these equations subject to one minor alteration. Rather than preserve the entire environment in function closures and in the denotations of recursively declared identifiers (which belong to F and G respectively), we keep a free variable list [22] built up with the aid of the functions *rent* and *torn* introduced in 1.5.1.

The utility of this interpretation of recursion lies in the theorem to be proved in 2.5.9, under which the result of a program is independent of the choice of *novel* function. The equivalence between it and the standard form holds only when the syntax of declarations is restricted in such a way that variants of the proof of 2.6.8 show that all the operators discussed in this section have the same effect.

2.1.5. The conservation of the environment.

Much of the awkwardness in store semantics arises from declarations, which in Mal express a wide range of meanings within a small syntax. Whereas the standard treatment distinguishes sharply between the environment returned as the result of a declaration and that in which the succeeding expression is evaluated, here our wish to model an interpreter which always has the entire state at its behest militates against such a distinction. The need for it arises from the intended meaning of Δ_0 within Δ_1, for on emerging from Δ_1 the values attached to identifiers by Δ_0 will still be lurking around and may obscure parts of the original environment. We could remove these after-effects of Δ_0

by using a function akin to *revert*, but instead we cover them with a layer from the original environment except when Δ_1 has already done so. If the original environment yields only an empty list of denoted values there is nothing with which to conceal these accretions, but nor is there any reason to wish to, as in a correctly composed program no identifier will be used without either being declared explicitly or appearing in the library environment. Thus it is enough to set

$$trim=\lambda\Delta\rho_0\rho_1.\langle\lambda I.(\sim I:\mathscr{F}[\![\Delta]\!]\,\S\mathscr{A}[\![\Delta]\!]\wedge\#\rho_1[\![I]\!]>\#\rho_0[\![I]\!]>0\rightarrow\langle revert\rho_0\rho_1[\![I]\!]\downarrow 1\rangle,\langle\rangle)\,\S\rho_1[\![I]\!]\rangle\,\S\langle\rho_1[\![res]\!]\rangle\,\S\langle\rho_1[\![rec]\!]\rangle.$$

Discarding the superfluous elements from the environment by using $\lambda\Delta\rho_0\rho_1.divert(revert\rho_0\rho_1)(snip[\![\Delta]\!]\rho_1)$ instead of *trim* would in practice be more efficient and would give rise to less complex equations for multiple declarations; furthermore it could be shown to be equivalent to the corresponding notion in standard semantics by the means we shall adopt in any case. We have chosen the more extravagant approach above because it extends readily to the stack semantics of 3.1.1. In this there are no free variable lists attached to closures so any function created in Δ_1 which requires to refer to identifiers declared in Δ_0 must do so by inspecting an omnipresent environment in which their values are preserved.

The equations for declarations thus result in every new environment layer being piled on top of its predecessors, thereby masking them. The constituents $\Delta_1,\ldots,\Delta_n$ of Δ_1 and...and Δ_n, however, should be evaluated in the environment pertaining on entry to the block, which must therefore be retrieved from under whatever flotsam there may be. The primitive intended to carry this out is used even on the set of values associated with

I when I=E is a constituent of rec (Δ_1 and...and Δ_n) because although I=E leaves the environment unchanged preceding declarations may not do so. Accordingly we set

$clip=\lambda\Delta\rho_0\rho_1.\langle\lambda I.\ I:\mathscr{S}[\![\Delta]\!]\,\S\mathscr{H}[\![\Delta]\!]\wedge\#\rho_0[\![I]\!]>0\rightarrow\langle revert\rho_0\rho_1[\![I]\!]\downarrow 1\rangle,\langle\rangle\rangle\,\S\rho_1[\![I]\!]\rangle$
$\S\langle\rho_1[\![res]\!]\rangle\,\S\langle\rho_1[\![rec]\!]\rangle.$

Recovering the desired denoted values after carrying out all of $\Delta_1,\ldots,\Delta_n$ is no easy task, as they may lie under refuse generated by intermediate within declarations. To dredge them up we carry the environment that they create through to the point at which in standard semantics *conserve* would make them operative. Accordingly we take *pick* $[\![\Delta_1$ and...and $\Delta_n]\!]$ to be

$\lambda\rho^*\rho.\langle\lambda I.(I:\mathscr{S}[\![\Delta_{i(n)1}]\!]\,\S\mathscr{H}[\![\Delta_{i(n)1}]\!]\wedge\#(\rho^*\downarrow 1)[\![I]\!]>0\rightarrow\langle revert(\rho^*\downarrow 2)\rho[\![I]\!]\downarrow 2\rangle,$
$\ldots,$
$I:\mathscr{S}[\![\Delta_{i(n)n}]\!]\,\S\mathscr{H}[\![\Delta_{i(n)n}]\!]\wedge\#(\rho^*\downarrow n)[\![I]\!]>0\rightarrow\langle revert(\rho^*\downarrow(n+1))\rho[\![I]\!]\downarrow 2\rangle$
$\langle\rangle\rangle\,\S\rho[\![I]\!]\rangle$
$\S\langle\rho[\![res]\!]\rangle\,\S\langle\rho[\![rec]\!]\rangle.$

Here we have taken over i from 1.3.5, as it is used in store semantics as well as the standard variety to leave an order of evaluation unspecified. In terms of the k of 1.3.5 we define

deal: $0^{\circ *}\rightarrow[U^*\rightarrow Z]\rightarrow Z$ by

$deal=\lambda\xi^*\psi\rho_0\upsilon_0.(\lambda n.(\xi^*\downarrow i(n)1)(\lambda\rho_1.\xi^*\downarrow i(n)2)(\lambda\rho_2.(\xi^*\downarrow i(n)3)(\lambda\rho_3.$
$\ldots(\lambda\rho_n\upsilon_n.\psi\langle\rho_0,\rho_1,\rho_2,\rho_3,\ldots,\rho_n\rangle\rho_n(\#\upsilon_n>\#\upsilon_0\rightarrow k(n)\upsilon_n,\upsilon_n))$
$\ldots)\rho_3)\rho_2)\rho_1))(\#\xi^*)\rho_0\upsilon_0.$

This can be used to set up the analogous function for expressions by writing $mete=\lambda\xi^*\zeta.deal\xi^*(\lambda\rho^*.\zeta)$.

As an environment created by applications of the primitive functions defined above may be of an inordinate size it is useful to have some means of comparing the shapes of two members of U, δ and $\grave{\rho}$, obtained by transforming programs in accordance with the rules of 1.4.6. To this end *neat* is defined by

$neat=\lambda\langle\delta,\grave{\rho}\rangle.\bigwedge\{\#\delta[\![I]\!]=\#\grave{\rho}[\![I]\!]\mid I:Ide\}\wedge(\#\delta[\![res]\!]=\#\grave{\rho}[\![res]\!])\wedge(\#\delta[\![rec]\!]=\#\grave{\rho}[\![rec]\!]).$

2.1.6. Tracing algorithms.

Much of this dissertation is concerned with the properties of pairs of values which arise in the course of computations that are avowedly equivalent. The halves of such pairs will both lie in one domain or, failing this, in two domains which will be designated by identical letters, so that the first half can be represented by the same Greek character as the second. Consequently when dealing with continuations, say, we shall label a typical member of $Z^\circ \times Z^\circ$ by $\hat{\zeta}$ and assume without explicit mention that the first and second components of $\hat{\zeta}$ are $\acute{\zeta}$ and $\grave{\zeta}$ respectively. Analogous rules will apply to all other products of corresponding domains, even when the elements are subscripted; thus $\hat{\varepsilon}_0$, for instance, will be that element of $E^\circ \times E^\circ$ which could equally well be written as $\langle \acute{\varepsilon}_0, \grave{\varepsilon}_0 \rangle$.

Of particular importance in store semantics is the lattice P of state vectors each comprising an environment, a stack and a store. As a typical state vector is denoted by π whereas typical members of U, Y and S are denoted by ρ, υ and σ respectively, henceforth it will be convenient to identify π with $\langle \rho, \upsilon, \sigma \rangle$, thereby giving $\rho = \pi\downarrow 1$, $\upsilon = \pi\downarrow 2$ and $\sigma = \pi\downarrow 3$. This convention will extend also to subscripted and accented variables, so that underlying all that follows will be such unstated equalities as $\hat{\pi}_1 = \langle \acute{\pi}_1, \grave{\pi}_1 \rangle = \langle \langle \acute{\rho}_1, \acute{\upsilon}_1, \acute{\sigma}_1 \rangle, \langle \grave{\rho}_1, \grave{\upsilon}_1, \grave{\sigma}_1 \rangle \rangle$. The sole justification for this rebarbative usage is that it will make our notation less prolix than would otherwise be the case.

Our initial application of these conventions will be to devise a procedure for testing values $\acute{\omega}$ and $\grave{\omega}$ to see whether they can be witnessed at parallel points in state vectors $\acute{\pi}$ and $\grave{\pi}$ which have been created differently (by using two kinds of declaration or storage allocation, for example). Amongst these points are the entries in the environments and stacks, which can be lined up by

means of

$hoten=\lambda\hat{\omega}\hat{\rho}.\bigvee\{\bigvee\{1\leq\nu\leq\#\acute{\rho}[\![\mathrm{I}]\!]\wedge 1\leq\nu\leq\#\grave{\rho}[\![\mathrm{I}]\!]\rightarrow\hat{\omega}=\langle\acute{\rho}[\![\mathrm{I}]\!]\downarrow\nu,\grave{\rho}[\![\mathrm{I}]\!]\downarrow\nu\rangle,false\,|\,\mathrm{I}:\mathrm{Ide}\}$

$\vee(1\leq\nu\leq\#\acute{\rho}[\![\mathsf{res}]\!]\wedge 1\leq\nu\leq\#\grave{\rho}[\![\mathsf{res}]\!]\rightarrow\hat{\omega}=\langle\acute{\rho}[\![\mathsf{res}]\!]\downarrow\nu,\grave{\rho}[\![\mathsf{res}]\!]\downarrow\nu\rangle,false)$

$\vee(1\leq\nu\leq\#\acute{\rho}[\![\mathsf{rec}]\!]\wedge 1\leq\nu\leq\#\grave{\rho}[\![\mathsf{rec}]\!]\rightarrow\hat{\omega}=\langle\acute{\rho}[\![\mathsf{rec}]\!]\downarrow\nu,\grave{\rho}[\![\mathsf{rec}]\!]\downarrow\nu\rangle,false)$

$|\,\nu:\mathrm{N}\};$

$gyven=\lambda\hat{\omega}\hat{\upsilon}.\bigvee\{1\leq\nu\leq\#\acute{\upsilon}\wedge 1\leq\nu\leq\#\grave{\upsilon}\rightarrow\hat{\omega}=\langle\acute{\upsilon}\downarrow\nu,\grave{\omega}\downarrow\nu\rangle,false\,|\,\nu:\mathrm{N}\}.$

All the overt pairings between $\acute{\pi}$ and $\grave{\pi}$ can be obtained using $yclept:[\mathrm{W}^{\circ}\times\mathrm{W}^{\circ}]\rightarrow[\mathrm{P}^{\circ}\times\mathrm{P}^{\circ}]\rightarrow\mathrm{T}$, which is defined by

$yclept=\lambda\hat{\omega}\hat{\pi}.hoten\hat{\omega}\hat{\rho}\vee gyven\hat{\omega}\hat{\upsilon}\vee gyven\hat{\omega}\langle\acute{\sigma}\downarrow 2,\grave{\sigma}\downarrow 2\rangle\vee gyven\hat{\omega}\langle\acute{\sigma}\downarrow 3,\grave{\sigma}\downarrow 3\rangle.$

Here we take W, the domain of witnessed values, to be L+B+L*+J+F+G+J+P so that label entry points and return links activated by res are placed in different summands signified by J.

The total tally of pairs is found by proceeding from these values through the states with the aid of an algorithm which is like that used in principle during the marking phase of a garbage collector. Each pair reached gives rise in turn to others; for instance, two locations yield their contents while two function closures provide the members of their free variable lists. To avoid returning the result $\perp$ when, say, $\hat{\alpha}:\mathrm{L}^{\circ}\times\mathrm{L}^{\circ}$ and $\hat{\sigma}:\mathrm{S}^{\circ}\times\mathrm{S}^{\circ}$ are such that $\langle hold\acute{\alpha}\acute{\sigma},hold\grave{\alpha}\grave{\sigma}\rangle=\langle\langle\acute{\alpha}\rangle,\langle\grave{\alpha}\rangle\rangle$ we effectively 'mark off' witnessed values when they are encountered by introducing an integral parameter ν_1. Furthermore we distinguish between locations accessible without passing through members of P and those requiring a route through a member thereof by means of a further parameter ν_0. Accordingly we utilize the algorithm provided for *seen*, which is intended to give all the pairs witnessed from any particular pair by searching to depth ν_1; this it achieves through

$seen=\lambda\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}.\nu_1<1\rightarrow\hat{\omega}_0=\hat{\omega}_1,$

$\acute{\omega}_1:L\vee\grave{\omega}_1:L\rightarrow seen\nu_0(\nu_1-1)\hat{\omega}_0(access\hat{\omega}_1\hat{\pi})\hat{\pi},$

$\hat{\omega}_1:B\times B\rightarrow false,$

$\hat{\omega}_1:L^*\times L^*\rightarrow\bigvee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}\wedge gyven\hat{\omega}_2\hat{\omega}_1\,|\,\hat{\omega}_2:W\times W\},$

$\hat{\omega}_1:J\times J\rightarrow\bigvee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$

$\wedge(hoten\omega_2\langle\acute{\omega}_1\downarrow 2,\grave{\omega}_1\downarrow 2\rangle$

$\vee gyven\hat{\omega}_2\langle\acute{\omega}_1\downarrow 3,\grave{\omega}_1\downarrow 3\rangle\wedge\nu_0<2)\,|\,\hat{\omega}_2:W\times W\},$

$\hat{\omega}_1:F\times F\rightarrow\bigvee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}\wedge hoten\hat{\omega}_2\langle\acute{\omega}_1\downarrow 2,\grave{\omega}_1\downarrow 2\rangle$

$|\,\hat{\omega}_2:W\times W\},$

$\hat{\omega}_1:G\times G\rightarrow\bigvee\{(seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}\wedge hoten\hat{\omega}_2\hat{\rho}_0$

$\vee seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}_0\wedge yclept\hat{\omega}_2\hat{\pi}_0\wedge\nu_0<1)$

$\wedge\hat{\pi}_0=\langle\langle\acute{\omega}_1\downarrow 2,\langle\rangle,\acute{\omega}_1\downarrow 3\rangle,\langle\grave{\omega}_1\downarrow 2,\langle\rangle,\grave{\omega}_1\downarrow 3\rangle\rangle\,|\,\hat{\omega}_2:W\times W\},$

$\hat{\omega}_1:J\times J\rightarrow\bigvee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$

$\wedge(hoten\hat{\omega}_2\langle\acute{\omega}_1\downarrow 2,\grave{\omega}_1\downarrow 2\rangle$

$\vee gyven\hat{\omega}_2\langle\acute{\omega}_1\downarrow 3,\grave{\omega}_1\downarrow 3\rangle\wedge\nu_0<2)\,|\,\hat{\omega}_2:W\times W\},$

$\hat{\omega}_1:P\times P\rightarrow\bigvee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\omega}_1\wedge yclept\hat{\omega}_2\hat{\omega}_1\wedge\nu_0<1\,|\,\hat{\omega}_2:W\times W\},$

$false.$

In fact as we extract the contents of locations by means of $access=\lambda\hat{\omega}\hat{\pi}.\langle(\acute{\omega}:L\rightarrow(area\acute{\omega}\acute{\sigma}\rightarrow hold\acute{\omega}\acute{\sigma},\acute{\omega}),\acute{\omega}),(\grave{\omega}:L\rightarrow(area\grave{\omega}\grave{\sigma}\rightarrow hold\grave{\omega}\grave{\sigma},\grave{\omega}),\grave{\omega})\rangle$, $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}$ is actually independent of $\hat{\rho}$ and $\hat{0}$ although the predicate of 3.2.1 is not. The nature of $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}$ when $\hat{\omega}_1:G\times G$ must remain inscrutable until 2.6.8.

We gather up all the values which will be witnessed when this procedure is applied indefinitely using $kent=\lambda\nu\hat{\omega}\hat{\pi}.\bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}\wedge yclept\hat{\omega}_1\hat{\pi}\,|\,\nu_1:N\}\,|\,\hat{\omega}_1:W\times W\}.$
This in turn enables us to define
$site=\lambda\alpha\rho\upsilon\sigma.kent0\langle\alpha,\alpha\rangle\langle\langle\rho,\upsilon,\sigma\rangle,\langle\rho,\upsilon,\sigma\rangle\rangle\wedge\alpha:L,$
$plot=\lambda\alpha\rho\upsilon\sigma.kent1\langle\alpha,\alpha\rangle\langle\langle\rho,\upsilon,\sigma\rangle,\langle\rho,\upsilon,\sigma\rangle\rangle\wedge\alpha:L$ and
$spot=\lambda\alpha\rho\upsilon\sigma.kent2\langle\alpha,\alpha\rangle\langle\langle\rho,\langle\rangle,\sigma\rangle,\langle\rho,\langle\rangle,\sigma\rangle\rangle\wedge\alpha:L$, in which the predicate $\alpha:L$ ensures that only proper locations yield proper results.

Although *kent* and its derivatives are monotonic there is no reason to expect them to be continuous, as they are constructed from disjunctions of infinite numbers of predicates. That they do indeed provide discontinuities is demonstrated by the sequence $\{\phi_n \mid n\geq 0\}$ such that $\phi_0=\lambda\delta.\,arid[\delta/I]$ and $\phi_{n+1}=\lambda\delta.\,arid[\langle\mathscr{I}[\![I]\!](\zeta\circ revert(arid)),\phi_n\delta\,\langle\rangle\rangle/I]$ for all $n\geq 0$. When the program I::I has ζ as its continuation, $\bigsqcup\phi_n\perp$ is an appropriate environment for it; moreover, as might be imagined, for every α:L *kent* is such that

$kent0\langle\alpha,\alpha\rangle\langle\langle\bigsqcup\phi_n\perp,\langle\rangle,empty\rangle,\langle\bigsqcup\phi_n\perp,\langle\rangle,empty\rangle\rangle=false.$

Notwithstanding this, induction on n shows that whenever $n\geq 0$ $seen0(n+1)\langle\alpha,\alpha\rangle\langle\omega,\omega\rangle\langle\langle\phi_n\perp,\langle\rangle,empty\rangle,\langle\phi_n\perp,\langle\rangle,empty\rangle\rangle=\perp$ for every ω:W such that $yclept\langle\omega,\omega\rangle\langle\langle\phi_n\perp,\langle\rangle,empty\rangle,\langle\phi_n\perp,\langle\rangle,empty\rangle\rangle=true$, so that $\bigsqcup kent0\langle\alpha,\alpha\rangle\langle\langle\phi_n\perp,\langle\rangle,empty\rangle,\langle\phi_n\perp,\langle\rangle,empty\rangle\rangle=\perp$ for all α:L.

It might be hoped that this discontinuity could be eliminated by the use of different methods of enforcing termination when tracing the locations $\langle\grave{\alpha},\grave{\alpha}\rangle$ in a pair $\langle\acute{\sigma},\grave{\sigma}\rangle$ having $\langle hold\acute{\alpha}\acute{\sigma},hold\grave{\alpha}\grave{\sigma}\rangle=\langle\langle\acute{\alpha}\rangle,\langle\grave{\alpha}\rangle\rangle$; among the more obvious of these methods would be that discussed in 3.6.3. Unfortunately all such expedients are doomed to failure in the present case. To see that this is so let *site* momentarily be a continuous function formed in such a way that for every α:L and ρ:U

$site\alpha(arid[\langle\mathscr{I}[\![I]\!](\zeta\circ revert(arid)),\rho,\langle\rangle\rangle/I])\langle\rangle(empty)=site\alpha\rho\langle\rangle(empty).$

By induction on n $site\alpha(\phi_n\perp)\langle\rangle(empty)=site\alpha(\phi_0\perp)\langle\rangle(empty)$, so considerations of continuity dictate that $site\alpha(\bigsqcup\phi_n\perp)\langle\rangle(empty)=site\alpha(\phi_0\perp)\langle\rangle(empty)$. Yet if *site* is to trace precisely the locations which can be accessed $site\alpha(\phi_0\alpha)\langle\rangle(empty)$ must be *true* while $site\alpha(\phi_0(dummy))\langle\rangle(empty)$ must be *false*; *site* being monotonic, $site\alpha(\phi_0\perp)\langle\rangle(empty)$ has therefore to take the value $\perp$. In consequence $site\alpha(\bigsqcup\phi_n\perp)\langle\rangle(empty)$ is $\perp$ although it

should really be *false*.

A similar fate befalls functions obeying the constraint imposed on *novel* in 2.1.1, for they may be monotonic but they cannot be continuous. This can be established by noting that any monotonic function $novel: U \to Y \to S \to L$ satisfies $novel(\phi_n \alpha)()(empty) \sqsupseteq novel(\phi_n \bot)()(empty)$ for every $\alpha: L$ and for all $n \geq 0$, so that if no location α satisfies $\alpha = novel(\phi_n \alpha)()(empty)$ then $novel(\phi_n \bot)()(empty)$ must be $\bot$. Hence when *novel* is constrained as in 2.1.1 $\bigsqcup novel(\phi_n \bot)()(empty)$ takes the value $\bot$ but $novel(\bigsqcup \phi_n \bot)()(empty)$ is a proper location provided that L contains three or more elements.

The root of the discontinuities in *kent* lies in its reliance on a domain U which distinguishes between the environments $\bot$ and $arid[\bot/I]$. The first component of this domain, $Ide \to D^{\circ *}$, could be supplanted by $Ide \to D^*$ only by building a different value $\bot$ into J and G, the members of which can be affected by applying *fix* to environments as in appendix 2. Thus were the first component of U to be $Ide \to D^*$ taking J to be $Z^\circ \times U \times Y$ would give $(\lambda\zeta'.\zeta'(fix(\lambda\rho'.arid[\langle \zeta', \rho', \langle\rangle \rangle / I]))()(empty))(\mathscr{E}[\![I]\!](\zeta \circ revert(arid)))$ the value $\bot$, whereas taking J to be $Z^\circ \times U^\circ \times Y$ would yield a reasonable rendering of $\mathscr{E}[\![I::I]\!]\zeta(arid)()(empty)$. Structuring J as $Z^\circ \times U^\circ \times Y$, however, would again give rise to a discontinuous version of *kent* even if $Ide \to D^*$ were to take the place of $Ide \to D^{\circ *}$. In short it is necessary either to form environments in a finitary manner which avoids using *fix* (as will be done in 3.1.1) or to abandon any prospect of making *kent* continuous. The latter course of action will be adopted here despite the fact that it entails shoring up the theoretical foundations of *novel* store semantics for the reasons discussed below.

Because appendix 2 makes explicit mention of *novel* and *recur*, neither of which is a continuous function of the parameters

drawn from U, Y and S, the existence of the entities required therein cannot be taken for granted. On inspecting the semantic equations it becomes evident that the only demand made by *novel* and *recur* is that the domain Z contain continuations which belie their title by failing to be continuous, whereas the equations appear to presume the continuity of the functions only when *fix* is applied. Since *fix* is required to account for recursion and labelled blocks, to describe while loops and to set up the valuations, the spaces which must comprise solely continuous functions are Z→Z, U→U, [Exp→O]→[Exp→O] and [Dec→O]→[Dec→O]. Thus no difficulties are encountered in the equations if Z is allowed to contain discontinuous functions provided that O remains equal to Z→Z, the space of continuous mappings from Z into itself. Although Z will still be written as U→Y→S→A, in the context of *novel* store semantics it will be understood to have some members which are not continuous. Considerations of cardinal arithmetic preclude identifying Z with the set of all mappings from U×Y×S into A, but in 2.4.4 Z will be given a form which is sufficiently large to encompass all the continuations necessary and is also small enough to permit the construction of the reflexive domain W.

Thus the introduction of *novel* (which is monotonic but not continuous) entails making a minor modification to the lattices which provide the interpretations of programs. This modification is sufficiently striking to make it desirable to distinguish between *novel* store semantics, which is illustrated by appendix 2, and *new* store semantics. The latter invokes *new* where the former invokes *novel* and makes use of a direct translation of the standard valuation $\mathscr{S}$ into store semantics instead of the more complex treatment of 2.1.4; further details of the difference between the two kinds of equation will be given in 2.2.7.

The intention of *spot* is to isolate precisely those locations which can be accessed without passing through an intervening stack, as they alone among the locations revealed by *plot* can be assigned to by a program.

2.1.7. Proposition.

Let $\nu_0<2,\hat{\omega}_2$ and $\hat{\pi}$ be such that $seen\nu_0\nu_3\hat{\omega}_3\hat{\omega}_2\hat{\pi}$ is proper for every ν_3 and $\hat{\omega}_3$. If ν_1, ν_2, $\hat{\omega}_0$ and $\hat{\omega}_1$ satisfy $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}\wedge seen1\nu_2\hat{\omega}_1\hat{\omega}_2\hat{\pi}=true$ then $seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0\hat{\omega}_2\hat{\pi}=true$.

⊲The proof proceeds by an induction on ν_2 which we shall only outline. When $\nu_2<1$, for all $\nu_0<2$, ν_1, $\hat{\omega}_0$, $\hat{\omega}_1$, $\hat{\omega}_2$ and $\hat{\pi}$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}\wedge seen1\nu_2\hat{\omega}_1\hat{\omega}_2\hat{\pi}\supset seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}\wedge\hat{\omega}_1=\hat{\omega}_2\supset seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$. Assume, therefore, that for some ν_2 and for all $\nu_0<2$, $\hat{\omega}_2$ and $\hat{\pi}$ such that $seen\nu_0\nu_3\hat{\omega}_3\hat{\omega}_2\hat{\pi}$ is proper (for every ν_3 and $\hat{\omega}_3$) we have $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}\wedge seen1\nu_2\hat{\omega}_1\hat{\omega}_2\hat{\pi}\supset seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$ for all ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$. Take any suitable $\nu_0,\hat{\omega}_2$ and $\hat{\pi}$ together with some ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ which are subject to the constraint $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}\wedge seen1(\nu_2+1)\hat{\omega}_1\hat{\omega}_2\hat{\pi}=true$.

If $\hat{\omega}_2$:L or $\hat{\omega}_2$:L, $seen1\nu_2\hat{\omega}_1(access\hat{\omega}_2\hat{\pi})\hat{\pi}=true$, so by the induction hypothesis $seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0(access\hat{\omega}_2\hat{\pi})\hat{\pi}=true$ and by the definition in 2.1.6 $seen\nu_0(\nu_1+\nu_2+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}=true$.

If $\hat{\omega}_2$:J×J, $seen1\nu_2\hat{\omega}_1\hat{\omega}_4\hat{\pi}=true$ for some $\hat{\omega}_4$ satisfying $hoten\hat{\omega}_4\langle\hat{\omega}_1+2,\hat{\omega}_1+2\rangle=true$ or $gyven\hat{\omega}_4\langle\hat{\omega}_1+3,\hat{\omega}_1+3\rangle=true$. It is only possible that $seen\nu_0\nu_3\hat{\omega}_3\hat{\omega}_2\hat{\pi}$ be proper for all ν_3 and $\hat{\omega}_3$ if $seen\nu_0\nu_3\hat{\omega}_3\hat{\omega}_4\hat{\pi}$ is proper for all ν_3 and $\hat{\omega}_3$, as the definition of *seen* reveals. Accordingly we may apply the induction hypothesis to $\hat{\omega}_4$, obtaining $seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0\hat{\omega}_4\hat{\pi}=true$ and, since $seen\nu_0(\nu_1+\nu_2+1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$ is proper, $seen\nu_0(\nu_1+\nu_2+1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}=true$.

We can also have $\hat{\omega}_2$:L*×L*, $\hat{\omega}_2$:F×F, $\hat{\omega}_2$:G×G or $\hat{\omega}_2$:J×J, for all of which validating the step in the induction resembles closely

the paragraph above. Because $seen1(\nu_1+1)\hat{\omega}_1\hat{\omega}_2\hat{\pi}=true$, however, we cannot have $\hat{\omega}_2:P\times P$. In consequence we may conclude that, when ν_0, $\hat{\omega}_2$ and $\hat{\pi}$ agree with the premises of the proposition, $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}\wedge seen1\nu_2\hat{\omega}_1\hat{\omega}_2\hat{\pi}\supset seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$ for all ν_1, ν_2, $\hat{\omega}_0$ and $\hat{\omega}_1$.⊁

A similar result can of course be established if $seen2\nu_2\hat{\omega}_1\hat{\omega}_2\hat{\pi}$ is *true*: should $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}$ be *true* in this case $seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$ will be *true* whatever the value of ν_0.

2.1.8. Corollary.

Let $\hat{\pi}$ be such that $seen0\nu_3\hat{\omega}_3\hat{\omega}_2\hat{\pi}$ is proper for all ν_3 and $\hat{\omega}_3$ whenever $yclept\hat{\omega}_2\hat{\pi}=true$. Then for all $\nu_0<2$ and $\hat{\omega}_0$ if there are ν_1 and $\hat{\omega}_1$ having $kent1\hat{\omega}_1\hat{\pi}=true$ and $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}=true$ then $kent\nu_0\hat{\omega}_0\hat{\pi}=true$.

⊀For any ν_0, ν_3 and $\hat{\omega}_3$ $seen\nu_0\nu_3\hat{\omega}_3\hat{\omega}_2\hat{\pi}$ is proper whenever $yclept\hat{\omega}_2\hat{\pi}=true$ by a trivial induction using the structure of *seen*. Suppose that $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}=true$ for some $\hat{\omega}_1$ having $kent1\hat{\omega}_1\hat{\pi}=true$ and for some ν_1 and $\hat{\omega}_0$. Then $seen1\nu_1\hat{\omega}_1\hat{\omega}_2\hat{\pi}=true$ for some $\hat{\omega}_2$ having $yclept\hat{\omega}_2\hat{\pi}=true$, and by 2.1.7 $seen\nu_0(\nu_1+\nu_2)\hat{\omega}_0\hat{\omega}_2\hat{\pi}=true$ so that $kent\nu_0\hat{\omega}_0\hat{\pi}=true$.⊁

2.2. Inclusive predicates.

2.2.1. Cyclic relations between values.

As a prelude to proving that the standard equations for rec I==E and rec I=E are equivalent, we now start to verify that the description of Mal suggested by 1.4.5 is similar to one yielded by a form of store semantics which uses *new*. A further result will relate this to the form using *novel*, which gains its significance from the discussion of 2.1.1. Because we prefer not to specify the ultimate domains to which continuations map their arguments the exposition will be couched in terms of a predicate α on pairs, the first and second elements of which belong to the final domains for standard semantics and for store semantics respectively; thus were these domains to coincide one possibility for α would be the test for equality. Though we need not assume that these domains are the same we can identify them both by A without fear of confusion; similar remarks apply to all the other lattices, such as U, Y and S. Economizing on names in this way allows us to use the pairing conventions of 2.1.6.

Two programs, one evaluated under standard semantics and the other evaluated under store semantics, might be deemed to be equivalent if for every suitable pair of inputs the resulting pair of answers in $A^\circ \times A^\circ$ satisfied α. As we are dealing not with computing mechanisms but with functions, non-terminating programs can be subsumed under this provided that $\alpha\langle \perp,\perp\rangle = true$. The standard entity set up by a compiled program which is ready to be executed is a store transformation $\theta : C^\circ$; the comparable notion in store semantics is the entry point $\langle \zeta,\rho,\upsilon\rangle : Z^\circ \times U \times Y$. Accordingly we seek a predicate σ on $C^\circ \times [Z^\circ \times U \times Y]$ such that $\sigma\langle \theta,\langle \zeta,\rho,\upsilon\rangle\rangle$ is *true* only if $\alpha\langle \theta\sigma,\zeta\rho\upsilon\delta\rangle$ is *true* for every appropriate pair $\delta : S^\circ \times S^\circ$. It is plain that not all pairs can be appropriate, for when m yields a stored label the equivalence of the two interpretations of goto m

depends on the contents of the relevant locations. We must therefore also find a predicate s on $S^\circ\times S^\circ$ to relate the contents of locations which tally; then $c(\theta,(\zeta,\rho,\upsilon))\supset\bigwedge\{a(\theta\sigma,\zeta\rho\upsilon\delta)\mid s\hat{\sigma}\}$.

Suppose that one lattice L is adopted by both the models we are currently considering. We can define a projection r_0 of the domain V for store semantics into that for standard semantics by $r_0=\lambda\beta.\beta\mid B\sqcup\beta\mid L^*\sqcup(\beta:Z^\circ\times U\times Y\to\perp,\beta:O^\circ\times U\to\perp,\perp)$, where $\perp$ in C and $\perp$ in $E\to K\to C$ are used in the conditional expression; we can also define a projection q_0 of the standard V into itself by $q_0=\lambda\beta.\beta\mid B\sqcup\beta\mid L^*\sqcup(\beta:C^\circ\to\perp,\beta:[E\to K\to C]^\circ\to\perp,\perp)$. Let $\clubsuit$ be the functor on stores introduced in 1.3.1. If $\hat{\sigma}:S^\circ\times S^\circ$ is a pair such that $\lambda\alpha.area\alpha\sigma=\lambda\alpha.area\alpha\delta$, $\#\sigma\downarrow 2=\#\delta\downarrow 2$ and $\#\sigma\downarrow 3=\#\delta\downarrow 3$ and if

$$\bigwedge\{\bigvee\{\hat{\beta}=(hold\alpha\sigma,hold\alpha\delta)\mid\alpha\}\vee given\hat{\beta}(\sigma\downarrow 2\S\sigma\downarrow 3,\delta\downarrow 2\S\delta\downarrow 3)\to\vec{\beta}=\tilde{\beta}\vee\hat{\beta}:C^\circ\times[Z^\circ\times U\times Y]\vee\hat{\beta}:[E\to K\to C]^\circ\times[O^\circ\times U],true\mid\hat{\beta}\}$$

is *true*, then $\clubsuit q_0\sigma=\clubsuit r_0\delta$ and for any *new* function

$$\begin{aligned}area(new(\clubsuit r_0\delta))\delta&=area(new(\clubsuit r_0\delta))\sigma\\&=area(new(\clubsuit q_0\sigma))\sigma\\&=area(new\sigma)\sigma\\&=\bigwedge\{area\alpha\sigma\mid\alpha:L\}\to\perp,false\\&=\bigwedge\{area\alpha\delta\mid\alpha:L\}\to\perp,false.\end{aligned}$$

Hence $new\circ\clubsuit r_0$ satisfies the postulate on *new* functions of 1.2.1 and $new(\clubsuit r_0\delta)=new\sigma$ when σ and δ are related as above. We can therefore take $new\circ\clubsuit r_0$ to be the function referred to in *new* store semantics and demand that the equivalence between locations, $l\hat{\alpha}$, be $\hat{\alpha}=\grave{\alpha}\wedge\acute{\alpha}:L\wedge\grave{\alpha}:L$; from now on stores σ and δ corresponding in the manner above will have $l(new\sigma,new\delta)=true$ unless $new\sigma$ and $new\delta$ are both $\perp$ or both τ.

Although the course of a computation depends only on the contents of locations within the area of store we demand that in equivalent stores all the locations hold equivalent values, as the semantic equations do not proscribe access to locations outside

this area. Were we to amend the equations or to introduce more sophisticated predicates akin to those of 2.4.1 we could weaken this requirement so that only locations in the store area were compared, but under our present circumstances we insist that

$s\theta \supset \bigwedge\{(area\acute{\alpha}\acute{\sigma} \wedge area\grave{\alpha}\grave{\sigma} \vee \sim area\acute{\alpha}\acute{\sigma} \wedge \sim area\grave{\alpha}\grave{\sigma}) \wedge v\langle hold\acute{\alpha}\acute{\sigma}, hold\grave{\alpha}\grave{\sigma}\rangle \mid l\alpha\}$

for all $\theta: S^\circ \times S^\circ$.

Here v is intended to relate pairs $\hat{\beta}: V^\circ \times V^\circ$ in such a way that they produce matching effects when used in their respective computations, so that pairs of values which are read into (or written from) equivalent stores will also be subject to it. Accordingly we must also have

$s\theta \supset \#\acute{\sigma}\downarrow 2 = \#\grave{\sigma}\downarrow 2 \wedge \#\acute{\sigma}\downarrow 3 = \#\grave{\sigma}\downarrow 3 \wedge \bigwedge\{given\hat{\beta}\langle \acute{\sigma}\downarrow 2 \S \acute{\sigma}\downarrow 3, \grave{\sigma}\downarrow 2 \S \grave{\sigma}\downarrow 3\rangle \rightarrow v\hat{\beta}, true \mid \hat{\beta}\}$.

Since the effect of a stored label entry point may be to change the flow of control we expect that if $\acute{\beta}: C^\circ$ and $\grave{\beta}: Z^\circ \times U \times Y$ then $\hat{\beta}$ will be such that $v\hat{\beta} \supset c\hat{\beta}$.

The recursion operator for *novel* store semantics provides a valuation $\mathscr{S}$ which may alter the area of the store whereas standard semantics does not. To ensure that the areas we compare remain identical, in *new* store semantics we eliminate *replace* and take $\mathscr{S}[\![\Delta]\!]\rho'\sigma'\downarrow\upsilon$ to be

$\langle \mathscr{T}[\![\Delta]\!] \circ (\lambda\zeta\rho''\upsilon''\sigma''.(\lambda\pi.\zeta(\pi\downarrow 1)((\rho''[\![\mathscr{H}[\![\Delta]\!]\downarrow\upsilon]\!]\downarrow 1\rangle \S \pi\downarrow 2)(\pi\downarrow 3))(\rho''[\![rec]\!]\downarrow 1)),$

$tear[\![\Delta]\!]\rho', \sigma'\rangle$

when $1 \leq \upsilon \leq \#\mathscr{H}[\![\Delta]\!]$.

2.2.2. The information yielded by projections.

The connection between c, s and v proposed above entails a circularity in their definition which cannot be eluded by an appeal to the Tarski fixed point theorem [24] since no function havin them as a fixed point can be monotonic. We use an induction rule like that of 1.3.1 to build them up alongside the reflexive domains. In passing from Pal to Mal we have altered D but not

V, so the form of this rule remains the same even if its content is fresh. It is also valid for store semantics, since we extend to functors and projections the convention that the domains used therein have the names of their standard counterparts.

Our intention is to get predicates which at each stage in the iterative construction of the two lattices labelled V will relate values in one to values in the other. Each predicate v_{n+1} will add to the information yielded by the one at the stage before, v_n, in such a way that the entire sequence will reveal all that we want to know about the values. Owing to the presence of perfect information at every stage if $\hat{\beta}$:B×B, we can assert that in this case $v_n\hat{\beta}$ is $b\hat{\beta}$ where b is a given predicate on B×B. We shall tacitly assume that b is the identity but there is no reason to do so: we might, for example, wish to prove that one program acted similarly to another which interchanged true with false and ∧ with ∨ and which set ~ before the premise of every loop or conditional clause, when $b\langle true,false\rangle$ would need to be *true* and $b\langle true,true\rangle$ would be *false*. Should $\hat{\beta}$ be in L*×L* we know likewise that $v_n\hat{\beta}$ is $\hat{\beta}=\grave{\beta}$ since we have taken $l\hat{\alpha}$ to be $\acute{\alpha}=\grave{\alpha}$ whenever $\hat{\alpha}$:L×L.

It is less easy to decide on the nature of $v_n\langle\perp,\perp\rangle$, although as $q\perp=\perp$ for any injection q our discussion of information suggests that $v_{n+1}\langle\perp,\perp\rangle=v_n\langle\perp,\perp\rangle$ for all n. If we take our initial projection q_0 to be $\lambda\beta.\beta|\,B\sqcup\beta\,|L^*$ and stipulate that $v\hat{\beta}\supset v_0\langle q_0\acute{\beta},q_0\grave{\beta}\rangle$ then $v_0\langle\perp,\perp\rangle$ must be *true* as there is presumably some $\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle$ with $v\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle=true$. There is, however, no need to choose this particular initial projection as other possibilities give rise to the same domain V; in particular this problem might be circumvented were the standard q_0 to be $\lambda\beta.\beta|\,B\sqcup\beta\,|L^*\sqcup(\beta:J\to\perp,\beta:F\to\perp,\perp)$ with $\perp$ belonging to C and to E→K→C in the first two consequents of the conditional expression. Furthermore though we are prepared

to countenance the possibility that the final answer in A given by a computation might be $\perp$ we permit improper intermediate results only if an error has occurred. Accordingly we take $v_n\langle\perp,\perp\rangle$ to be $\perp$ and $v_n\langle\top,\top\rangle$ to be $\top$ in the domain T, and we use a set of projections which is not the most obvious.

There is a natural candidate for that part of this set which is derived from the standard domain V, since we can simply take the q_0 of 2.2.1 and define q_{n+1} to be $\mathfrak{V}q_n$ when $n\geq 0$. Loosely $\mathfrak{V}q_n = q_0 \sqcup \mathfrak{J}q_n \sqcup \mathfrak{f}q_n$ for the functors $\mathfrak{J}$ and $\mathfrak{f}$ suggested in 1.3.3, but in fact $\mathfrak{V}q_n = \lambda\beta.\beta{:}B\to\beta, \beta{:}L^*\to\beta, \beta{:}J\to(\mathfrak{C}q_n)\circ\beta, \beta{:}F\to(\mathfrak{E}q_n\to\mathfrak{K}q_n\to\mathfrak{C}q_n)\circ\beta, \perp$; as $\mathfrak{V}\perp = q_0$ the postulate about V in 1.3.1 ensures that $\lambda\beta.\beta = \bigsqcup q_n$.

In store semantics we might expect the initial projection to be $\lambda\beta.\beta\,|\,B \sqcup \beta\,|\,L^* \sqcup (\beta{:}J\to\langle\perp, cut(\beta\downarrow 2), cut(\beta\downarrow 3)\rangle, \beta{:}F\to\langle\perp, cut(\beta\downarrow 2)\rangle, \perp)$, out of which could be built other mappings using $\mathfrak{Z}$, $\mathfrak{U}$ and $\mathfrak{Y}$, functors for Z, U and Y which will be defined in 2.4.1. This would not truncate V correctly, however: $\mathfrak{C}q_n\circ\theta$, say, somehow 'includes' $\langle\mathfrak{Z}q_n\circ\zeta, \mathfrak{U}q_n\rho, \mathfrak{Y}q_n\upsilon\rangle$, because the environment and stack of the latter correspond to ones sealed into θ in such a way that they cannot be affected by applying $\mathfrak{C}q_n$. Taking $q_{n+1}\langle\zeta,\rho,\upsilon\rangle$ to be $\langle\mathfrak{Z}q_n\circ\zeta,\rho,\upsilon\rangle$ would give a more satisfactory likeness, but would nevertheless remain inappropriate, as when the code part $\mathfrak{Z}q_n\circ\zeta$ was supplied with the arguments ρ and υ it would yield $\lambda\sigma.\mathfrak{A}q_n(\zeta(\mathfrak{U}q_n\rho)(\mathfrak{Y}q_n\upsilon)(\mathfrak{S}q_n\sigma))$, or $\mathfrak{Z}q_n\zeta\rho\upsilon$, although only the state transformation $\lambda\sigma.\mathfrak{A}q_n(\zeta\rho\upsilon(\mathfrak{S}q_n\sigma))$ could hope to emulate $\lambda\sigma.\mathfrak{A}q_n(\theta(\mathfrak{S}q_n\sigma))$. Thus for every projection q on V we take $\mathfrak{C}q$ on Z to be $\lambda\zeta\rho\upsilon.\mathfrak{A}q\circ\zeta\rho\upsilon\circ\mathfrak{S}q$ and $\mathfrak{J}q$ on J to be $\lambda\beta.\langle\mathfrak{C}q\circ(\beta\downarrow 1), \beta\downarrow 2, \beta\downarrow 3\rangle$.

The continuation supplied to a compiled expression in store semantics is expected to have the same environment as that provided to the resulting state transformation. Moreover the resemblance to standard semantics suggests that its stack will differ from the one originally given by having an extra element

at the top. This addition alone is compared with the expressed value returned in the standard equations, so the projection needed for Z, $\mathfrak{K}q$, is $\lambda\zeta\rho\upsilon.\mathfrak{A}q\circ\zeta\rho(\upsilon=()\rightarrow\upsilon,\langle\mathfrak{C}q(\upsilon\downarrow1)\rangle\,\S\upsilon\dagger1)\circ\mathfrak{H}q$, whilst that induced on $0^\circ\times U$, $\mathfrak{F}q$, is $\lambda\beta.\langle(\mathfrak{K}q\rightarrow\mathfrak{K}q)^\circ(\beta\downarrow1),\beta\downarrow2\rangle$. Here we have $\mathfrak{C}q=\lambda\varepsilon.\varepsilon:L\rightarrow\varepsilon,q\varepsilon$ as $E=L+V$ and $\mathfrak{H}q=\lambda\sigma.\langle\lambda\alpha.\langle(\sigma\downarrow1)\alpha\downarrow1,q((\sigma\downarrow1)\alpha\downarrow2)\rangle\rangle\,\S q^*(\sigma\downarrow2)\,\S q^*(\sigma\downarrow3)$ as $S=[L\rightarrow[T\times V]]\times V^*\times V^*$.

The appropriate truncations of the space of stored values are therefore obtained by taking q_0 to be $\lambda\beta.\beta\,|B\sqcup\beta\,|L^*\sqcup(\beta:J\rightarrow\langle\perp,\beta\downarrow2,\beta\downarrow3\rangle,\beta:F\rightarrow\langle\perp,\beta\downarrow2\rangle,\perp)$. As before $q_{n+1}=\mathfrak{V}q_n$ and $\mathfrak{V}q=q_0\sqcup\mathfrak{J}q\sqcup\mathfrak{f}q$, but now $\mathfrak{J}q$ and $\mathfrak{f}q$ modify only the first components of their arguments. The proof that $\lambda\beta.\beta=\bigsqcup q_n$ requires us to verify that $\bigsqcup q_n=\lambda\beta.fix(\mathfrak{W})(\beta\,|\,W)$ when $\mathfrak{W}$ is the original functor for W, the domain of witnessed values, in store semantics; the details will not be given here since those for a more interesting variant may be found in 2.4.2.

Having decided how to regard $v\langle\perp,\perp\rangle$ and what information is available to the predicates at each stage we can at last provide recurrences generating them. The essential relations between stored values are given by

$v_0=\lambda\hat\beta.\hat\beta:B\times B\rightarrow b\hat\beta,\hat\beta:L^*\times L^*\rightarrow\#\acute\beta=\#\grave\beta\wedge\bigwedge\{l\hat\alpha\vee\sim gyven\hat\alpha\hat\beta\,|\hat\alpha\},\hat\beta:J\times J\vee\hat\beta:F\times F$ and

$v_{n+1}=\lambda\hat\beta.v_0\hat\beta\wedge(\hat\beta:J\times J\rightarrow c_{n+1}\hat\beta,\hat\beta:F\times F\rightarrow f_{n+1}\hat\beta,true)$.

These can be explicated using others, the formation of which is reminiscent of the inverse limit construction for V:

$e_{n+1}=\lambda\hat\varepsilon.\hat\varepsilon:L\times L\rightarrow l\hat\varepsilon,\hat\varepsilon:V\times V\rightarrow v_n\hat\varepsilon,false$;

$s_{n+1}=\lambda\delta.\bigwedge\{(area\acute\alpha\delta\wedge area\grave\alpha\partial\vee\sim area\acute\alpha\delta\wedge\sim area\grave\alpha\partial)\wedge v_n\langle hold\acute\alpha\delta,hold\grave\alpha\partial\rangle\,|\,l\hat\alpha\}$

$\qquad\wedge\#\delta\downarrow2=\#\partial\downarrow2\wedge\#\delta\downarrow3=\#\partial\downarrow3\wedge\bigwedge\{gyven\hat\beta\langle\delta\downarrow2\,\S\delta\downarrow3,\partial\downarrow2\,\S\partial\downarrow3\rangle\rightarrow v_n\hat\beta,true\,|\hat\beta\}$;

$c_{n+1}=\lambda\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle.\bigwedge\{a_{n+1}\langle\mathfrak{C}q_n\theta\delta,\mathfrak{C}q_n\zeta\rho\upsilon\partial\rangle\,|s_{n+1}\hat\delta\}$;

$k_{n+1}=\lambda\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle.\bigwedge\{c_{n+1}\langle\mathfrak{K}q_n\kappa\acute\varepsilon,\langle\mathfrak{K}q_n\zeta,\rho,\langle\grave\varepsilon\rangle\,\S\upsilon\rangle\rangle\,|e_{n+1}\hat\varepsilon\}$;

$f_{n+1}=\lambda\langle\phi,\langle\xi,\rho\rangle\rangle.\bigwedge\{k_{n+1}\langle\lambda\varepsilon.\mathfrak{J}q_n\phi\varepsilon\kappa,\langle\mathfrak{K}q_n(\xi(\mathfrak{K}q_n\zeta\circ revert\rho'\,)),divert\rho'\rho,\upsilon\rangle\rangle$

$\qquad\qquad|k_{n+1}\langle\kappa,\langle\zeta,\rho',\upsilon\rangle\rangle\}$.

Only in those predicates which involve applying the arguments as functions is it necessary to truncate them; indeed when the arguments are in L (as they are for $s\hat{\sigma}$, say) even this use of the projections can be eschewed. To extract all that these relations can tell us we therefore define the countable conjunctions $v=\lambda\hat{\beta}.\bigwedge v_n\hat{\beta}$, $e=\lambda\hat{\varepsilon}.\bigwedge e_{n+1}\hat{\varepsilon}$, $s=\lambda\hat{\sigma}.\bigwedge s_{n+1}\hat{\sigma}$, $c=\lambda\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle.\bigwedge c_{n+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle$, $k=\lambda\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle.\bigwedge k_{n+1}\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle$, and $f=\lambda\langle\phi,\langle\xi,\rho\rangle\rangle.\bigwedge f_{n+1}\langle\phi,\langle\xi,\rho\rangle\rangle$. We allow for the possibility that $\mathsf{A}^\circ\times\mathsf{A}^\circ$ may depend on $\mathsf{V}^\circ\times\mathsf{V}^\circ$ by cutting down its predicate to an appropriate level at each stage; hence we also require $a=\lambda\hat{\sigma}.\bigwedge a_{n+1}\hat{\sigma}$. For instance, were A to be S we might take $\mathfrak{A}q_n$ and a_{n+1} to be $\mathfrak{S}q_n$ and $\lambda\hat{\sigma}.\hat{\sigma}=\langle\bot,\bot\rangle\vee\hat{\sigma}=\langle\top,\top\rangle\rightarrow true, s_{n+1}\hat{\sigma}$ respectively, since these satisfy the prerequisites of our next few results.

2.2.3. Lemma.

Suppose that $a_1\langle\bot,\bot\rangle=true$, $a_1\hat{\sigma}$ is always proper, and if $v_{n+1}\hat{\beta}\supset v_n\hat{\beta}$ and $v_n\hat{\beta}\supset v_{n+1}((q_n\times q_n)\hat{\beta})$ for all $\hat{\beta}:\mathsf{V}^\circ\times\mathsf{V}^\circ$ then $a_{n+2}\hat{\sigma}\supset a_{n+1}\hat{\sigma}$ and $a_{n+1}\hat{\sigma}\supset a_{n+2}((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\sigma})$ for all $\hat{\sigma}:\mathsf{A}^\circ\times\mathsf{A}^\circ$. For every member of the relevant domains and for every $n\geq 0$,

(i) $v_{n+1}\hat{\beta}\supset v_n\hat{\beta}$ and $v_n\hat{\beta}\supset v_{n+1}((q_n\times q_n)\hat{\beta})$;

(ii) $e_{n+2}\hat{\varepsilon}\supset e_{n+1}\hat{\varepsilon}$ and $e_{n+1}\hat{\varepsilon}\supset e_{n+2}((\mathfrak{C}q_n\times\mathfrak{C}q_n)\hat{\varepsilon})$;

(iii) $s_{n+2}\hat{\sigma}\supset s_{n+1}\hat{\sigma}$ and $s_{n+1}\hat{\sigma}\supset s_{n+2}((\mathfrak{S}q_n\times\mathfrak{S}q_n)\hat{\sigma})$;

(iv) $c_{n+2}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle\supset c_{n+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle$
and $c_{n+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle\supset c_{n+2}\langle\mathfrak{C}q_n\theta,\langle\mathfrak{C}q_n\zeta,\rho,\upsilon\rangle\rangle$;

(v) $k_{n+2}\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle\supset k_{n+1}\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle$
and $k_{n+1}\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle\supset k_{n+2}\langle\mathfrak{K}q_n\kappa,\langle\mathfrak{K}q_n\zeta,\rho,\upsilon\rangle\rangle$;

(vi) $f_{n+2}\langle\phi,\langle\xi,\rho\rangle\rangle\supset f_{n+1}\langle\phi,\langle\xi,\rho\rangle\rangle$
and $f_{n+1}\langle\phi,\langle\xi,\rho\rangle\rangle\supset f_{n+2}\langle\mathfrak{F}q_n\phi,\mathfrak{F}q_n\langle\xi,\rho\rangle\rangle$.

◁Manifestly if $v_1\hat{\beta}=true$ $v_0\hat{\beta}=true$. On the other hand if $v_0\hat{\beta}=true$ then $v_1((q_0\times q_0)\hat{\beta})=true$ unless, perhaps, $\hat{\beta}:\mathsf{J}\times\mathsf{J}$ or $\hat{\beta}:\mathsf{F}\times\mathsf{F}$.

Should $\hat{\beta}:J\times J$, $v_1((q_0\times q_0)\hat{\beta})=c_1\langle\perp,\langle\perp,\grave{\beta}\downarrow 2,\grave{\beta}\downarrow 3\rangle\rangle=true$ as $\mathfrak{C}q_0\perp=\perp$ and $a_1\langle\perp,\perp\rangle=true$; should $\hat{\beta}:F\times F$, $v_1((q_0\times q_0)\hat{\beta})=f_1\langle\perp,\langle\perp,\grave{\beta}\downarrow 2\rangle\rangle=true$ as $\mathfrak{F}q_0\perp=\perp$ and $a_1\langle\perp,\perp\rangle=true$. Hence (i) holds for $n=0$.

Now assume that (i) is valid when $n=m$ for some $m\geq 0$ and that (ii) to (vi) are valid whenever $m-1\geq n\geq 0$. We shall show that (i) holds when $n=m+1$ and that (ii) to (vi) hold whenever $m\geq n\geq 0$.

For any $\hat{\varepsilon}:E\times E$, by (i) and the definition of q_m, $e_{m+2}\hat{\varepsilon}\wedge\sim\hat{\varepsilon}:L\times L\supset v_{m+1}\hat{\varepsilon}\supset v_m\hat{\varepsilon}\supset e_{m+1}\hat{\varepsilon}$ and $e_{m+1}\hat{\varepsilon}\wedge\sim\hat{\varepsilon}:L\times L\supset v_m\hat{\varepsilon}\supset v_{m+1}((q_m\times q_m)\hat{\varepsilon})\supset e_{m+2}((\mathfrak{C}q_m\times\mathfrak{C}q_m)\hat{\varepsilon})$. As $e_{m+1}\hat{\varepsilon}=l\hat{\varepsilon}$ if $\hat{\varepsilon}:L\times L$, (ii) is valid when $n=m$.

The proof that (iii) is valid when $n=m$ is similar to that for (ii) except in that use is made of $\mathfrak{S}q_m$ rather than $\mathfrak{C}q_m$.

Suppose that $c_{m+2}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle=true$ and that $s_{m+1}\vartheta=true$ for some ϑ. The paragraph above shows that $s_{m+2}((\mathfrak{S}q_m\times\mathfrak{S}q_m)\vartheta)=true$, so, writing for convenience $\hat{\sigma}=\langle\mathfrak{C}q_{m+1}\theta(\mathfrak{S}q_m\acute{\sigma}),\mathfrak{C}q_{m+1}\zeta\rho\upsilon(\mathfrak{S}q_m\grave{\sigma})\rangle$, $\hat{\vartheta}=\langle\mathfrak{A}q_{m+1}(\theta(\mathfrak{S}q_m\acute{\sigma})),\mathfrak{A}q_{m+1}\zeta\rho\upsilon(\mathfrak{S}q_m\grave{\vartheta})\rangle$ and $a_{m+2}\hat{\vartheta}=true$ as $q_{m+1}\circ q_m=q_m$. Now $a_{m+2}\hat{\vartheta}\supset a_{m+1}\hat{\vartheta}\supset a_{m+2}((\mathfrak{A}q_m\times\mathfrak{A}q_m)\hat{\vartheta})\supset a_{m+1}((\mathfrak{A}q_m\times\mathfrak{A}q_m)\hat{\vartheta})$; hence $a_{m+1}\langle\mathfrak{C}q_m\theta\acute{\sigma},\mathfrak{C}q_m\zeta\rho\upsilon\grave{\vartheta}\rangle=true$ and $c_{m+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle=true$.

Conversely, if $c_{m+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle=true$ and $s_{m+2}\hat{\vartheta}=true$ then $s_{m+1}\hat{\vartheta}=true$, so $a_{m+1}\langle\mathfrak{C}q_m\theta\acute{\sigma},\mathfrak{C}q_m\zeta\rho\upsilon\grave{\vartheta}\rangle=true$ and $a_{m+2}\langle\mathfrak{C}q_m\theta\acute{\sigma},\mathfrak{C}q_m\zeta\rho\upsilon\grave{\vartheta}\rangle=true$. Since ϑ is any suitable pair of stores $c_{m+2}\langle\mathfrak{C}q_m\theta,\langle\mathfrak{C}q_m\zeta,\rho,\upsilon\rangle\rangle=true$ and (iv) follows for $n=m$.

For any $\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle:K^\circ\times J$,

$$
\begin{aligned}
k_{m+2}\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle\wedge e_{m+1}\hat{\varepsilon}&\supset k_{m+2}\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle\wedge e_{m+2}((\mathfrak{C}q_m\times\mathfrak{C}q_m)\hat{\varepsilon})\\
&\supset c_{m+2}\langle\mathfrak{K}q_{m+1}\kappa(\mathfrak{C}q_m\acute{\varepsilon}),\langle\mathfrak{K}q_{m+1}\zeta,\rho,\langle\mathfrak{C}q_m\grave{\varepsilon}\rangle\,\S\upsilon\rangle\rangle\\
&\supset c_{m+1}\langle\mathfrak{K}q_{m+1}\kappa(\mathfrak{C}q_m\acute{\varepsilon}),\langle\mathfrak{K}q_{m+1}\zeta,\rho,\langle\mathfrak{C}q_m\grave{\varepsilon}\rangle\,\S\upsilon\rangle\rangle\\
&\supset c_{m+2}\langle\mathfrak{K}q_m\kappa(\mathfrak{C}q_m\acute{\varepsilon}),\langle\mathfrak{K}q_m\zeta,\rho,\langle\mathfrak{C}q_m\grave{\varepsilon}\rangle\,\S\upsilon\rangle\rangle\\
&\supset c_{m+1}\langle\mathfrak{K}q_m\kappa(\mathfrak{C}q_m\acute{\varepsilon}),\langle\mathfrak{K}q_m\zeta,\rho,\langle\mathfrak{C}q_m\grave{\varepsilon}\rangle\,\S\upsilon\rangle\rangle\\
&\supset c_{m+1}\langle\mathfrak{K}q_m\kappa\acute{\varepsilon},\langle\mathfrak{K}q_m\zeta,\rho,\langle\grave{\varepsilon}\rangle\,\S\upsilon\rangle\rangle
\end{aligned}
$$

whilst

$$k_{m+1}(\kappa,(\zeta,\rho,\upsilon)) \wedge e_{m+2}\hat{\varepsilon} \supset k_{m+1}(\kappa,(\zeta,\rho,\upsilon)) \wedge e_{m+1}\hat{\varepsilon}$$
$$\supset c_{m+1}(\mathfrak{K}q_m\kappa\hat{\varepsilon},(\mathfrak{K}q_m\zeta,\rho,(\hat{\varepsilon})^{\S}\upsilon))$$
$$\supset c_{m+2}(\mathfrak{K}q_m\kappa\hat{\varepsilon},(\mathfrak{K}q_m\zeta,\rho,(\hat{\varepsilon})^{\S}\upsilon)) .$$

Hence the validity of (v) when $n=m$ follows from that of (ii) and that of (iv). That (vi) holds is similarly a direct consequence of (v).

Finally, for any $\hat{\beta}$ in J×J or F×F

$$v_{m+2}\hat{\beta} \supset (\hat{\beta}:J\times J\to c_{m+2}\hat{\beta}, f_{m+2}\hat{\beta})$$
$$\supset (\hat{\beta}:J\times J\to c_{m+1}\hat{\beta}, f_{m+1}\hat{\beta})$$
$$\supset v_{m+1}\hat{\beta}$$
$$\supset (\hat{\beta}:J\times J\to c_{m+1}\hat{\beta}, f_{m+1}\hat{\beta})$$
$$\supset (\hat{\beta}:J\times J\to c_{m+2}((\mathfrak{J}q_m\times\mathfrak{J}q_m)\hat{\beta}), f_{m+2}((\mathfrak{F}q_m\times\mathfrak{F}q_m)\hat{\beta}))$$
$$\supset v_{m+2}((q_{m+1}\times q_{m+1})\hat{\beta}),$$

and by induction the result is proven for all n.⊁

A predicate a will be said to be 'inclusive' if whenever $\{\hat{\delta}_m \mid m\geq 0\}$ is a sequence with $\hat{\delta}_{m+1}\supseteq\hat{\delta}_m$ for all $m\geq 0$ $\bigwedge a\hat{\delta}_m \supset a(\bigsqcup\hat{\delta}_m)$.

2.2.4. Lemma.

Suppose that for any $n\geq 0$ if v_n is inclusive a_{n+1} is inclusive; then v, e, s, c, k, f and a, defined as above, are inclusive.

⊀We shall show first that v_n is inclusive for every n. Certainly, if $\{\hat{\beta}_m\}$ is an increasing sequence $\bigwedge v_0\hat{\beta}_m \supset v_0(\bigsqcup\hat{\beta}_m)$ and v_0 is inclusive.

Suppose that for some n v_n is inclusive, so that a_{n+1} is inclusive and by trivial calculations e_{n+1} and s_{n+1} are inclusive. Let $\{(\theta_m,(\zeta_m,\rho_m,\upsilon_m))\}$ be an increasing sequence of members of J×J such that $c_{n+1}(\theta_m,(\zeta_m,\rho_m,\upsilon_m))=true$ for all $m\geq 0$; the continuity of $\mathfrak{C}q_n$ ensures that for all δ we have

$\bigwedge a_{n+1}\langle \mathfrak{C} q_n \theta_m \sigma, \mathfrak{C} q_n \zeta_m \rho_m \upsilon_m \delta\rangle \supset a_{n+1}\langle \bigsqcup \mathfrak{C} q_n \theta_m \sigma, \bigsqcup \mathfrak{C} q_n \zeta_m \rho_m \upsilon_m \delta\rangle$

$\supset a_{n+1}\langle \mathfrak{C}\, q_n (\bigsqcup \theta_m)\sigma, \mathfrak{C}\, q_n (\bigsqcup \zeta_m)(\bigsqcup \rho_m)(\bigsqcup \upsilon_m)\delta\rangle$

and thus that $c_{n+1}\langle \bigsqcup \theta_m, \bigsqcup\langle \zeta_m, \rho_m, \upsilon_m\rangle\rangle = true$. We may now show in turn that as c_{n+1} is inclusive so is k_{n+1} and that as k_{n+1} is inclusive so is f_{n+1}. Thus when $\{\hat{\beta}_m\}$ is an increasing sequence of elements of $V^\circ \times V^\circ$ we may presume that $\bigwedge v_{n+1}\hat{\beta}_m \supset v_{n+1}(\bigsqcup \hat{\beta}_m)$.

Consequently we know that v_n is inclusive for all n, and using the hypothesis of the lemma we may conclude that $\bigwedge a_{n+1}\hat{\delta}_m \supset a_{n+1}(\bigsqcup \hat{\delta}_m)$ for all n and for every increasing sequence $\{\hat{\delta}_m\}$. Taking any such sequence,

$\bigwedge a\hat{\delta}_m \neq \bigwedge\bigwedge a_{n+1}\hat{\delta}_m \supset \bigwedge a_{n+1}(\bigsqcup \hat{\delta}_m) \supset a(\bigsqcup \hat{\delta}_m)$ by the definition of a.

Because v_n, e_{n+1}, s_{n+1}, j_{n+1}, k_{n+1} and f_{n+1} are inclusive for all n the commutativity of conjunctions shows also that our v, e, s, c, k and f are inclusive.⊁

The predicates above thus fulfil our hope that in moving from stage q_n to stage q_{n+1} the information available increases. It only remains to be shown that the total arrived at by the end of the process yields the self-referential relations desired.

2.2.5. Proposition.

Suppose that $a_1\langle \bot, \bot\rangle = true$, $a_1\hat{\delta}$ is always proper, and if v_n is inclusive, equals $v \circ (q_n \times q_n)$ and has $v_{n+1}\hat{\beta} \supset v_n\hat{\beta}$ and $v_n\hat{\beta} \supset v_{n+1}((q_n \times q_n)\hat{\beta})$ for all $\hat{\beta}: V^\circ \times V^\circ$ then a_{n+1} is inclusive, equals $a \circ (\mathfrak{A} q_n \times \mathfrak{A} q_n)$ and has $a_{n+2}\hat{\delta} \supset a_{n+1}\hat{\delta}$ and $a_{n+1}\hat{\delta} \supset a_{n+2}((\mathfrak{A} q_n \times \mathfrak{A} q_n)\delta)$ for all $\hat{\delta}: A^\circ \times A^\circ$. The final predicates obey the following conditions:

(i) $v = \lambda\hat{\beta}.\hat{\beta}: B\times B \to b\hat{\beta}, \hat{\beta}: L^* \times L^* \to \#\hat{\beta} = \#\grave{\beta} \wedge \bigwedge\{gyven\grave{\alpha}\hat{\beta} \to l\grave{\alpha}, true \mid \grave{\alpha}\},$

$\hat{\beta}: J\times J \to c\hat{\beta}, \hat{\beta}: F\times F \to f\hat{\beta}, false;$

(ii) $e = \lambda\hat{\varepsilon}.\hat{\varepsilon}: L\times L \to l\hat{\varepsilon}, \hat{\varepsilon}: V\times V \to v\hat{\varepsilon}, false;$

(iii) $s = \lambda\delta.\bigwedge\{(area\acute{\alpha}\delta \wedge area\grave{\alpha}\delta \vee \sim area\acute{\alpha}\delta \wedge \sim area\grave{\alpha}\delta) \wedge v\langle hold\acute{\alpha}\delta, hold\grave{\alpha}\delta\rangle \mid l\grave{\alpha}\}$

$\wedge \#\delta\downarrow 2 = \#\delta\downarrow 2 \wedge \#\delta\downarrow 3 = \#\delta\downarrow 3 \wedge \bigwedge\{gyven\hat{\beta}\langle \delta\downarrow 2 \S \delta\downarrow 3, \delta\downarrow 2 \S \delta\downarrow 3\rangle \to v\hat{\beta}, true \mid \hat{\beta}\};$

(iv) $c=\lambda\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle.\bigwedge\{a\langle\theta\delta,\zeta\rho\upsilon\partial\rangle\,|\,s\delta\}$;

(v) $k=\lambda\langle\kappa,\langle\zeta,\rho,\upsilon\rangle\rangle.\bigwedge\{c\langle\kappa\hat{\varepsilon},\langle\zeta,\rho,\langle\grave{\varepsilon}\rangle\S\upsilon\rangle\rangle\,|\,e\hat{\varepsilon}\}$;

(vi) $f=\lambda\langle\phi,\langle\xi,\rho\rangle\rangle.\bigwedge\{k\langle\lambda\varepsilon.\phi\varepsilon\kappa,\langle\xi(\zeta\circ revert\rho'),divert\rho'\rho,\upsilon\rangle\rangle\,|\,k\langle\kappa,\langle\zeta,\rho',\upsilon\rangle\rangle\}$

That (i), (ii) and (iii) hold for the predicates set up above is an immediate consequence of the way in which conjunctions distribute over conditional clauses. Thus, for example,

$e=\lambda\hat{\varepsilon}.\bigwedge e_{n+1}\hat{\varepsilon}$

$=\lambda\hat{\varepsilon}.\bigwedge(\hat{\varepsilon}:L\times L\to l\hat{\varepsilon},\hat{\varepsilon}:V\times V\to v_n\hat{\varepsilon},false)$

$=\lambda\hat{\varepsilon}.\,\hat{\varepsilon}:L\times L\to l\hat{\varepsilon},\hat{\varepsilon}:V\times V\to\bigwedge v_n\hat{\varepsilon},false$

$=\lambda\hat{\varepsilon}.\,\hat{\varepsilon}:L\times L\to l\hat{\varepsilon},\hat{\varepsilon}:V\times V\to v\hat{\varepsilon},false.$

In conjunction with 2.2.3 this shows that $v\circ(q_n\times q_n)=v_n$, $e\circ(\mathbb{C}q_n\times\mathbb{C}q_n)=e_{n+1}$ and $s\circ(\mathbb{S}q_n\times\mathbb{S}q_n)=s_{n+1}$ for every $n\geq 0$, since when $\hat{\beta}$ is a typical member of $V^\circ\times V^\circ$

$v_n\hat{\beta}\supset\bigwedge\{v_{m+1}((q_n\times q_n)\hat{\beta})\,|\,m\geq n\}$

$\supset\bigwedge\{v_{m+1}((q_n\times q_n)\hat{\beta})\,|\,m\geq n\}\wedge\bigwedge\{v_{m+1}((q_n\times q_n)\hat{\beta})\,|\,m\leq n\}$

$\supset\bigwedge\{v_{m+1}((q_n\times q_n)\hat{\beta})\,|\,m\geq 0\}$

$\supset v((q_n\times q_n)\hat{\beta})$

and $v((q_n\times q_n)\hat{\beta})\supset v_n((q_n\times q_n)\hat{\beta})\supset v_n\hat{\beta}$ from the definitions of the predicates. Hence if, say, A is S and a_{n+1} is $\lambda\theta.\theta=\langle\bot,\bot\rangle\vee\theta=\langle\top,\top\rangle\to true,s_{n+1}\theta$ we are assured that $a\circ(\mathbb{A}q_n\times\mathbb{A}q_n)=a_{n+1}$. By the same token $c\circ(\mathbb{J}q_n\times\mathbb{J}q_n)=c_{n+1}$ (so that for every $\hat{\varepsilon}:E\times E$ $c_{n+1}\langle\mathbb{K}q_n\kappa\varepsilon,\langle\mathbb{K}q_n\zeta,\rho,\langle\grave{\varepsilon}\rangle\S\upsilon\rangle\rangle\supset c\langle\mathbb{K}q_n\kappa\hat{\varepsilon},\langle\mathbb{K}q_n\zeta,\rho,\langle\grave{\varepsilon}\rangle\S\upsilon\rangle\rangle$) and $f\circ(\mathbb{J}q_n\times\mathbb{J}q_n)=f_{n+1}$.

Suppose that $c\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle=true$ and that $s\theta=true$ for some θ. Then for every n $s_{n+1}\theta=true$ so $a\langle\mathbb{C}q_n\theta\delta,\mathbb{C}q_n\zeta\rho\upsilon\partial\rangle=a_{n+1}\langle\mathbb{C}q_n\theta\delta,\mathbb{C}q_n\zeta\rho\upsilon\partial\rangle=true$ and by 2.2.4 $a\langle\theta\delta,\zeta\rho\upsilon\partial\rangle=a\langle\bigsqcup\mathbb{C}q_n\theta\delta,\bigsqcup\mathbb{C}q_n\zeta\rho\upsilon\partial\rangle=\bigwedge a\langle\mathbb{C}q_n\theta\delta,\mathbb{C}q_n\zeta\rho\upsilon\partial\rangle=true$ as $\{\langle\mathbb{C}q_n\theta\delta,\mathbb{C}q_n\zeta\rho\upsilon\partial\rangle\}$ is an increasing sequence.

Conversely, if $a\langle\theta\delta,\zeta\rho\upsilon\partial\rangle=true$ whenever $s\theta=true$ suppose that $s_{n+1}\theta=true$ for some n; $s((\mathbb{S}q_n\times\mathbb{S}q_n)\theta)=true$ and thus

$a_{n+1}\langle \mathbb{C}q_n\theta\sigma', \mathbb{C}q_n\zeta\rho\upsilon\delta\rangle = a\langle \theta(\mathbb{S}q_n\sigma'), \zeta\rho\upsilon(\mathbb{S}q_n\delta)\rangle = true$. Hence (iv) holds when c is defined by the countable conjunctions of 2.2.2.

Suppose that $k\langle \kappa, \langle \zeta, \rho, \upsilon\rangle\rangle = true$ and that $e\hat{\varepsilon} = true$ for some $\langle \kappa, \langle \zeta, \rho, \upsilon\rangle\rangle$ and $\hat{\varepsilon}: E^\circ \times E^\circ$. Then for every $n \geq 0$ in fact $e_{n+1}\hat{\varepsilon} = true$ while 2.2.4 gives

$$\begin{aligned}\bigwedge c_{n+1}\langle \mathbb{A}q_n\kappa\hat{\varepsilon}, \langle \mathbb{A}q_n\zeta, \rho, \langle \hat{\varepsilon}\rangle \S\upsilon\rangle\rangle &\supset \bigwedge c\langle \mathbb{A}q_n\kappa\hat{\varepsilon}, \langle \mathbb{A}q_n\zeta, \rho, \langle \hat{\varepsilon}\rangle \S\upsilon\rangle\rangle \\ &\supset c\langle \bigsqcup \mathbb{A}q_n\kappa\hat{\varepsilon}, \bigsqcup \langle \mathbb{A}q_n\zeta, \rho, \langle \hat{\varepsilon}\rangle \S\upsilon\rangle\rangle \\ &\supset c\langle \kappa\hat{\varepsilon}, \langle \zeta, \rho, \langle \hat{\varepsilon}\rangle \S\upsilon\rangle\rangle ;\end{aligned}$$

consequently $c\langle \kappa\hat{\varepsilon}, \langle \zeta, \rho, \langle \hat{\varepsilon}\rangle \S\upsilon\rangle\rangle = true$.

Contrariwise, should $\langle \kappa, \langle \zeta, \rho, \upsilon\rangle\rangle$ be such that whenever $e\hat{\varepsilon} = true$ we have $c\langle \kappa\hat{\varepsilon}, \langle \zeta, \rho, \langle \hat{\varepsilon}\rangle \S\upsilon\rangle\rangle = true$ then inevitably

$$e_{n+1}\hat{\varepsilon} \supset e((\mathbb{C}q_n \times \mathbb{C}q_n)\hat{\varepsilon}) \supset c\langle \kappa(\mathbb{C}q_n\hat{\varepsilon}), \langle \zeta, \rho, \langle \mathbb{C}q_n\hat{\varepsilon}\rangle \S\upsilon\rangle\rangle \supset c_{n+1}\langle \mathbb{A}q_n\kappa\hat{\varepsilon}, \langle \mathbb{A}q_n\zeta, \rho, \langle \hat{\varepsilon}\rangle \S\upsilon\rangle\rangle$$

so that $k_{n+1}\langle \kappa, \langle \zeta, \rho, \upsilon\rangle\rangle = true$ for every $n \geq 0$.

The proof that (vi) holds stems from the definition of k much as the proof of (v) given in the preceding two paragraphs stems from the definition of e. Its only noteworthy feature is the use of the fact that $\mathbb{A}q_n\zeta \circ revert\rho' = \mathbb{A}q_n(\zeta \circ revert\rho')$ for all ζ, ρ' and n.⊁

We could try to adopt the method of Morris [12] by viewing the link between standard and store semantics as a homomorphism between the source and target meanings of programs for which both the source language and the target language would be Mal. Here, however, this approach does not seem to be very helpful, for we cannot turn the predicate v into a projection. Though there is an initial mapping v'_0 from the domain V for store semantics into the domain V for standard semantics it has no natural inverse v''_0, and in consequence functors like $\mathbb{V}$ cannot provide iterated mappings v'_{n+1} and v''_{n+1} out of which limiting projections could be formed as joins. Furthermore no other techniques will yield projections, since our next result will show that the relations set up above are unique.

2.2.6. Corollary.

Suppose that v, e, s, c, k, f and a are predicates such that $a(\perp,\perp)=true$ and a is inclusive. Suppose further that for every $n\geq 0$ if $v\hat{\beta}\supset v_n\hat{\beta}$ and $v_n\hat{\beta}\supset v((q_n\times q_n)\hat{\beta})$ for all $\hat{\beta}:V^\circ\times V^\circ$ then $a\hat{\delta}\supset a_{n+1}\hat{\delta}$ and $a_{n+1}\delta\supset a((\mathfrak{A}q_n\times\mathfrak{A}q_n)\delta)$ for all $\hat{\delta}:A^\circ\times A^\circ$. Should the conclusion of 2.2.5 hold for these predicates they must coincide with those set up above.

⊲We shall show that for every n and $\hat{\beta}$ these predicates are related to those of 2.2.2 by the implications $v\hat{\beta}\supset v_n\hat{\beta}$ and $v_n\hat{\beta}\supset v((q_n\times q_n)\hat{\beta})$. When $n=0$ any $\hat{\beta}:V^\circ\times V^\circ$ satisfies $v\hat{\beta}\supset v_0\hat{\beta}$; moreover because $a(\perp,\perp)=true$ we can verify in turn that $c(\perp,\perp)=true$, $k(\perp,\perp)=true$ and $f(\perp,\perp)=true$, so that $v((q_0\times q_0)\hat{\beta})=v_0\hat{\beta}$.

Let n be such that $v\hat{\beta}\supset v_n\hat{\beta}$ and $v_n\hat{\beta}\supset v((q_n\times q_n)\hat{\beta})$ for every $\hat{\beta}:V^\circ\times V^\circ$; then as usual corresponding remarks are valid for e_{n+1} and s_{n+1}. For any $\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle$, if $c\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle\wedge s_{n+1}\hat{\theta}=true$ then in fact $s((\clubsuit q_n\times\clubsuit q_n)\hat{\partial})=true$ and

$$a\langle\theta(\clubsuit q_n\delta),\zeta\rho\upsilon(\clubsuit q_n\partial)\rangle\supset a_{n+1}\langle\theta(\clubsuit q_n\delta),\zeta\rho\upsilon(\clubsuit q_n\partial)\rangle$$
$$\supset a\langle\complement q_n\theta\delta,\complement q_n\zeta\rho\upsilon\partial\rangle$$
$$\supset a_{n+1}\langle\complement q_n\theta\delta,\complement q_n\zeta\rho\upsilon\partial\rangle$$

so $c_{n+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle=true$, whereas if $c_{n+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle=true$ then $s\hat{\theta}\supset s_{n+1}\hat{\theta}\supset a_{n+1}\langle\complement q_n\theta\delta,\complement q_n\zeta\rho\upsilon\partial\rangle\supset a\langle\complement q_n\theta\delta,\complement q_n\zeta\rho\upsilon\partial\rangle$. Hence c satisfies $c\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle\supset c_{n+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle$ and $c_{n+1}\langle\theta,\langle\zeta,\rho,\upsilon\rangle\rangle\supset c\langle\mathfrak{I}q_n\theta,\mathfrak{I}q_n\langle\zeta,\rho,\upsilon\rangle\rangle$; from this we may obtain the analogous property of k and thus infer that $f\langle\phi,\langle\xi,\rho\rangle\rangle\supset f_{n+1}\langle\phi,\langle\xi,\rho\rangle\rangle$ and $f_{n+1}\langle\phi,\langle\xi,\rho\rangle\rangle\supset f\langle\mathfrak{f}q_n\phi,\mathfrak{f}q_n\langle\xi,\rho\rangle\rangle$ in all cases. By an argument akin to that in 2.2.3, $v\hat{\beta}\supset v_{n+1}\hat{\beta}$ and $v_{n+1}\hat{\beta}\supset v((q_{n+1}\times q_{n+1})\hat{\beta})$ for every $\hat{\beta}:V^\circ\times V^\circ$.

As a is inclusive the reasoning of 2.2.4 shows that so are all the other predicates. The induction above therefore proves that $v\hat{\beta}\supset\bigwedge v_n\hat{\beta}\supset\bigwedge v((q_n\times q_n)\hat{\beta})\supset v(\bigsqcup(q_n\times q_n)\hat{\beta})\supset v\hat{\beta}$ for all $\hat{\beta}:V^\circ\times V^\circ$. Similarly $a\hat{\delta}\supset\bigwedge a_{n+1}\hat{\delta}\supset a\hat{\delta}$ for every $\hat{\delta}:A^\circ\times A^\circ$ and we may conclude that this set of predicates is indeed that constructed in 2.2.2.⊳

2.2.7. Properties of program texts.

As will be confirmed in 2.2.8, the results of 2.2.5 would not be changed if the undesirable members of the domains were not factored out (in defiance of the policy propounded in 1.2.2); were this done the ultimate inclusive predicates could themselves be used to restrict attention to the proper values in the components of the space S. Any decision about whether or not to make the domains as small as possible by removing redundancy and by invoking propositions about slit lattices can therefore be made without regard to the construction given above. There is, however, one situation in which the improper elements cannot be deleted: the application of the Knaster fixed point theorem [9] to facilitate recursion involves 'seeding' the environment with $\perp$, and if, say, $\rho[\perp/\mathrm{I}]$ and $\langle\zeta,\perp,\upsilon\rangle$ are identified with $\perp$ in U and $\perp$ in D respectively the equation for ϕ provided by appendix 2 will cause the entire state vector to collapse to $\perp$. This cannot happen in standard semantics, where a label entry point having $\perp$ as its environment produces a store transformation belonging to C, not an improper element of D, even when the first component of U is taken to be $\mathrm{Ide}\to\mathrm{D}^*$ instead of the more usual $\mathrm{Ide}\to\mathrm{D}^{\circ *}$.

The relations set up so far can compare stores but not environments; in keeping with the preceding paragraph they will be extended to U in a way which accepts that the denoted value $\perp$ is legitimate (by contrast with the stored value $\perp$ considered in 2.2.2). Other theorems would require the introduction of functors such as Φ, but here it is enough to let

$$g=\lambda\langle\gamma,\langle\xi,\rho,\sigma\rangle\rangle.\bigwedge\{c\langle\gamma\kappa,\langle\lambda\rho''\upsilon''\sigma''.\xi\zeta\rho[\langle\rho'',\upsilon'',\sigma''\rangle/\mathrm{rec}]\langle\rangle\sigma,\rho',\upsilon'\rangle\rangle \mid k\langle\kappa,\langle\zeta,\rho',\upsilon'\rangle\rangle\};$$

$$d=\lambda\hat{\delta}.\hat{\delta}=\langle\perp,\perp\rangle\vee\hat{\delta}=\langle\top,\top\rangle\to true,\hat{\delta}:\mathrm{E}\times\mathrm{E}\to e\hat{\delta},\hat{\delta}:\mathrm{G}\times\mathrm{G}\to g\hat{\delta},false;$$

$$u=\lambda\hat{\rho}.\bigwedge\{\hat{\rho}[\![\mathrm{I}]\!]=\langle\rangle\to true,d\langle\hat{\rho}[\![\mathrm{I}]\!]\downarrow 1,\hat{\rho}[\![\mathrm{I}]\!]\downarrow 1\rangle\wedge(\#\hat{\rho}[\![\mathrm{I}]\!]\leq\#\hat{\rho}[\![\mathrm{I}]\!]) \mid \mathrm{I}:\mathrm{Ide}\}$$

$$\wedge(\hat{\rho}[\![\mathrm{res}]\!]=\langle\rangle\to true,k\langle\hat{\rho}[\![\mathrm{res}]\!]\downarrow 1,\hat{\rho}[\![\mathrm{res}]\!]\downarrow 1\rangle\wedge(\#\hat{\rho}[\![\mathrm{res}]\!]\leq\#\hat{\rho}[\![\mathrm{res}]\!])).$$

It is now clear under which circumstances the standard semantics of an expression should be deemed to be equivalent to its store semantics: when applied to continuations related by k the 'compiled' versions of the expression must form continuations related by c. In principle whether this is so may depend on the valuation enployed, so it is necessary to introduce predicates covering all the possibilities as follows:

$$E=\lambda E.\bigwedge\{c\langle\mathscr{E}[\![E]\!]\hat{\rho}\kappa,\langle\mathscr{E}[\![E]\!]\zeta,\hat{\rho},\upsilon\rangle\rangle \mid rent[\![E]\!]\hat{\rho}\wedge u\hat{\rho}\wedge k\langle\kappa,\langle\zeta,\hat{\rho},\upsilon\rangle\rangle\};$$

$$L=\lambda E.\bigwedge\{c\langle\mathscr{L}[\![E]\!]\hat{\rho}\kappa,\langle\mathscr{L}[\![E]\!]\zeta,\hat{\rho},\upsilon\rangle\rangle \mid rent[\![E]\!]\hat{\rho}\wedge u\hat{\rho}\wedge k\langle\kappa,\langle\zeta,\hat{\rho},\upsilon\rangle\rangle\};$$

$$R=\lambda E.\bigwedge\{c\langle\mathscr{R}[\![E]\!]\hat{\rho}\kappa,\langle\mathscr{R}[\![E]\!]\zeta,\hat{\rho},\upsilon\rangle\rangle \mid rent[\![E]\!]\hat{\rho}\wedge u\hat{\rho}\wedge k\langle\kappa,\langle\zeta,\hat{\rho},\upsilon\rangle\rangle\};$$

$$G=\lambda E.\bigwedge\{c\langle\mathscr{G}[\![E]\!]\hat{\rho}\kappa,\langle\mathscr{G}[\![E]\!]\zeta,\hat{\rho},\upsilon\rangle\rangle \mid torn[\![E]\!]\hat{\rho}\wedge u\hat{\rho}\wedge k\langle\kappa,\langle\zeta,\hat{\rho},\upsilon\rangle\rangle\};$$

$$P=\lambda E.\mathscr{J}[\![E]\!]=\langle\rangle\vee\bigwedge\{\bigwedge\{c\langle\mathscr{P}[\![E]\!]\hat{\rho}\kappa\downarrow\nu,\mathscr{P}[\![E]\!]\zeta\hat{\rho}\upsilon\downarrow\nu\rangle \mid 1\le\nu\le\#\mathscr{J}[\![E]\!]\}$$
$$\mid torn[\![E]\!]\hat{\rho}\wedge u\hat{\rho}\wedge k\langle\kappa,\langle\zeta,\hat{\rho},\upsilon\rangle\rangle\};$$

$$Q=\lambda E.\mathscr{K}[\![E]\!]=\langle\rangle\vee\bigwedge\{\bigwedge\{c\langle\mathscr{Q}[\![E]\!]\hat{\rho}\kappa\downarrow\nu,\mathscr{Q}[\![E]\!]\zeta\hat{\rho}\upsilon\downarrow\nu\rangle \mid 1\le\nu\le\#\mathscr{K}[\![E]\!]\}$$
$$\mid torn[\![E]\!]\hat{\rho}\wedge u\hat{\rho}\wedge k\langle\kappa,\langle\zeta,\hat{\rho},\upsilon\rangle\rangle\}.$$

Whereas the standard continuation κ supplied to any expression takes exactly one expressed value as an argument, the nature of the environment handed on to a continuation χ reflects the declaration responsible and the valuation adopted. Taking $\nu=0$ to represent the use of $\mathscr{D}$ and $\nu=1$ to represent the use of $\mathscr{T}$ the environment is constrained by

$$knit=\lambda\Delta\nu\hat{\rho}_0\hat{\rho}_1.\bigwedge\{I:\mathscr{I}[\![\Delta]\!]\S\mathscr{K}[\![\Delta]\!]\rightarrow\hat{\rho}_1[\![I]\!]\downarrow 1:E\wedge revert\hat{\rho}_0\hat{\rho}_1[\![I]\!]\downarrow\nu=\hat{\rho}_0[\![I]\!]\downarrow\nu,$$
$$\#\hat{\rho}_1[\![I]\!]=0\wedge revert\hat{\rho}_0\hat{\rho}_1[\![I]\!]=\hat{\rho}_0[\![I]\!]$$
$$\wedge(\#\hat{\rho}_0[\![I]\!]>0\rightarrow\hat{\rho}_1[\![I]\!]\downarrow 1=\hat{\rho}_0[\![I]\!]\downarrow 1,true)\mid I:Ide\}$$
$$\wedge\hat{\rho}_1[\![\mathsf{res}]\!]=\hat{\rho}_0[\![\mathsf{res}]\!]\wedge\hat{\rho}_1[\![\mathsf{rec}]\!]=\hat{\rho}_0[\![\mathsf{rec}]\!].$$

The predicates on declarations corresponding to those set up above for expressions are therefore

$$D=\lambda\Delta.\bigwedge\{c\langle\mathscr{D}[\![\Delta]\!]\hat{\rho}_0\chi,\langle\mathscr{D}[\![\Delta]\!]\zeta,\hat{\rho}_0,\upsilon\rangle\rangle$$
$$\mid rent[\![\Delta]\!]\hat{\rho}_0\wedge u\hat{\rho}_0\wedge\bigwedge\{c\langle\chi\hat{\rho}_1,\langle\zeta,\hat{\rho}_1,\upsilon\rangle\rangle \mid knit[\![\Delta]\!]0\hat{\rho}_0\hat{\rho}_1\wedge u\hat{\rho}_1\}\};$$

$$T=\lambda\Delta.\bigwedge\{c\langle\mathscr{T}[\![\Delta]\!]\hat{\rho}_0\chi,\langle\mathscr{T}[\![\Delta]\!]\zeta,\hat{\rho}_0,\upsilon\rangle\rangle$$
$$\mid torn[\![\Delta]\!]\hat{\rho}_0\wedge u\hat{\rho}_0\wedge\bigwedge\{c\langle\chi\hat{\rho}_1,\langle\zeta,\hat{\rho}_1,\upsilon\rangle\rangle \mid knit[\![\Delta]\!]1\hat{\rho}_0\hat{\rho}_1\wedge u\hat{\rho}_1\}\}.$$

In the next section these predicates will be employed in a proof that the standard equations of appendix 1 are equivalent to certain *new* store equations. These will not be written out explicitly, as they can be derived from appendix 2 merely by substituting *news* for every occurrence of *novels*, viewing *mv* as a primitive which grabs fresh locations with the aid of *new* (rather than the *novel* function defined in 2.1.1) and adopting the valuation $\mathscr{S}$ of 2.2.1. It will be assumed in accordance with 2.2.1 that the standard and store *new* functions are such that $new\sigma = new\vartheta$ whenever the pair ϑ is subject to $s\vartheta = true$.

The proof will proceed by a structural induction on the syntactic constructs of the language which it would be unduly laborious to give in full. Consequently only those lemmata which exhibit the salient features of Mal will be mentioned. Furthermore the fact that paired semantic equations raise the error flag τ under identical conditions will frequently be left unsaid, and the corresponding passage in the proof will be withheld. To deal with the flag, however, $a\langle \tau,\tau\rangle$ will be taken to be *true*; the hypotheses of 2.2.3, as well as the properties of non-terminating computations, demand also that $a\langle \bot,\bot\rangle$ be *true*.

2.2.8. General existence and uniqueness results.

The principles underlying the construction of 2.2.5 can be applied in contexts quite unlike the present one; in particular they have been developed independently by Plotkin [15] for use in a study of λ-definability. Before the proof of equivalence is embarked upon they will therefore be placed in a more general framework. Thus in the following few paragraphs attention will be confined to two domains labelled V and two functors designated by $\mathscr{V}$ for which $\mathscr{V}V$ can be identified with its image under a natural injection into V. In addition it will be convenient to presume that

there is an inclusive predicate v_0 defined on $V^\circ\times V^\circ$ as well as two projections named q_0, each of which maps one of the versions of V into itself. The restriction of the discussion to two domains is, of course, totally superfluous and is adopted merely to conform with the conventions introduced in 2.1.6.

When a' and a'' are inclusive predicates defined on $A^\circ\times A^\circ$ (a product of two domains which may not be identical) and when $\langle q',q'\rangle$ and $\langle q'',q''\rangle$ are two pairs of homonymous projections of the domains called A, the relation $\langle a',\langle q',q'\rangle\rangle\geq\langle a'',\langle q'',q''\rangle\rangle$ will be deemed to hold if and only if for all $\hat{\xi}\in A^\circ\times A^\circ$ $a'\hat{\xi}\supset a''\hat{\xi}$, $a''\hat{\xi}\supset a'((q''^\circ\times q''^\circ)\hat{\xi})$ and $(q'^\circ\times q'^\circ)\hat{\xi}\equiv(q''^\circ\times q''^\circ)\hat{\xi}$. The meaning ascribed to $\supset$ by 1.1.2 makes $\geq$ into a transitive and symmetric ordering.

If $\mathfrak{A}$ is a functor generated from the basic functors of 1.2.8, conventionally $\mathfrak{A}q$ is a projection of $\mathfrak{A}V$ when q is a projection of V. Similarly there may be a function $\mathfrak{a}$ such that $\mathfrak{a}v\langle q,q\rangle$ is an inclusive predicate on $\mathfrak{A}V^\circ\times\mathfrak{A}V^\circ$ when v is an inclusive predicate on $V^\circ\times V^\circ$ and q represents two homonymous projections (one for each domain V). Given a functor $\mathfrak{A}$ such a function $\mathfrak{a}$ will be termed a 'predictor' for $\mathfrak{A}$ based on $\langle v_0,\langle q_0,q_0\rangle\rangle$ if and only if
$\langle\mathfrak{a}v'\langle q',q'\rangle,\langle\mathfrak{A}q',\mathfrak{A}q'\rangle\rangle\geq\langle\mathfrak{a}v''\langle q'',q''\rangle,\langle\mathfrak{A}q'',\mathfrak{A}q''\rangle\rangle\geq\langle\mathfrak{a}v_0\langle q_0,q_0\rangle,\langle\mathfrak{A}q_0,\mathfrak{A}q_0\rangle\rangle$
whenever v' and v'' are inclusive predicates on $V^\circ\times V^\circ$ and q' and q'' are projections of V such that
$\langle v',\langle q',q'\rangle\rangle\geq\langle v'',\langle q'',q''\rangle\rangle\geq\langle v_0,\langle q_0,q_0\rangle\rangle$, $q'\top$ and $q''\top$ are improper, $q'(cut\beta)\supset\perp$ unless $q'\beta=\perp$ and $q''(cut\beta)\supset\perp$ unless $q''\beta=\perp$. Only the first of these conditions on v', v'', q' and q'' is significant, of course, since the others are necessary merely to make sure that the image of a slit continuous lattice is itself slit and continuous.

For the purposes of the present discussion it is necessary

to demand that $\mathfrak{b}$ be a predictor for $\mathfrak{P}$ based on some $\langle v_0,\langle q_0,q_0\rangle\rangle$ for which $\langle \mathfrak{b}v_0\langle q_0,q_0\rangle,\langle \mathfrak{P}q_0,\mathfrak{P}q_0\rangle\rangle \geq \langle v_0,\langle q_0,q_0\rangle\rangle$. Sequences of inclusive predicates and projections are obtained from $\mathfrak{b}$ and $\mathfrak{P}$ by setting $v_{n+1}=\mathfrak{b}v_n\langle q_n,q_n\rangle$ and $q_{n+1}=\mathfrak{P}q_n$ for all $n\geq 0$. Induction shows that $\langle v_{n+1},\langle q_{n+1},q_{n+1}\rangle\rangle \geq \langle v_n,\langle q_n,q_n\rangle\rangle$ for all $n\geq 0$; in fact as the relation $\geq$ is transitive $\langle v_m,\langle q_m,q_m\rangle\rangle \geq \langle v_n,\langle q_n,q_n\rangle\rangle$ when $m\geq n+1$. Under these circumstances $v_n\hat{\beta}\supset v_m((q_n{}^\circ\times q_n{}^\circ)\hat{\beta})$ for every $\hat{\beta}\in V^\circ\times V^\circ$, so taking v to be $\lambda\hat{\beta}.\bigwedge v_n\hat{\beta}$ and q to be $\bigsqcup q_n$ gives a predicate v and a pair of projections $\langle q,q\rangle$ having $\langle v,\langle q,q\rangle\rangle \geq \langle v_n,\langle q_n,q_n\rangle\rangle$ for each $n\geq 0$. Because $\mathfrak{P}$ is continuous q is the least projection for which $q=\mathfrak{P}q$ and $q\sqsupseteq q_0$. Moreover v satisfies an inequality involving the initial predicate and projections, to wit $\langle \mathfrak{b}v\langle q,q\rangle,\langle \mathfrak{P}q,\mathfrak{P}q\rangle\rangle = \langle v,\langle q,q\rangle\rangle \geq \langle v_0,\langle q_0,q_0\rangle\rangle$, and it is the only inclusive predicate subject to this inequality when $\lambda\beta.\beta=\bigsqcup q_n$; the proof that this is so will now be outlined.

If $\hat{\beta}$ is any pair having $\mathfrak{b}v\langle q,q\rangle\hat{\beta}=true$ the nature of $\mathfrak{b}$ conspires with the fact that $\langle v,\langle q,q\rangle\rangle \geq \langle v_n,\langle q_n,q_n\rangle\rangle$ to ensure that $\mathfrak{b}v_n\langle q_n,q_n\rangle\hat{\beta}=true$ (and therefore that $v_{n+1}\hat{\beta}=true$). Together with the knowledge that $v_1\hat{\beta}\supset v_0\hat{\beta}$ this implies that $v\hat{\beta}=true$. Likewise $v\hat{\beta}=\bot$ when $\mathfrak{b}v\langle q,q\rangle\hat{\beta}=\bot$ and $v\hat{\beta}=\top$ when $\mathfrak{b}v\langle q,q\rangle\hat{\beta}=\top$, so $\mathfrak{b}v\langle q,q\rangle\hat{\beta}\supset v\hat{\beta}$ for every $\hat{\beta}\in V^\circ\times V^\circ$.

Conversely, if $\hat{\beta}$ satisfies $v\hat{\beta}=true$ then $v_n\hat{\beta}=true$ for all $n\geq 0$; hence $\mathfrak{b}v\langle q,q\rangle((\mathfrak{P}q_n{}^\circ\times\mathfrak{P}q_n{}^\circ)\hat{\beta})=true$ for all $n\geq 0$, as $\langle v,\langle q,q\rangle\rangle \geq \langle v_n,\langle q_n,q_n\rangle\rangle$ and $\mathfrak{b}$ is a predictor for $\mathfrak{P}$. Since $\mathfrak{b}v\langle q,q\rangle$ is inclusive and $q=\bigsqcup\mathfrak{P}q_n$ this means that $\mathfrak{b}v\langle q,q\rangle((q^\circ\times q^\circ)\hat{\beta})=true$. Similar remarks apply if $v\hat{\beta}=\bot$ or $v\hat{\beta}=\top$ so $v\hat{\beta}\supset\mathfrak{b}v\langle q,q\rangle((q^\circ\times q^\circ)\hat{\beta})$ for every $\hat{\beta}\in V^\circ\times V^\circ$. Consequently v contributes to a solution of the inequality $\langle \mathfrak{b}v\langle q,q\rangle,\langle \mathfrak{P}q,\mathfrak{P}q\rangle\rangle \geq \langle v,\langle q,q\rangle\rangle \geq \langle v_0,\langle q_0,q_0\rangle\rangle$, and $v\circ(q^\circ\times q^\circ)$ must coincide with $\mathfrak{b}v\langle q,q\rangle\circ(q^\circ\times q^\circ)$.

The proof that v is the sole inclusive predicate having $\langle \mathfrak{v}v\langle q,q\rangle,\langle \mathfrak{V}q,\mathfrak{V}q\rangle\rangle=\langle v,\langle q,q\rangle\rangle \geq \langle v_0,\langle q_0,q_0\rangle\rangle$ if $\lambda\beta.\beta=\bigsqcup q_n$ is merely an abstract version of 2.2.6. When v is any inclusive predicate which together with a pair of projections $\langle q,q\rangle$ is subject to this inequality, and when $q=\lambda\beta.\beta$, induction establishes that $\langle v,\langle q,q\rangle\rangle \geq \langle v_n,\langle q_n,q_n\rangle\rangle$ for all $n\geq 0$. Thus for every $\hat{\beta}\in V^{\circ}\times V^{\circ}$ v is such that $v\hat{\beta}\supset\bigwedge v_n\hat{\beta}$ and $\bigwedge v_n\hat{\beta}\supset\bigwedge v((q_n{}^{\circ}\times q_n{}^{\circ})\hat{\beta})$; because $\bigwedge v((q_n{}^{\circ}\times q_n{}^{\circ})\hat{\beta})\supset v(\bigsqcup((q_n{}^{\circ}\times q_n{}^{\circ})\hat{\beta}))$ this particular v satisfies $\langle v,\langle q,q\rangle\rangle \geq \langle \lambda\hat{\beta}.\bigwedge v_n\hat{\beta},\langle \bigsqcup q_n,\bigsqcup q_n\rangle\rangle$, and if $q=\bigsqcup q_n$ then $v=\lambda\hat{\beta}.\bigwedge v_n\hat{\beta}$.

Suppose that $\mathfrak{A}$ and $\mathfrak{B}$ are functors on the category of slit continuous lattices which correspond with predictors $\mathfrak{a}$ and $\mathfrak{b}$ taking inclusive predicates and projections defined on $V^{\circ}\times V^{\circ}$ into inclusive predicates on $\mathfrak{A}V^{\circ}\times\mathfrak{A}V^{\circ}$ and $\mathfrak{B}V^{\circ}\times\mathfrak{B}V^{\circ}$. When $\mathfrak{C}V$ is V, $\mathfrak{A}V\times\mathfrak{B}V$, $\mathfrak{A}V+\mathfrak{B}V$, $\mathfrak{A}V^{\circ}$, $\mathfrak{A}V^{*}$, $\mathfrak{A}V\mapsto\mathfrak{B}V$ or $\mathfrak{A}V\to\mathfrak{B}V$, $\mathfrak{a}$ and $\mathfrak{b}$ induce $\mathfrak{c}$, a predictor for $\mathfrak{C}$ which is based on $\langle v_0,\langle q_0,q_0\rangle\rangle$ when $\mathfrak{a}$ and $\mathfrak{b}$ are based on $\langle v_0,\langle q_0,q_0\rangle\rangle$ (provided that $\mathfrak{A}V$ is flat if $\mathfrak{C}V$ is $\mathfrak{A}V\mapsto\mathfrak{B}V$); thus should $\mathfrak{C}V$ be $\mathfrak{A}V\to\mathfrak{B}V$, for instance, $\mathfrak{c}v\langle q,q\rangle\hat{\theta}$ will be $\bigwedge\{\mathfrak{b}v\langle q,q\rangle\langle \mathfrak{B}q(\hat{\theta}(\mathfrak{A}q\hat{\xi})),\mathfrak{B}q(\hat{\theta}(\mathfrak{A}q\hat{\xi}))\rangle \mid \mathfrak{a}v\langle q,q\rangle\hat{\xi}\wedge(\hat{\xi}:\mathfrak{A}V^{\circ}\times\mathfrak{A}V^{\circ})\}$ for every inclusive predicate v, every pair of projections $\langle q,q\rangle$ and every $\hat{\xi}:\mathfrak{C}V^{\circ}\times\mathfrak{C}V^{\circ}$. Furthermore, when $\mathfrak{A}V$ and $\mathfrak{B}V$ coincide, $\lambda\hat{\xi}.\mathfrak{a}v\langle q,q\rangle\hat{\xi}\vee\mathfrak{b}v\langle q,q\rangle\hat{\xi}$ and $\lambda\hat{\xi}.\mathfrak{a}v\langle q,q\rangle\hat{\xi}\wedge\mathfrak{b}v\langle q,q\rangle\hat{\xi}$ are predictors based on $\langle v_0,\langle q_0,q_0\rangle\rangle$. As even the functors introduced in 2.2.2 yield predictors based on $\langle v_0,\langle q_0,q_0\rangle\rangle$, 2.2.5 can be proved simply by verifying that $\langle v_1,\langle q_1,q_1\rangle\rangle \geq \langle v_0,\langle q_0,q_0\rangle\rangle$; the direct proof has been given purely for pedagogical reasons.

Inclusive predicates will later be required in several other situations, such as that of 2.4.5. Again, however, a direct proof of their existence will be given in preference to a demonstration that the appropriate replacement for $\mathfrak{b}$ is a predictor based on $\langle v_0,\langle q_0,q_0\rangle\rangle$. This decision is dictated by the lengthy

formulae arising from its opposite: ⅋ will be replaced by ⅋, which takes three projections as its arguments and requires a predictor depending on three predicates and six projections.

To provide a final illustration of the power of predictors consider the problem mentioned in 1.5.4, in which it is necessary to relate the effect of valuations to the effect of their conjugates. When the two forms of V involved are B+L*+J+F and B+L*+F an appropriate choice of $\mathfrak{v}$ is provided by taking $\mathfrak{v}v\langle q,q\rangle\hat{\beta}$ to be $v_0\hat{\beta}\wedge(\hat{\beta}:F\times F\to\mathfrak{f}v\langle q,q\rangle\hat{\beta},true)$, where v_0 is roughly as in 2.2.2; if F is [E→G]° and E is L+V then $\mathfrak{f}v\langle q,q\rangle\hat{\phi}$ can be $\bigwedge\{\mathfrak{g}v\langle q,q\rangle\langle\hat{\phi}\,(\mathfrak{C}q\acute{\varepsilon}),\grave{\phi}\,(\mathfrak{C}q\grave{\varepsilon})\rangle\mid\mathfrak{e}v\langle q,q\rangle\hat{\varepsilon}\}$ and $\mathfrak{e}v\langle q,q\rangle\hat{\varepsilon}$ can be $(\hat{\varepsilon}:L\times L\to\mathfrak{l}\hat{\varepsilon},\hat{\varepsilon}:V\times V\to v\langle q\acute{\varepsilon},q\grave{\varepsilon}\rangle,false)$. The two forms of G are K→C and $\{\gamma\in[S\to[E\times S]]\mid(\gamma\perp\downarrow 2=\perp)\wedge(\gamma\top\downarrow 2)=\top\}$ so taking $\mathfrak{G}q$ to be both $\lambda\gamma\kappa\sigma.\gamma\,(\lambda\varepsilon\sigma.\kappa\,(\mathfrak{C}q\varepsilon)\,(\mathfrak{S}q\sigma))\,(\mathfrak{G}q\sigma)$ and $\lambda\gamma\sigma.\,(\mathfrak{C}q\times\mathfrak{G}q)\,(\gamma\,(\mathfrak{S}q\sigma))$ gives

$$\bigvee\{\bigwedge\{(((\lambda\kappa.\mathfrak{G}q\acute{\gamma}\kappa\acute{\sigma}=\lambda\kappa.\,((\lambda\varepsilon\sigma.\kappa(\mathfrak{C}q\varepsilon)(\mathfrak{S}q\sigma))*\psi)(\mathfrak{G}q\acute{\sigma}))$$
$$\wedge\mathfrak{e}v\langle q,q\rangle\langle\psi\,(\mathfrak{S}q\acute{\sigma})\downarrow 1,\grave{\gamma}\,(\mathfrak{S}q\grave{\sigma})\downarrow 1\rangle\wedge\mathfrak{s}v\langle q,q\rangle\langle\psi\,(\mathfrak{S}q\acute{\sigma})\downarrow 2,\grave{\gamma}\,(\mathfrak{S}q\grave{\sigma})\downarrow 2\rangle)$$
$$\vee((\lambda\kappa.\mathfrak{G}q\acute{\gamma}\kappa\acute{\sigma}=\perp)\wedge(\grave{\gamma}\,(\mathfrak{S}q\grave{\sigma})=\perp)\vee((\lambda\kappa.\mathfrak{G}q\acute{\gamma}\kappa\acute{\sigma}=\top)\wedge(\grave{\gamma}\,(\mathfrak{S}q\grave{\sigma})=\top)))$$
$$\mid\mathfrak{s}v\langle q,q\rangle\hat{\sigma}\}\mid\psi\in[S\to[E\times S]]\}$$

as a candidate for $\mathfrak{g}v\langle q,q\rangle\hat{\gamma}$. If the domain A such that K=E→S→A is a continuous lattice then $\mathfrak{g}v\langle q,q\rangle$ is inclusive provided that v is inclusive; in fact $\mathfrak{g}$ is a predictor for $\mathfrak{G}$ based on some $\langle v_0,\langle q_0,q_0\rangle\rangle$ and predicates can be provided for a suitable formulation of 1.5.5. This formulation would allow the restrictions imposed by *crushed* to be reduced by admitting function applications and denotations belonging to G. A further version of the proposition can be obtained by adopting S→[E°×S°] instead of $\{\gamma\in[S\to[E\times S]]\mid(\gamma\perp\downarrow 2=\perp)\wedge(\gamma\top\downarrow 2)=\top\}$ and by amending $\mathfrak{f}$ and $\mathfrak{g}$ somewhat. Precisely the same technique can be applied to the comparison of methods of passing parameters in languages that do not have stores to which assignments can be made.

2.3. Two equivalent formalisms.

2.3.1. Lemma.

If $G[\![E]\!] \wedge P[\![E]\!] \wedge Q[\![E]\!] = true$ then $E[\![E]\!] \wedge L[\![E]\!] \wedge R[\![E]\!] = true$.

⊲Suppose $torn[\![E]\!]\hat{\rho} \wedge u\hat{\rho} \wedge k\langle \kappa, \langle \zeta, \grave{\rho}, \upsilon \rangle \rangle \wedge s\hat{\sigma} = true$, and set $\zeta_0 = \zeta \circ revert\grave{\rho}$ and $\alpha^* = news(\#\mathscr{I}[\![E]\!])\sigma$; in accordance with 2.2.1 assume that $\alpha^* = news(\#\mathscr{I}[\![E]\!])\delta$. Because $a\langle \bot, \bot \rangle \wedge a\langle \top, \top \rangle = true$ the proof that $a\langle \mathscr{E}[\![E]\!]\hat{\rho}\kappa\sigma, \mathscr{E}[\![E]\!]\zeta\grave{\rho}\upsilon\delta \rangle = true$ is trivial unless α^* is proper.

In this case define $\hat{\rho}_1 = \langle \hat{\rho}[\alpha^*/\mathscr{I}[\![E]\!]], \grave{\rho}[\alpha^*/\mathscr{I}[\![E]\!]] \rangle$, $\hat{\rho}_0 = \langle fix(\lambda\rho.\hat{\rho}_1[\mathscr{L}[\![E]\!]\rho\kappa/\mathscr{K}[\![E]\!]]), fix(\lambda\rho.\grave{\rho}_1[\mathscr{L}[\![E]\!]\zeta_0\rho\upsilon/\mathscr{K}[\![E]\!]]) \rangle$ and $\hat{\sigma}_0 = \langle updates\alpha^*(\mathscr{P}[\![E]\!]\hat{\rho}_0\kappa)\sigma, updates\alpha^*(\mathscr{P}[\![E]\!]\zeta_0\grave{\rho}_0\upsilon)\delta \rangle$, so that $\mathscr{E}[\![E]\!]\hat{\rho}\kappa\sigma = \mathscr{G}[\![E]\!]\hat{\rho}_0\kappa\sigma_0$ and $\mathscr{E}[\![E]\!]\zeta\grave{\rho}\upsilon\delta = \mathscr{G}[\![E]\!]\zeta_0\grave{\rho}_0\upsilon\delta_0$. Thus to prove that $a\langle \mathscr{E}[\![E]\!]\hat{\rho}\kappa\sigma, \mathscr{E}[\![E]\!]\zeta\grave{\rho}\upsilon\delta \rangle = true$ it suffices to show that $rent[\![E]\!]\hat{\rho}_0 \wedge u\hat{\rho}_0 \wedge k\langle \kappa, \langle \zeta_0, \grave{\rho}_0, \upsilon \rangle \rangle \wedge s\hat{\sigma}_0 = true$ since $G[\![E]\!] = true$. Certainly $u\hat{\rho}_1 = true$ as α^* is proper; moreover $rent[\![E]\!]\hat{\rho}_0 \wedge k\langle \kappa, \langle \zeta_0, \grave{\rho}_0, \upsilon \rangle \rangle = true$ as $revert\grave{\rho}\grave{\rho}_0 = \grave{\rho}$. Because $P[\![E]\!] = true$, should $u\hat{\rho}_1 = true$ we will have $c\langle \mathscr{P}[\![E]\!]\hat{\rho}_1\kappa\downarrow\nu, \mathscr{P}[\![E]\!]\zeta_0\grave{\rho}_1\upsilon\downarrow\nu \rangle = true$ when $1 \le \nu \le \#\mathscr{I}[\![E]\!]$ and thus $s\hat{\sigma}_0 = true$. Hence it is enough to show that $u\hat{\rho}_1 = true$.

Patently $\hat{\rho}_0 = \bigsqcup\{fun\nu \mid \nu : \mathbb{N}\}$ and $fun\nu \sqsupseteq fun(\nu-1)$ when $\nu \ge 1$, where

$fun = \lambda\nu.\langle \hat{\rho}_1[\nu = 0 \to \bot^*, \mathscr{L}[\![E]\!](fun(\nu-1)\downarrow 1)\kappa/\mathscr{K}[\![E]\!]],$

$\grave{\rho}_1[\nu = 0 \to \bot^*, \mathscr{L}[\![E]\!]\zeta_0(fun(\nu-1)\downarrow 2)\upsilon/\mathscr{K}[\![E]\!]] \rangle$;

because u is inclusive by 2.2.4 to show that $u\hat{\rho}_0 = true$ we need only verify that $u(fun\nu) = true$ when $\nu \ge 0$. Since $d\langle \bot, \bot \rangle \wedge u\hat{\rho}_1 = true$ $u(fun0) = true$; furthermore when $u(fun\nu) = true$, $u(fun(\nu+1)) = true$, since $rent[\![E]\!](fun\nu\downarrow 1) \wedge k\langle \kappa, \langle \zeta_0, fun\nu\downarrow 2, \upsilon \rangle \rangle \wedge Q[\![E]\!] = true$ and $u(fun(\nu+1)) = \bigwedge\{c\langle (fun(\nu+1)\downarrow 1)[\![I]\!]\downarrow 1, (fun(\nu+1)\downarrow 2)[\![I]\!]\downarrow 1 \rangle \mid I : \mathscr{K}[\![E]\!]\}$. Hence $u\hat{\rho}_0 = true$ and $c\langle \mathscr{E}[\![E]\!]\hat{\rho}\kappa, \langle \mathscr{E}[\![E]\!]\zeta, \grave{\rho}, \upsilon \rangle \rangle = true$, δ being an arbitrary suitable pair of stores. As this holds whenever $\hat{\rho}$ and the continuations are appropriate we may deduce that $E[\![E]\!] = true$.

To show now that $L[\![E]\!] \wedge R[\![E]\!] = true$ we now need to demonstrate only that if $k\langle \kappa, \langle \zeta, \rho, \upsilon \rangle \rangle = true$ then

$k\langle lv\kappa,\langle mv\zeta,\rho,\upsilon\rangle\rangle \wedge k\langle rv\kappa,\langle sv\zeta,\rho,\upsilon\rangle\rangle = true$ where the function mv differs from that of 2.1.1 by invoking *new* instead of *novel*. Suppose therefore that $k\langle \kappa,\langle \zeta,\rho,\upsilon\rangle\rangle \wedge e\hat{\varepsilon} \wedge s\hat{\sigma} = true$. Should $\hat{\varepsilon}:L\times L$, $\langle lv\kappa\acute{\varepsilon}\acute{\sigma},mv\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle = \langle \kappa\acute{\varepsilon}\acute{\sigma},\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle$ whilst $\langle rv\kappa\acute{\varepsilon}\acute{\sigma},sv\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle = area\alpha\acute{\varepsilon}\acute{\sigma} \to \langle \kappa(hold\acute{\varepsilon}\acute{\sigma})\acute{\sigma},\zeta\rho(\langle hold\grave{\varepsilon}\grave{\sigma}\rangle \S\upsilon)\sigma\rangle,\langle \top,\top\rangle$; should $\hat{\varepsilon}:V\times V$, writing $\alpha = new\acute{\sigma} = new\grave{\sigma}$ leads to $\langle lv\kappa\acute{\varepsilon}\acute{\sigma},mv\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle = \alpha:L \to \langle \kappa\alpha(update\alpha\acute{\varepsilon}\acute{\sigma}),\zeta\rho(\langle \alpha\rangle \S\upsilon)(update\alpha\grave{\varepsilon}\grave{\sigma})\rangle,\langle \top,\top\rangle$ whilst $\langle rv\kappa\acute{\varepsilon}\acute{\sigma},mv\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle = \langle \kappa\acute{\varepsilon}\acute{\sigma},\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle$. Because $s\hat{\sigma} = true$, in the first case $e\langle hold\acute{\varepsilon}\acute{\sigma},hold\grave{\varepsilon}\grave{\sigma}\rangle = true$ and in the second $s\langle update\alpha\acute{\varepsilon}\acute{\sigma},update\alpha\grave{\varepsilon}\grave{\sigma}\rangle = true$, giving $a\langle lv\kappa\acute{\varepsilon}\acute{\sigma},mv\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle \wedge a\langle rv\kappa\acute{\varepsilon}\acute{\sigma},sv\zeta\rho(\langle \grave{\varepsilon}\rangle \S\upsilon)\grave{\sigma}\rangle = true$. This being so whenever $e\hat{\varepsilon} \wedge s\hat{\sigma} = true$, we can conclude that $k\langle lv\kappa,\langle mv\zeta,\rho,\upsilon\rangle\rangle \wedge k\langle rv\kappa,\langle sv\zeta,\rho,\upsilon\rangle\rangle = true$.⊁

This result reveals what complications are avoided by never passing on $\bot$ or $\top$ as the intermediate result of an erroneous computation. If, for instance, we did not check $news(\#\mathcal{I}[\![E]\!])\acute{\sigma}$ to ensure that it was proper before evaluating $\mathcal{E}[\![E]\!]\acute{\rho}\kappa$ we would need to include such conditions as $\theta\bot = \bot \wedge \zeta\rho\upsilon\bot = \bot$ in the definition of $c\langle \theta,\langle \zeta,\rho,\upsilon\rangle\rangle$. Verifying these extra conditions would be tedious; indeed the recursion operator of 1.4.4 uses a continuation which is not strict and which therefore would not satisfy them.

2.3.2. Lemma.

For all I:Ide and B:Bas $G[\![I]\!] \wedge G[\![B]\!] = true$; when Φ:Abs has a body E:Exp such that $L[\![E]\!] = true$ $G[\![\Phi]\!] = true$.

⊀Suppose $torn[\![I]\!]\acute{\rho} \wedge u\hat{\rho} \wedge k\langle \kappa,\langle \zeta,\grave{\rho},\upsilon\rangle\rangle = true$ and set $\hat{\delta} = \langle \acute{\rho}[\![I]\!]\downarrow 1,\grave{\rho}[\![I]\!]\downarrow 1\rangle$. If $\hat{\delta}:E\times E$ $e\hat{\delta} = true$ and thus $c\langle \kappa\acute{\delta},\langle \zeta,\grave{\rho},\langle \grave{\delta}\rangle \S\upsilon\rangle\rangle = true$, whereas if $\hat{\delta}:G\times G$ $c\langle \acute{\delta}\kappa,\langle \lambda\rho''\upsilon''\sigma''.(\grave{\delta}\downarrow 1)\grave{\zeta}((\grave{\delta}\downarrow 2)[\langle \rho'',\upsilon'',\sigma''\rangle / rec])\langle\rangle(\grave{\delta}\downarrow 3),\grave{\rho},\upsilon\rangle\rangle = true$

and $c(\mathscr{G}[\![\mathrm{I}]\!]\acute{\rho}\kappa,(\mathscr{G}[\![\mathrm{I}]\!]\zeta,\grave{\rho},\upsilon))=true$; the result is immediate if $\hat{\delta}=(\bot,\bot)$ or $\hat{\delta}=(\top,\top)$.

Similarly, if $\hat{\beta}=(\mathscr{B}[\![\mathrm{B}]\!],\mathscr{B}[\![\mathrm{B}]\!])$ $b\hat{\beta}=true$ as $\acute{\beta}=\grave{\beta}$, and thus when $u\hat{\rho}\wedge k(\kappa,(\zeta,\grave{\rho},\upsilon))=true$ we have

$c(\mathscr{G}[\![\mathrm{B}]\!]\acute{\rho}\kappa,(\mathscr{G}[\![\mathrm{B}]\!]\zeta,\grave{\rho},\upsilon))=c(\kappa\acute{\beta},(\zeta,\rho,(\grave{\beta})\,\S\upsilon))=true$

If Φ:Abs we show that $G[\![\Phi]\!]=true$ by verifying that when $rent[\![\Phi]\!]\acute{\rho}\wedge u\acute{\rho}=true$ $e\hat{\varepsilon}=true$ where $\hat{\varepsilon}=(\mathscr{F}[\![\Phi]\!]\acute{\rho},\mathscr{F}[\![\Phi]\!]\grave{\rho})$. Thus suppose that $k(\kappa,(\zeta,\rho,\upsilon))=true$; we wish to show that when $\zeta_0=\zeta\circ revert\rho$ and $\grave{\rho}_0=divert\rho(\grave{\varepsilon}\downarrow 2)$ $k(\lambda\varepsilon.\acute{\varepsilon}\varepsilon\kappa,((\grave{\varepsilon}\downarrow 1)\zeta_0,\grave{\rho}_0,\upsilon))=true$.

Should Φ be fn().E suppose that $e\hat{\varepsilon}_0\wedge s\hat{\sigma}_0=true$ and define $\hat{\varepsilon}_1=\hat{\varepsilon}_0:\mathrm{L}\times\mathrm{L}\to(area\acute{\varepsilon}_0\acute{\sigma}_0\to(hold\acute{\varepsilon}_0\acute{\sigma}_0,hold\grave{\varepsilon}_0\grave{\sigma}_0),(\top,\top)),\hat{\varepsilon}_0$. Now $\acute{\varepsilon}\acute{\varepsilon}_0\kappa\acute{\sigma}_0=\#\acute{\varepsilon}_1|\mathrm{L}^*=0\to\mathscr{L}[\![\mathrm{E}]\!](rend[\![\mathrm{E}]\!]\acute{\rho})\kappa\acute{\sigma}_0,\top$ by 1.5.2 and $(\grave{\varepsilon}\downarrow 1)\zeta_0\grave{\rho}_0\upsilon\grave{\sigma}_0=\#\grave{\varepsilon}_1|\mathrm{L}^*=0\to\mathscr{L}[\![\mathrm{E}]\!]\zeta_0\grave{\rho}_0\upsilon\grave{\sigma}_0,\top$. Because $\#\acute{\varepsilon}_1|\mathrm{L}^*=\#\grave{\varepsilon}_1|\mathrm{L}^*$ always, these are trivially equal unless $\#\acute{\varepsilon}_1|\mathrm{L}^*=0$. In this case, writing $\acute{\rho}_0=rend[\![\mathrm{E}]\!]\acute{\rho}$, $rent[\![\mathrm{E}]\!]\acute{\rho}_0\wedge u\hat{\rho}_0\wedge k(\kappa,(\zeta_0,\grave{\rho}_0,\upsilon))\wedge s\hat{\sigma}_0=true$ since $k(\kappa,(\zeta,\rho,\upsilon))=true$ and $a(\mathscr{L}[\![\mathrm{E}]\!]\acute{\rho}_0\kappa\acute{\sigma}_0,\mathscr{L}[\![\mathrm{E}]\!]\zeta_0\grave{\rho}_0\upsilon\grave{\sigma}_0)=true$ since $L[\![\mathrm{E}]\!]=true$. Thus $c(\mathscr{L}[\![\mathrm{E}]\!]\acute{\rho}_0\kappa,(\mathscr{L}[\![\mathrm{E}]\!]\zeta_0,\grave{\rho}_0,\upsilon))=true$ and $k(\lambda\varepsilon.\acute{\varepsilon}\varepsilon\kappa,((\grave{\varepsilon}\downarrow 1)\zeta_0,\grave{\rho}_0,\upsilon))=true$.

Should Φ be fnI.E, when $e\hat{\varepsilon}_0=true$ $rent[\![\mathrm{E}]\!]\acute{\rho}_1\wedge u\hat{\rho}_1=true$ where $\hat{\rho}_1=((rend[\![\mathrm{E}]\!]\acute{\rho}_0)[\acute{\varepsilon}_0/\mathrm{I}],\grave{\rho}_0[\grave{\varepsilon}_0/\mathrm{I}])$. Moreover $k(\kappa,(\zeta_0,\grave{\rho}_1,\upsilon))=true$ as $k(\kappa,(\zeta,\rho,\upsilon))=true$, so $c(\acute{\varepsilon}\acute{\varepsilon}_0\kappa,((\varepsilon\downarrow 1)\zeta_0,\grave{\rho}_0,(\grave{\varepsilon}_1)\,\S\upsilon))=c(\mathscr{L}[\![\mathrm{E}]\!]\acute{\rho}_1\kappa,(\mathscr{L}[\![\mathrm{E}]\!]\zeta_0,\grave{\rho}_1,\upsilon))=true$ by 1.5.2.

The proof is similar for the other kinds of function.⊁

The comparable results dealing with alterations to the store are less interesting than the corresponding lemmata for other theorems and will therefore be left out. The sole point worthy of note is that in the semantic equations for OE and $\mathrm{E}_0\Omega\mathrm{E}_1$ we apply *rv* to the answer notwithstanding its apparent membership of B already in order to avoid passing $\top$ or $\bot$ (resulting from overflow, say) as an argument to the continuation.

2.3.3. Lemma.

If $R[\![E_0]\!] \wedge L[\![E_1]\!] = true$ then $G[\![E_0 E_1]\!] = true$.

⊲Let $\langle \kappa, \langle \zeta, \rho, \upsilon \rangle \rangle$ be any pair such that for some $\hat{\rho}$ $torn[\![E_0 E_1]\!] \hat{\rho} \wedge u \hat{\rho} \wedge k \langle \kappa, \langle \zeta, \rho, \upsilon \rangle \rangle = true$, and assume that expressions are evaluated from left to right to avoid needless petty complexity in the proof. Define

$\psi_0 = \lambda \varepsilon^*.\varepsilon^* \downarrow 1 : F \to (\varepsilon^* \downarrow 1)(\varepsilon^* \downarrow 2)\kappa, rv(\lambda \beta. 1 \le \beta \le \#\varepsilon^* \downarrow 1 \,|\, L^* \to \kappa(\varepsilon^* \downarrow 1 \downarrow \beta), \tau)(\varepsilon^* \downarrow 2)$

$\zeta_0 = \lambda \rho \upsilon.\upsilon \downarrow 2 : F \to (\upsilon \downarrow 2 \downarrow 1)(\zeta \circ revert\rho)(divert\rho(\upsilon \downarrow 2 \downarrow 2))(\langle \upsilon \downarrow 1 \rangle \S \upsilon \dagger 2), sv\zeta_1 \rho \upsilon$

and $\zeta_1 = \lambda \rho \upsilon. 1 \le \upsilon \downarrow 1 \,|\, N \le \#\upsilon \downarrow 2 \,|\, L^* \to \zeta\rho(\langle \upsilon \downarrow 2 \downarrow (\upsilon \downarrow 1) \rangle \S \upsilon \dagger 2), \tau$.

As $R[\![E_0]\!] = true$ to show that $c\langle \mathscr{G}[\![E_0 E_1]\!] \rho \kappa, \langle \mathscr{G}[\![E_0 E_1]\!] \zeta_0, \rho, \upsilon \rangle \rangle = true$ it suffices to verify that

$k\langle \lambda \varepsilon'.\mathscr{L}[\![E_1]\!] \rho (\lambda \varepsilon''.\psi_0 \langle \varepsilon', \varepsilon'' \rangle), \langle \mathscr{L}[\![E_1]\!] \zeta_0, \rho, \upsilon \rangle \rangle = true$.

Accordingly take any $\hat{\varepsilon}_0$ with $e\hat{\varepsilon}_0 = true$; because $L[\![E_1]\!] = true$, $c\langle \mathscr{L}[\![E_1]\!] \rho (\lambda \varepsilon.\psi_0 \langle \acute{\varepsilon}_0, \varepsilon \rangle), \langle \mathscr{L}[\![E_1]\!] \zeta_0, \rho, \langle \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = true$ when $k\langle \lambda \varepsilon.\psi_0 \langle \acute{\varepsilon}_0, \varepsilon \rangle, \langle \zeta_0, \rho, \langle \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = true$. If $\hat{\varepsilon}_0 : F \times F$, $f\hat{\varepsilon}_1 = true$ and $k\langle \lambda \varepsilon.\acute{\varepsilon}_0 \varepsilon \kappa, \langle (\grave{\varepsilon}_0 \downarrow 1)(\zeta \circ revert\rho), divert\rho(\grave{\varepsilon}_0 \downarrow 2), \upsilon \rangle \rangle = true$ by 2.2.5, so that $k\langle \lambda \varepsilon.\psi_0 \langle \acute{\varepsilon}_0, \varepsilon \rangle, \langle \zeta_0, \rho, \langle \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = true$. Otherwise it is enough to show that $k\langle \kappa_1, \langle \zeta_1, \rho, \langle \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = true$ where $\kappa_1 = \lambda \beta. 1 \le \beta \le \#\varepsilon_0 \,|\, L^* \to \kappa(\acute{\varepsilon}_0 \downarrow \beta), \tau$, since then $k\langle rv\kappa_1, \langle sv\zeta_1, \rho, \langle \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = true$.

If $1 \le \acute{\varepsilon}_1 \,|\, N \le \#\acute{\varepsilon}_0 \,|\, L^*$ for some $\hat{\varepsilon}_1$ having $e\hat{\varepsilon}_1 = true$, $l\langle \acute{\varepsilon}_0 \downarrow \acute{\varepsilon}_1, \grave{\varepsilon}_0 \downarrow \grave{\varepsilon}_1 \rangle = l\langle \acute{\varepsilon}_0 \downarrow \acute{\varepsilon}_1, \grave{\varepsilon}_0 \downarrow \acute{\varepsilon}_1 \rangle = true$ and $c\langle \kappa_1 \acute{\varepsilon}_1, \langle \zeta_1, \rho, \langle \grave{\varepsilon}_1, \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = c\langle \kappa(\acute{\varepsilon}_0 \downarrow \acute{\varepsilon}_1), \langle \zeta, \rho, \langle \grave{\varepsilon}_0 \downarrow \acute{\varepsilon}_1 \rangle \S \upsilon \rangle \rangle = true$. Moreover $c\langle \kappa_1 \acute{\varepsilon}_1, \langle \zeta_1, \rho, \langle \grave{\varepsilon}_1, \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = c\langle \tau, \langle \tau, \rho, \upsilon \rangle \rangle = true$ when a coercion error occurs, so in all possible cases $k\langle \kappa_1, \langle \zeta_1, \rho, \langle \grave{\varepsilon}_0 \rangle \S \upsilon \rangle \rangle = true$ and $c\langle \mathscr{G}[\![E_0 E_1]\!] \rho \kappa, \langle \mathscr{G}[\![E_0 E_1]\!] \zeta, \rho, \upsilon \rangle \rangle = true$. Because $\langle \kappa, \langle \zeta, \rho, \upsilon \rangle \rangle$ is any suitable pair it is apodictic that $G[\![E_0 E_1]\!] = true$.⊳

Next we mention briefly the results about imperative features which change the flow of control.

2.3.4. Lemma.

If $E[\![E]\!]=true$ then $G[\![\text{val } E]\!]\wedge G[\![\text{res } E]\!]\wedge G[\![\text{goto } E]\!]=true$.

⊲By 2.3.1 $L[\![E]\!]\wedge R[\![E]\!]=true$ so it is enough to demonstrate that when we are given some $\hat{\rho}$ and $\langle\kappa,\langle\zeta,\grave{\rho},\upsilon\rangle\rangle$ having $u\hat{\rho}\wedge k\langle\kappa,\langle\zeta,\grave{\rho},\upsilon\rangle\rangle=true$ then $k\langle\kappa,\langle\zeta\circ revert\grave{\rho},\grave{\rho}[\langle\zeta,\grave{\rho},\upsilon\rangle/\text{res}],\upsilon\rangle\rangle=true$ and $k\langle\hat{\rho}[\![\text{res}]\!]\downarrow 1,\langle\zeta_1,\grave{\rho},\upsilon\rangle\rangle\wedge k\langle\lambda\varepsilon.\varepsilon,\langle\zeta_2,\grave{\rho},\upsilon\rangle\rangle=true$ where $\zeta_1=\lambda\rho\upsilon.(\rho[\![\text{res}]\!]\downarrow 1\downarrow 1)(\rho[\![\text{res}]\!]\downarrow 1\downarrow 2)(\langle\upsilon\downarrow 1\rangle\S\rho[\![\text{res}]\!]\downarrow 1\downarrow 3)$ and $\zeta_2=\lambda\rho\upsilon.(\upsilon\downarrow 1\downarrow 1)(\upsilon\downarrow 1\downarrow 2)(\upsilon\downarrow 1\downarrow 3)$. The first of these equalities holds trivially; to validate the second note that $k\langle\hat{\rho}[\![\text{res}]\!]\downarrow 1,\langle\zeta_1,\grave{\rho},\upsilon\rangle\rangle=k\langle\hat{\rho}[\![\text{res}]\!]\downarrow 1,\grave{\rho}[\![\text{res}]\!]\downarrow 1\rangle=true$ whilst when $\hat{\varepsilon}:J\times J$ has $e\hat{\varepsilon}=true$ $c\langle\grave{\varepsilon},\langle\zeta_2,\grave{\rho},\langle\grave{\varepsilon}\rangle\S\upsilon\rangle\rangle=e\hat{\varepsilon}=true$.⊳

It is this result which demonstrates that standard continuations are abstractions of label entry points which preserve the stack rather than of ones consisting merely of a code pointer and an environment. The difference between these kinds of entry points is illustrated by the program $m,x=0,0$ inside $(x:=1+(l:\ m:=l;\ x);$ if $x=1$ then goto m else $x)$. Under standard and store semantics the location denoted by x will ultimately contain 2 if expressions are evaluated from left to right (which implies that $mete=\lambda\xi^*.\xi^*\downarrow 1\circ\ldots\circ\xi^*\downarrow(\#\xi^*)$). If the entry point corresponding to m does not keep the stack as part of its value, however, chaos may ensue on returning to the block and the location denoted by x may finally contain anything. Indeed for this program it would not even help to reset the stack pointer on executing a jump as here the desired stack is higher than that available when the jump is made; only in a language such as Algol 60 which is devoid of stored labels would this mechanism be guaranteed to work. Corroboration for this will be given in 3.3.5, where the stack semantics of Mal will be analysed.

2.3.5. Lemma.

Let Δ be I=E, I==E, $I_1,\ldots,I_n$=E or $I_1,\ldots,I_n$==E for some I or $I_1,\ldots,I_n$ and some E such that $E[\![E]\!]=true$; then $D[\![\Delta]\!]\wedge T[\![\Delta]\!]=true$.

⊲First we show that $D[\![\Delta]\!]=true$ when Δ is I=E or I==E. Suppose that $rent[\![\Delta]\!]\rho\wedge u\hat{\rho}=true$ and that $c\langle\chi\rho_0,\langle\zeta,\grave{\rho}_0,\upsilon\rangle\rangle=true$ whenever $knit[\![\Delta]\!]0\hat{\rho}\hat{\rho}_0\wedge u\hat{\rho}_0=true$. Define $\kappa_0=\lambda\varepsilon.\chi(arid[\varepsilon/I])$ and $\zeta_0=\lambda\rho\upsilon.\zeta\rho[\upsilon\downarrow1/I](\upsilon\uparrow1)$; by 2.3.1 $L[\![E]\!]\wedge R[\![E]\!]=true$ so to show that $c\langle\mathscr{D}[\![\Delta]\!]\rho\chi,\langle\mathscr{D}[\![\Delta]\!]\zeta,\grave{\rho},\upsilon\rangle\rangle=true$ it suffices to verify that $k\langle\kappa_0,\langle\zeta_0,\grave{\rho},\upsilon\rangle\rangle=true$. Taking any $\hat{\varepsilon}$ with $e\hat{\varepsilon}=true$ set $\hat{\rho}_0=\langle arid[\hat{\varepsilon}/I],\grave{\rho}[\grave{\varepsilon}/I]\rangle$; then $knit[\![\Delta]\!]0\hat{\rho}\hat{\rho}_0\wedge u\hat{\rho}_0=true$ and $c\langle\kappa_0\varepsilon,\langle\zeta_0,\grave{\rho},\langle\grave{\varepsilon}\rangle\,\S\upsilon\rangle\rangle=c\langle\chi\acute{\rho}_0,\langle\zeta,\grave{\rho}_0,\upsilon\rangle\rangle=true$. This being so for any suitable $\hat{\rho}$ and $\langle\chi,\langle\zeta,\grave{\rho},\upsilon\rangle\rangle$, $D[\![I=E]\!]\wedge D[\![I==E]\!]=true$.

To establish that $T[\![\Delta]\!]=true$ when Δ is I=E or I==E take any $\hat{\rho}$ and $\langle\chi,\langle\zeta,\grave{\rho},\upsilon\rangle\rangle$ having $torn[\![\Delta]\!]\rho\wedge u\hat{\rho}=true$ and $c\langle\chi\acute{\rho}_0,\langle\zeta,\grave{\rho}_0,\upsilon\rangle\rangle=true$ whenever $knit[\![\Delta]\!]1\hat{\rho}\hat{\rho}_0=true$. Define

$\kappa_1=rv(\lambda\varepsilon.(\lambda\delta.\delta:\mathsf{L}\rightarrow\chi(arid[\delta/I])\circ update\,\delta\varepsilon),\tau)(\acute{\rho}[\![I]\!]\downarrow1))$,

$\kappa_2=rv(\lambda\varepsilon.\chi(arid[\varepsilon/I]))$,

$\zeta_1=sv(\lambda\rho\upsilon.\rho[\![I]\!]\downarrow1:\mathsf{L}\rightarrow\zeta\rho(\upsilon\uparrow1)\circ update(\rho[\![I]\!]\downarrow1)(\upsilon\downarrow1),\tau)$ and

$\zeta_2=sv(\lambda\rho\upsilon.\zeta(invert\rho(arid[\upsilon\downarrow1/I]))(\upsilon\uparrow1))$. As before, we need only verify that $k\langle\kappa_1,\langle\zeta_1,\grave{\rho},\upsilon\rangle\rangle=true$ in order to show that $c\langle\mathscr{T}[\![I=E]\!]\rho\chi,\langle\mathscr{T}[\![I=E]\!]\zeta,\grave{\rho},\upsilon\rangle\rangle=true$; because $\acute{\rho}[\![I]\!]\downarrow1$ is in L if and only if $\grave{\rho}[\![I]\!]\downarrow1$ is in L even this requirement reduces to proving that $knit[\![I=E]\!]1\hat{\rho}\langle arid[\acute{\rho}[\![I]\!]\downarrow1/I],\grave{\rho}\rangle=true$ and that $s\langle update(\acute{\rho}[\![I]\!]\downarrow1)\acute{\beta}\acute{\sigma},update(\grave{\rho}[\![I]\!]\downarrow1)\grave{\beta}\grave{\sigma}\rangle=true$ when $v\hat{\beta}\wedge s\hat{\sigma}=true$. Both these conditions are obviously fulfilled, so $k\langle\kappa_1,\langle\zeta_1,\grave{\rho},\upsilon\rangle\rangle=true$ and, $\hat{\rho}$ being any suitable pair, $T[\![I=E]\!]=true$. Likewise $knit[\![I==E]\!]1\hat{\rho}\langle arid[\acute{\beta}/I],invert\grave{\rho}(arid[\grave{\beta}/I])\rangle=true$ if $v\hat{\beta}=true$, and thus $c\langle\kappa_2\acute{\varepsilon}\acute{\sigma},\zeta_2\grave{\rho}(\langle\grave{\varepsilon}\rangle\,\S\upsilon)\grave{\sigma}\rangle=true$ when $e\hat{\varepsilon}\wedge s\hat{\sigma}=true$; hence $c\langle\mathscr{T}[\![I==E]\!]\rho\chi,\langle\mathscr{T}[\![I==E]\!]\grave{\zeta},\grave{\rho},\upsilon\rangle\rangle=true$ as $R[\![E]\!]=true$.

The proof that $D[\![\Delta]\!]\wedge T[\![\Delta]\!]=true$ when Δ is $I_1,\ldots,I_n$=E or $I_1,\ldots,I_n$==E is only marginally more interesting and can be ignored.⊳

2.3.6. Lemma.

Let Δ_2 be Δ_0 within Δ_1 for some Δ_0 and Δ_1; if $D[\![\Delta_0]\!] \wedge D[\![\Delta_1]\!] = true$ then $D[\![\Delta_2]\!] = true$ whilst if $D[\![\Delta_0]\!] \wedge T[\![\Delta_1]\!] = true$ then $T[\![\Delta_2]\!] = true$.

⊲Suppose that $D[\![\Delta_0]\!] \wedge D[\![\Delta_1]\!] = true$ and let $\hat{\rho}$ be such that $rent[\![\Delta_2]\!]\rho \wedge u\hat{\rho} = true$. To show that $D[\![\Delta_2]\!] = true$ it suffices to prove that for any such $\hat{\rho}$ inevitably $c(\mathscr{D}[\![\Delta_2]\!]\rho\chi, (\mathscr{D}[\![\Delta_2]\!]\zeta, \grave{\rho}, \upsilon)) = true$ if $\bigwedge\{c(\chi\rho_1, (\zeta, \grave{\rho}_1, \upsilon)) \mid knit[\![\Delta_2]\!] 0\hat{\rho}\hat{\rho}_1 \wedge u\hat{\rho}_1\} = true$. Take any $(\chi, (\zeta, \grave{\rho}, \upsilon))$ constrained by this equality, and define $\zeta_0 = \zeta \circ trim[\![\Delta_1]\!]\grave{\rho}$; now $\mathscr{D}[\![\Delta_2]\!]\rho\chi = \mathscr{D}[\![\Delta_0]\!](\lambda\rho.\mathscr{D}[\![\Delta_1]\!](divert\rho\rho)\chi)$ and $\mathscr{D}[\![\Delta_2]\!]\zeta\grave{\rho} = \mathscr{D}[\![\Delta_0]\!](\mathscr{D}[\![\Delta_1]\!]\zeta_0)\grave{\rho}$, so, $D[\![\Delta_0]\!]$ being $true$, it is enough to establish that $c(\mathscr{D}[\![\Delta_1]\!](divert\rho\rho_0)\chi, (\mathscr{D}[\![\Delta_1]\!]\zeta_0, \grave{\rho}_0, \upsilon)) = true$ for all $\hat{\rho}_0$ satisfying $knit[\![\Delta_0]\!] 0\hat{\rho}\hat{\rho}_0 \wedge u\hat{\rho}_0 = true$.

For any such $\hat{\rho}_0$ take any $\hat{\rho}_3$ with $knit[\![\Delta_1]\!] 0\hat{\rho}_0\hat{\rho}_3 \wedge u\hat{\rho}_3 = true$. As demonstrated below (in the proof that $T[\![\Delta_2]\!] = true$), writing $\hat{\rho}_1 = (\rho_3, trim[\![\Delta_1]\!]\grave{\rho}\grave{\rho}_3)$ gives $knit[\![\Delta_1]\!] 0\hat{\rho}\hat{\rho}_1 \wedge u\hat{\rho}_1 = true$, which in turn shows that $knit[\![\Delta_2]\!] 0\hat{\rho}\hat{\rho}_1 = true$, since $\mathscr{I}[\![\Delta_2]\!] = \mathscr{I}[\![\Delta_1]\!]$ and $\mathscr{K}[\![\Delta_2]\!] = \mathscr{K}[\![\Delta_1]\!]$. Hence $c(\chi\rho_3, (\zeta_0, \grave{\rho}_3, \upsilon)) = c(\chi\rho_1, (\zeta, \grave{\rho}_1, \upsilon)) = true$ and, as $\hat{\rho}_3$ is an arbitrary suitable pair and $rent[\![\Delta_1]\!](divert\rho\rho_0) \wedge u(divert\rho\rho_0, \grave{\rho}_0) = true$, $c(\mathscr{D}[\![\Delta_1]\!](divert\rho\rho_0)\chi, (\mathscr{D}[\![\Delta_1]\!]\zeta_0, \grave{\rho}_0, \upsilon)) = true$. Consequently $c(\mathscr{D}[\![\Delta_2]\!]\rho\chi, (\mathscr{D}[\![\Delta_2]\!]\zeta, \grave{\rho}, \upsilon)) = true$ and $D[\![\Delta_2]\!] = true$.

Now suppose instead that $D[\![\Delta_0]\!] \wedge T[\![\Delta_1]\!] = true$; we shall prove that $T[\![\Delta_2]\!] = true$ by letting $\hat{\rho}$ be any environment pair such that $torn[\![\Delta_2]\!]\rho \wedge u\hat{\rho} = true$. Given any $(\chi, (\zeta, \grave{\rho}, \upsilon))$ satisfying $\bigwedge\{c(\chi\rho_1, (\zeta, \grave{\rho}_1, \upsilon)) \mid knit[\![\Delta_2]\!] 1\hat{\rho}\hat{\rho}_1 \wedge u\hat{\rho}_1\} = true$ we define $\zeta_0 = \zeta \circ trim[\![\Delta_1]\!]\grave{\rho}$. As $D[\![\Delta_0]\!] = true$, in order to convince ourselves that $c(\mathscr{T}[\![\Delta_2]\!]\rho\chi, (\mathscr{T}[\![\Delta_2]\!]\zeta, \grave{\rho}, \upsilon)) = true$ we need only establish that $c(\mathscr{T}[\![\Delta_1]\!](divert\rho\rho_0)\chi, (\mathscr{T}[\![\Delta_1]\!]\zeta_0, \grave{\rho}_0, \upsilon)) = true$ whenever $\hat{\rho}_0$ is subject to $knit[\![\Delta_0]\!] 0\hat{\rho}\hat{\rho}_0 \wedge u\hat{\rho}_0 = true$.

For any environment pair $\hat{\rho}_0$ which is such that

$knit⟦\Delta_0⟧0\hat{\rho}\hat{\rho}_0 \wedge u\hat{\rho}_0 = true$ we set $\hat{\rho}_2 = \langle divert\acute{\rho}\acute{\rho}_0, \grave{\rho}_0 \rangle$; then $\langle \acute{\rho}_2⟦I⟧\downarrow 1, \grave{\rho}_2⟦I⟧\downarrow 1 \rangle = \langle \acute{\rho}⟦I⟧\downarrow 1, \grave{\rho}⟦I⟧\downarrow 1 \rangle$ unless $I : \mathcal{I}⟦\Delta_0⟧ \S \mathcal{H}⟦\Delta_0⟧$, and $torn⟦\Delta_1⟧\acute{\rho}_2 \wedge u\hat{\rho}_2 = true$. As $T⟦\Delta_1⟧ = true$ we shall have $c\langle \mathcal{T}⟦\Delta_1⟧\acute{\rho}_2\chi, \langle \mathcal{T}⟦\Delta_1⟧\zeta_0, \grave{\rho}_2, \upsilon \rangle \rangle = true$ if $\langle \chi, \langle \zeta_0, \grave{\rho}_2, \upsilon \rangle \rangle$ is such that $c\langle \chi\acute{\rho}_3, \langle \zeta_0, \grave{\rho}_3, \upsilon \rangle \rangle = true$ when $knit⟦\Delta_1⟧1\hat{\rho}_2\hat{\rho}_3 \wedge u\hat{\rho}_3 = true$. Take any $\hat{\rho}_3$ obeying these constraints and define $\hat{\rho}_1 = \langle \acute{\rho}_3, trim⟦\Delta_1⟧\grave{\rho}\grave{\rho}_3 \rangle$. Unless $I : \mathcal{I}⟦\Delta_2⟧ \S \mathcal{H}⟦\Delta_2⟧$, $revert\hat{\rho}_0\grave{\rho}_3⟦I⟧ = \grave{\rho}_0⟦I⟧$ as $knit⟦\Delta_1⟧1\hat{\rho}_0\hat{\rho}_3 = true$ and $revert\grave{\rho}\grave{\rho}_0⟦I⟧ = \grave{\rho}⟦I⟧$ as $knit⟦\Delta_0⟧0\hat{\rho}\hat{\rho}_0 = true$, so that $revert\grave{\rho}\grave{\rho}_1⟦I⟧ = \grave{\rho}⟦I⟧$; if $I : \mathcal{I}⟦\Delta_2⟧ \S \mathcal{H}⟦\Delta_2⟧$ $revert\grave{\rho}_0\grave{\rho}_3⟦I⟧\dagger 1 = \grave{\rho}_0⟦I⟧\dagger 1$ and $revert\grave{\rho}_0\grave{\rho}_1⟦I⟧\dagger 1 = \grave{\rho}⟦I⟧\dagger 1$. Moreover, unless $I : \mathcal{I}⟦\Delta_2⟧ \S \mathcal{H}⟦\Delta_2⟧$ or $\#\grave{\rho}⟦I⟧ = 0$ the ultimate environment $\hat{\rho}_1$ satisfies

$$\grave{\rho}_1⟦I⟧\downarrow 1 = \#\grave{\rho}_3⟦I⟧ > \#\grave{\rho}⟦I⟧ \rightarrow revert\grave{\rho}\grave{\rho}_3⟦I⟧\downarrow 1, \grave{\rho}_3⟦I⟧\downarrow 1$$
$$= \#\grave{\rho}_3⟦I⟧ > \#\grave{\rho}⟦I⟧ \rightarrow \grave{\rho}⟦I⟧\downarrow 1, \grave{\rho}⟦I⟧\downarrow 1$$

because $\#\grave{\rho}_3⟦I⟧ \geq \#\grave{\rho}_0⟦I⟧ \geq \#\grave{\rho}⟦I⟧$.

In consequence, $knit⟦\Delta_1⟧1\hat{\rho}\hat{\rho}_1 \wedge u\hat{\rho}_1 = true$, giving $c\langle \chi\acute{\rho}_3, \langle \zeta_0, \grave{\rho}_3, \upsilon \rangle \rangle = c\langle \chi\acute{\rho}_1, \langle \zeta, \grave{\rho}_1, \upsilon \rangle \rangle = true$ and $c\langle \mathcal{T}⟦\Delta_1⟧(divert\acute{\rho}\acute{\rho}_0)\chi, \langle \mathcal{T}⟦\Delta_1⟧\zeta_0, \grave{\rho}_0, \upsilon \rangle \rangle = true$ whenever $\hat{\rho}_0$ conforms to $knit⟦\Delta_0⟧0\hat{\rho}\hat{\rho}_0 \wedge u\hat{\rho}_0 = true$. This demonstrates that $c\langle \mathcal{T}⟦\Delta_2⟧\acute{\rho}\chi, \langle \mathcal{T}⟦\Delta_2⟧\zeta, \grave{\rho}, \upsilon \rangle \rangle = true$ for all $\hat{\rho}$ and $\langle \chi, \langle \zeta, \grave{\rho}, \upsilon \rangle \rangle$ such that $torn⟦\Delta_2⟧\acute{\rho} \wedge u\hat{\rho} = true$ and $\bigwedge\{c\langle \chi\acute{\rho}_1, \langle \zeta, \grave{\rho}_1, \upsilon \rangle \rangle \mid knit⟦\Delta_2⟧1\hat{\rho}\hat{\rho}_1 \wedge u\hat{\rho}_1\} = true$, so in accordance with 2.2.7 we may deduce that $T⟦\Delta_2⟧ = true$.⊁

A proof that $G⟦\Delta_0$ inside $E_0⟧ = true$ when $D⟦\Delta_0⟧ \wedge L⟦E_0⟧ = true$ would follow the lines of that above quite closely but would be less complex, as it would contain no discussion of $knit⟦\Delta_1⟧0$.

2.3.7. Lemma.

Let Δ_0 be Δ_1 and...and Δ_n for some $\Delta_1, \ldots, \Delta_n$; if $D⟦\Delta_1⟧ \wedge \ldots \wedge D⟦\Delta_n⟧ = true$ then $D⟦\Delta_0⟧ = true$ whilst if $T⟦\Delta_1⟧ \wedge \ldots \wedge T⟦\Delta_n⟧ = true$ then $T⟦\Delta_0⟧ = true$.

⊲The proof proceeds by induction; plainly the result holds when $n=1$, so suppose that it holds for all sets of m declarations with $n>m$. By renaming the n given declarations if necessary, we can stipulate that the i, j and k of 1.3.5 induce the *run* and *deal* functions corresponding to evaluation from left to right.

Thus assume first that $D[\![\Delta_1]\!]\wedge\ldots\wedge D[\![\Delta_n]\!]=true$, so that by the induction hypothesis $D[\![\Delta_{n+1}]\!]=true$ where Δ_{n+1} is Δ_2 and...and Δ_n. Take any $\hat{\rho}$ and $\langle\chi,\langle\zeta,\grave{\rho},\upsilon\rangle\rangle$ such that $rent[\![\Delta_0]\!]\acute{\rho}\wedge u\hat{\rho}=true$ and $c\langle\chi\acute{\rho}_0,\langle\zeta,\grave{\rho}_0,\upsilon\rangle\rangle=true$ whenever $knit[\![\Delta_0]\!]0\hat{\rho}\hat{\rho}_0\wedge u\hat{\rho}_0=true$. Suppose that $\hat{\rho}_1$ satisfies $knit[\![\Delta_1]\!]0\hat{\rho}\hat{\rho}_1\wedge u\hat{\rho}_1=true$, and define $\grave{\rho}_2=clip[\![\Delta_1]\!]\grave{\rho}\grave{\rho}_1$, $\chi_0=\chi\circ divert\acute{\rho}_1$ and $\zeta_0=\zeta\circ pick[\![\Delta_1]\!]\langle\grave{\rho},\grave{\rho}_2\rangle$. Then if $\#\acute{\rho}[\![I]\!]>0$ for any $I:Ide$, we know that $\#\grave{\rho}[\![I]\!]>0$ and that

$$\begin{aligned}\grave{\rho}_2[\![I]\!]\downarrow 1&=I:\mathscr{I}[\![\Delta_1]\!]\,\S\mathscr{A}[\![\Delta_1]\!]\rightarrow revert\grave{\rho}\grave{\rho}_1[\![I]\!]\downarrow 1,\grave{\rho}_1[\![I]\!]\downarrow 1\\&=I:\mathscr{I}[\![\Delta_1]\!]\,\S\mathscr{A}[\![\Delta_1]\!]\rightarrow\grave{\rho}[\![I]\!]\downarrow 1,\grave{\rho}[\![I]\!]\downarrow 1\end{aligned}$$

as $knit[\![\Delta_1]\!]0\hat{\rho}\hat{\rho}_1=true$. Hence $rent[\![\Delta_{n+1}]\!]\acute{\rho}\wedge u\langle\acute{\rho},\grave{\rho}_2\rangle=true$ and once we have verified that $\bigwedge\{c\langle\chi_0\acute{\rho}_3,\langle\zeta_0,\grave{\rho}_3,\upsilon\rangle\rangle\mid knit[\![\Delta_{n+1}]\!]0\hat{\rho}_2\hat{\rho}_3\wedge u\hat{\rho}_3\}=true$ we shall have the result that

$c\langle\mathscr{D}[\![\Delta_{n+1}]\!]\acute{\rho}\chi_0,\langle\mathscr{D}[\![\Delta_{n+1}]\!]\zeta_0,\grave{\rho}_2,\upsilon\rangle\rangle=true$.

Take any $\hat{\rho}_3$ with $knit[\![\Delta_{n+1}]\!]0\hat{\rho}_2\hat{\rho}_3\wedge u\hat{\rho}_3=true$; writing $\hat{\rho}_0=\langle divert\acute{\rho}_1\acute{\rho}_3,pick[\![\Delta_1]\!]\langle\grave{\rho},\grave{\rho}_2\rangle\grave{\rho}_3\rangle$, for any I $revert\grave{\rho}\grave{\rho}_0[\![I]\!]=revert\grave{\rho}\grave{\rho}_3[\![I]\!]=revert\grave{\rho}(revert\grave{\rho}_1\grave{\rho}_3)[\![I]\!]=\grave{\rho}[\![I]\!]$ since $knit[\![\Delta_1]\!]0\hat{\rho}\hat{\rho}_1\wedge knit[\![\Delta_{n+1}]\!]0\hat{\rho}_2\hat{\rho}_3=true$. Moreover, unless $I:\mathscr{I}[\![\Delta_0]\!]\,\S\mathscr{A}[\![\Delta_0]\!]$ or $\#\grave{\rho}[\![I]\!]=0$ necessarily

$\grave{\rho}_0[\![I]\!]\downarrow 1=\grave{\rho}_3[\![I]\!]\downarrow 1=\grave{\rho}_2[\![I]\!]\downarrow 1=\grave{\rho}_1[\![I]\!]\downarrow 1=\grave{\rho}[\![I]\!]\downarrow 1$, whilst when $I:\mathscr{I}[\![\Delta_1]\!]\,\S\mathscr{A}[\![\Delta_1]\!]$ we must have

$$\begin{aligned}\grave{\rho}_0[\![I]\!]\downarrow 1&=\#\grave{\rho}[\![I]\!]>0\rightarrow revert\grave{\rho}_2\grave{\rho}_3[\![I]\!]\downarrow 2,\grave{\rho}_3[\![I]\!]\downarrow 1\\&=\#\grave{\rho}[\![I]\!]>0\rightarrow\grave{\rho}_1[\![I]\!]\downarrow 1,\#\grave{\rho}_2[\![I]\!]>0\rightarrow\grave{\rho}_1[\![I]\!]\downarrow 1,\grave{\rho}_3[\![I]\!]\downarrow 1\\&=\grave{\rho}_1[\![I]\!]\downarrow 1\end{aligned}$$

since $\#\grave{\rho}_2[\![I]\!]\geq\#\grave{\rho}_1[\![I]\!]\geq\#\grave{\rho}[\![I]\!]>0$ and $u\hat{\rho}_1=true$. Consequently $knit[\![\Delta_0]\!]0\hat{\rho}\hat{\rho}_0\wedge u\hat{\rho}_0=true$ and $c\langle\chi_0\acute{\rho}_3,\langle\zeta_0,\grave{\rho}_3,\upsilon\rangle\rangle=true$ for all suitable

environment pairs $\hat{\rho}_3$.

Hence whenever $knit[\![\Delta_1]\!]0\hat{\rho}\hat{\rho}_1{}^\wedge u\hat{\rho}_1 = true$ $c\langle\mathscr{D}[\![\Delta_{n+1}]\!]\delta(\chi\circ divert\delta_1),\langle\zeta_1,\grave{\rho}_1,\upsilon\rangle\rangle = true$ where $\zeta_1 = (\lambda\rho.\mathscr{D}[\![\Delta_{n+1}]\!](\zeta\circ pick[\![\Delta_1]\!]\langle\grave{\rho},\rho\rangle))\circ clip[\![\Delta_1]\!]\grave{\rho}$. Because $D[\![\Delta_1]\!] = true$ this implies that $c\langle\mathscr{D}[\![\Delta_1]\!]\delta(\lambda\rho.\mathscr{D}[\![\Delta_{n+1}]\!]\delta(\chi\circ divert\rho)),\langle\mathscr{D}[\![\Delta_1]\!]\zeta_1,\grave{\rho},\upsilon\rangle\rangle = true$. Finally $\lambda\rho.divert\rho\circ conserve = \lambda\rho\rho^*.conserve(\langle\rho\rangle\S\rho^*)$ and from 2.1.5 $\lambda\rho\rho^*.pick[\![\Delta_1]\!]\langle\grave{\rho},\rho\rangle\circ pick[\![\Delta_{n+1}]\!](\langle\rho\rangle\S\rho^*) = \lambda\rho\rho^*.pick[\![\Delta_0]\!](\langle\grave{\rho},\rho\rangle\S\rho^*)$ as $\mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]$ has no repeated members, so $c\langle\mathscr{D}[\![\Delta_0]\!]\delta\chi,\langle\mathscr{D}[\![\Delta_0]\!]\zeta,\grave{\rho},\upsilon\rangle\rangle = true$ and, $\langle\chi,\langle\zeta,\grave{\rho},\upsilon\rangle\rangle$ being any suitable pairing, $D[\![\Delta_0]\!] = true$.

The proof that $T[\![\Delta_0]\!] = true$ when $T[\![\Delta_1]\!]\wedge\ldots\wedge T[\![\Delta_n]\!] = true$ uses $knit[\![\Delta_m]\!]1$ instead of $knit[\![\Delta_m]\!]0$ when $1\leq m\leq n+1$ but is too similar to the foregoing to be worth giving.⊁

2.3.8. Lemma.

If $T[\![\Delta]\!] = true$ then $D[\![\text{rec }\Delta]\!]\wedge T[\![\text{rec }\Delta]\!] = true$.

⊰Suppose that $rent[\![\Delta]\!]\delta_1{}^\wedge u\hat{\rho}_1 = true$ and that $c\langle\chi\delta_0,\langle\zeta,\grave{\rho}_0,\upsilon\rangle\rangle = true$ whenever $knit[\![\Delta]\!]0\hat{\rho}_1\hat{\rho}_0{}^\wedge u\hat{\rho}_0 = true$. We wish to show that $c\langle\mathscr{D}[\![\text{rec }\Delta]\!]\delta_1\chi,\langle\mathscr{D}[\![\text{rec }\Delta]\!]\zeta,\grave{\rho}_1,\upsilon\rangle\rangle = true$; to this end given any δ_1 with $s\delta_1 = true$ we set $\alpha^* = news(\#\mathscr{I}[\![\Delta]\!])\sigma_1$, $\delta_2 = \langle updates\alpha^*dummy^*\sigma_1, updates\alpha^*dummy^*\delta_1\rangle$ and

$\hat{\rho}_2 = \langle fix(\lambda\rho.\delta_1[\alpha^*/\mathscr{I}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho\delta_2/\mathscr{K}[\![\Delta]\!]]),$
$\quad fix(\lambda\rho.\grave{\rho}_1[\alpha^*/\mathscr{I}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho\delta_2/\mathscr{K}[\![\Delta]\!]])\rangle$.

Certainly if α^* is improper we know that $a\langle\mathscr{D}[\![\text{rec }\Delta]\!]\delta_1\chi\sigma_1,\mathscr{D}[\![\text{rec }\Delta]\!]\zeta\grave{\rho}_1\upsilon\delta_1\rangle = true$, so we need consider only the proper case. Then $\mathscr{D}[\![\text{rec }\Delta]\!]\delta_1\chi\sigma_1 = \mathscr{T}[\![\Delta]\!]\delta_2\chi\sigma_2$ and $\mathscr{D}[\![\text{rec }\Delta]\!]\zeta\grave{\rho}_1\upsilon\delta_1 = \mathscr{T}[\![\Delta]\!]\grave{\zeta}\rho_2\upsilon\grave{\sigma}_2$, so it suffices to establish that $torn[\![\Delta]\!]\delta_2{}^\wedge u\hat{\rho}_2{}^\wedge s\delta_2 = true$ and that $c\langle\chi\delta_0,\langle\zeta,\grave{\rho}_0,\upsilon\rangle\rangle = true$ whenever $knit[\![\Delta]\!]1\hat{\rho}_2\hat{\rho}_0{}^\wedge u\hat{\rho}_0 = true$. For any such $\hat{\rho}_0$ $knit[\![\Delta]\!]0\hat{\rho}_1\hat{\rho}_0 = true$, as the definitions of 1.3.2 reveal that

$revert\grave{\rho}_1\grave{\rho}_0[\![I]\!] = revert\grave{\rho}_2\grave{\rho}_0[\![I]\!]\dagger(I:\mathscr{I}[\![\Delta]\!]\S\mathscr{K}[\![\Delta]\!]\to 1,0)$

$= \grave{\rho}_2[\![I]\!]\dagger(I:\mathscr{I}[\![\Delta]\!]\S\mathscr{K}[\![\Delta]\!]\to 1,0)$

$= \grave{\rho}_1[\![I]\!]$

for every I:Ide; hence all the conditions other than that $u\hat{\rho}_2 = true$ are satisfied trivially.

Define

$fun = \lambda\nu.\langle \acute{\rho}_1[\alpha^*/\mathscr{I}[\![\Delta]\!]][\nu=0\to\iota^*, \mathscr{K}[\![\Delta]\!](fun(\nu\text{-}1)\downarrow 1)\acute{\sigma}_2/\mathscr{K}[\![\Delta]\!]],$

$\grave{\rho}_1[\alpha^*/\mathscr{I}[\![\Delta]\!]][\nu=0\to\iota^*, \mathscr{S}[\![\Delta]\!](fun(\nu\text{-}1)\downarrow 2)\grave{\sigma}_2/\mathscr{K}[\![\Delta]\!]]\rangle$;

then $\hat{\rho}_2 = \bigsqcup\{fun\nu \mid \nu:\mathbb{N}\}$ and $fun\nu \sqsupseteq fun(\nu\text{-}1)$ when $\nu\geq 1$, so as u is inclusive we have only to show that $u(fun\nu) = true$ for all $\nu\geq 0$. From 2.2.7 it is plain that $u(fun0) = true$, so the basis for an inductive proof has been established.

⊲Assume that $u(fun\nu) = true$ for some $\nu\geq 0$; once we have verified that $g\langle (fun(\nu+1)\downarrow 1)[\![I]\!]\downarrow 1, (fun(\nu+1)\downarrow 2)[\![I]\!]\downarrow 1\rangle = true$ when $I:\mathscr{K}[\![\Delta]\!]$ we shall know that $u(fun(\nu+1)) = true$. Define $\hat{\rho}_4 = \langle tear[\![\Delta]\!](fun\nu\downarrow 1), tear[\![\Delta]\!](fun\nu\downarrow 2)\rangle$; in standard semantics $\mathscr{T}[\![\Delta]\!] = \mathscr{T}[\![\Delta]\!]\circ tear[\![\Delta]\!]$ by 1.5.2 and thus $\mathscr{S}[\![\Delta]\!] = \mathscr{S}[\![\Delta]\!]\circ tear[\![\Delta]\!]$ from 1.4.4. Hence when $I:\mathscr{K}[\![\Delta]\!]$

$\langle (fun(\nu+1)\downarrow 1)[\![I]\!]\downarrow 1, (fun(\nu+1)\downarrow 2)[\![I]\!]\downarrow 1\rangle = \langle \gamma_0, \langle \xi_0, \hat{\rho}_4, \grave{\sigma}_2\rangle\rangle$ where

$\gamma_0 = \lambda\kappa\sigma.\mathscr{T}[\![\Delta]\!]\acute{\rho}_4(\lambda\rho\sigma'.\kappa(\rho[\![I]\!]\downarrow 1 \mid E)\sigma)\acute{\sigma}_2$ and

$\xi_0 = \mathscr{T}[\![\Delta]\!]\circ(\lambda\zeta\rho\upsilon\sigma.(\lambda\pi'.\zeta(\pi'\downarrow 1)(\langle \rho[\![I]\!]\downarrow 1\rangle \S\pi'\downarrow 2)(\pi'\downarrow 3))(\rho[\![rec]\!]\downarrow 1))$.

Here we use the valuation $\mathscr{S}$ for *new* store semantics given in 2.2.1, not that of appendix 1.

⊲To establish that $g\langle \gamma_0, \langle \xi_0, \grave{\rho}_4, \grave{\sigma}_2\rangle\rangle = true$ take any $\langle \kappa_0, \langle \xi_0, \grave{\rho}_5, \upsilon_5\rangle\rangle$ having $k\langle \kappa_0, \langle \xi_0, \grave{\rho}_5, \upsilon_5\rangle\rangle = true$ and any $\hat{\sigma}_3$ having $s\hat{\sigma}_3 = true$. Define $\chi_1 = \lambda\rho\sigma.\kappa_0(\rho[\![I]\!]\downarrow 1 \mid E)\acute{\sigma}_3$ and

$\zeta_1 = \lambda\rho\upsilon\sigma.(\lambda\pi'.\zeta(\pi'\downarrow 1)(\langle \rho[\![I]\!]\downarrow 1\rangle \S\pi'\downarrow 2)(\pi'\downarrow 3))(\rho[\![rec]\!]\downarrow 1)$.

For all pairs $\hat{\sigma}_4$ and $\hat{\rho}_6$ constrained by $s\hat{\sigma}_4 = true$ and $u\hat{\rho}_6 \wedge knit[\![\Delta]\!]1\hat{\rho}_4\hat{\rho}_6 = true$, $e\langle \acute{\rho}_6[\![I]\!]\downarrow 1, \grave{\rho}_6[\![I]\!]\downarrow 1\rangle = true$ and $a\langle \chi_1\acute{\rho}_7\acute{\sigma}_4, \zeta_1\grave{\rho}_7\langle\rangle\grave{\sigma}_4\rangle = a\langle \kappa_0(\acute{\rho}_6[\![I]\!]\downarrow 1)\acute{\sigma}_3, \zeta_0\grave{\rho}_5(\langle \grave{\rho}_6[\![I]\!]\downarrow 1\rangle \S\upsilon_5)\grave{\sigma}_3\rangle = true$ where $\hat{\rho}_7 = \langle \acute{\rho}_6, \grave{\rho}_6[\langle \grave{\rho}_5, \upsilon_5, \grave{\sigma}_3\rangle/rec]\rangle$. Hence $c\langle \chi_1\acute{\rho}_7, \langle \zeta_1, \grave{\rho}_7, \langle\rangle\rangle\rangle = true$

for any $\hat{\rho}_7$ having $u\hat{\rho}_7 \wedge knit[\![\Delta]\!]1\langle \acute{\rho}_4, \grave{\rho}_4[\langle \grave{\rho}_5, \upsilon_5, \grave{\sigma}_3\rangle / rec]\rangle = true$, and, $T[\![\Delta]\!]$ being *true*, we can infer that $c\langle \mathscr{T}[\![\Delta]\!]\acute{\rho}_4\chi_1, \langle \mathscr{T}[\![\Delta]\!]\zeta_1, \grave{\rho}_4[\langle \grave{\rho}_5, \upsilon_5, \grave{\sigma}_3\rangle / rec], \langle\rangle\rangle = true$. Since $s\hat{\sigma}_2 = true$ we even know that $a\langle \gamma_0\kappa_0\sigma_3, \xi_0\zeta_0\grave{\rho}_4[\langle \grave{\rho}_5, \upsilon_5, \grave{\sigma}_3\rangle / rec]\langle\rangle \grave{\sigma}_2\rangle = true$; this being so for all pairs having $k\langle \kappa_0, \langle \zeta_0, \grave{\rho}_5, \upsilon_5\rangle\rangle = true$ and $s\hat{\sigma}_3 = true$ we can safely assert that $g\langle \gamma_0, \langle \xi_0, \grave{\rho}_4, \grave{\sigma}_2\rangle\rangle = true$.⊁

Because I is a typical member of $\mathscr{N}[\![\Delta]\!]$, $\bigwedge\{g\langle (fun(\nu+1)\downarrow 1)[\![I]\!]\downarrow 1, (fun(\nu+1)\downarrow 2)[\![I]\!]\downarrow 1\rangle \mid I : \mathscr{N}[\![\Delta]\!]\} = true$. In addition $\bigwedge\{gyven\langle \varepsilon, \varepsilon\rangle \langle \alpha^*, \alpha^*\rangle \rightarrow l\langle \varepsilon, \varepsilon\rangle, true \mid \varepsilon : E\} = true$ and $u\hat{\rho}_1 = true$, so $u(fun(\nu+1)) = true$.⊁

Now u is inclusive and $u(fun\nu) = true$ for all $\nu \geq 0$; in consequence $u\hat{\rho}_2 = true$ and, as $torn[\![\Delta]\!]\acute{\rho}_2$ and $s\hat{\sigma}_2$ are *true*, $a\langle \mathscr{T}[\![\Delta]\!]\acute{\rho}_2\chi\acute{\sigma}_2, \mathscr{T}[\![\Delta]\!]\zeta\grave{\rho}_2\upsilon\grave{\sigma}_2\rangle = true$. Hence for every $\langle \chi, \langle \zeta, \grave{\rho}_1, \upsilon\rangle\rangle$, $\hat{\rho}_1$ and $\hat{\sigma}_1$ subject to $rent[\![\Delta]\!]\acute{\rho}_1 \wedge u\hat{\rho}_1 \wedge s\hat{\sigma}_1 = true$ and to $\bigwedge\{c\langle \chi\acute{\rho}_0, \langle \zeta, \grave{\rho}_0, \upsilon\rangle\rangle \mid knit[\![rec\ \Delta]\!]0\hat{\rho}_1\hat{\rho}_0 \wedge u\hat{\rho}_0\} = true$ we have shown that $a\langle \mathscr{D}[\![rec\ \Delta]\!]\acute{\rho}_1\chi\acute{\sigma}_1, \mathscr{D}[\![rec\ \Delta]\!]\zeta\grave{\rho}_1\upsilon\grave{\sigma}_1\rangle = true$, and we may conclude that Δ satisfies $D[\![rec\ \Delta]\!] = true$.

That $T[\![rec\ \Delta]\!] = true$ is an immediate outcome of the manner in which the predicates of 2.2.7 can be combined.⊁

2.3.9. Theorem.

The meanings of a Mal program provided by standard semantics and by *new* store semantics are equivalent so long as the *new* functions chosen coincide on equivalent stores.

⊀This is simply a summary of the foregoing results in which we take for granted such propositions as that if $R[\![E_0]\!] \wedge G[\![E_1]\!] \wedge P[\![E_1]\!] \wedge Q[\![E_1]\!] = true$ then $G[\![E_2]\!] \wedge P[\![E_2]\!] \wedge Q[\![E_2]\!] = true$, where E_2 is while E_0 do E_1; 2.5.6 will discuss a contention akin to this. Because our proof has not needed any special features of Mal it is plain that a similar theorem will hold for other languages which can be described in standard terms.⊁

2.4. Reflexive projections.

2.4.1. Links between machine states.

Having proved that *new* store semantics and its standard counterpart are equivalent we can return to the original purpose for which the former was devised: to employ the algorithm of 2.1.6 to trace out the locations accessible from a program, thereby linking I==E and I=E. Here we shall develop means for comparing two state vectors in which some values appearing in the environment of one correspond to values in the store of the other. The outcome of this will be a theorem to the effect that when a location denoted by an identifier is never updated an object having affinities with the content may be denoted instead. This is a more profound property of programs than it may appear at first sight, for the connection between I::E and I:E is not perspicuous and that between rec I==E and rec I=E need not exist at all, as is indicated by 1.4.2. Indeed the complexities of recursion are such that whereas labels will be analysed adequately by 2.5.1 a full treatment of declarations must be deferred to 2.7.7.

Because results such as 2.5.8 will be couched in terms of *novel* store semantics, before drawing conclusions about the standard formalism we must relate the two kinds of store semantics. The lemmata required for this will be almost identical with those leading up to 2.5.9, and we shall not scruple to leave out many of them nor to use their twins in the proof of 2.5.9 without establishing them first.

The pairs of state vectors with which we shall be concerned are those arrived at by following through the execution of a program and its transform under the rules of 1.4.6. Thus we shall need a relation a (mapping pairs in $A^\circ \times A^\circ$ to truth values) to act on $\delta[\![E]\!]\zeta\rho\sigma\delta$ and $\delta[\![\blacktriangleleft[\![E]\!]\psi]\!]\zeta\rho\sigma\delta$ for every ψ:Ide↦B* and E:Exp and for every apposite $\hat{\zeta}$ and $\hat{\pi}$ (using the notation of 2.1.6). If π and $\hat{\pi}$

have themselves been obtained by evaluating an expression and its transform, at the very least they will satisfy $apt\psi\hat{\rho}=true$ and be constrained by some global relation of similarity which we signify by p_0. In addition those values $\acute{\omega}$ and $\grave{\omega}$ which can be reached through *seen* from starting points that tally in $\acute{\pi}$ and $\grave{\pi}$ will be comparable in some sense; consequently they will obey $w\hat{\omega}\hat{\pi}=true$ for a certain predicate w which in principle may depend on $\hat{\pi}$, the member of $P^{\circ}\times P^{\circ}$ being investigated. We therefore expect to be interested in some p of the form $p=\lambda\hat{\pi}.p_0\hat{\pi}\wedge\bigwedge\{w\hat{\omega}\hat{\pi}\,|\,kent0\hat{\omega}\hat{\pi}\}$.

Continuations $\acute{\zeta}$ and $\grave{\zeta}$ in Z are equivalent if when applied to equivalent states they produce results satisfying some given relation a. We might therefore hope to use some c such that $c\hat{\zeta}\wedge p\hat{\pi}\supset a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle$, but this is not possible. Built into the environments applied must be a particular configuration of locations and other denoted values which reflects how the continuations will act; should $\langle\acute{\rho}[\![I]\!]\downarrow 1,\grave{\rho}[\![I]\!]\downarrow 1\rangle$ be in V×L, for example, when the code for a program manipulates I that for its transform must manipulate $I. We thus write $c\hat{\zeta}\hat{\pi}_0$ rather than $c\hat{\zeta}$ and insist that $\hat{\rho}$ be compatible with $\hat{\rho}_0$ in some respect if $a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle$ is to be *true*. This consonance between $\hat{\rho}$ and $\hat{\rho}_0$ must ensure that $apt\psi\hat{\rho}=true$ if and only if $apt\psi\hat{\rho}_0=true$, so we set

$$fit=\lambda\hat{\pi}_0\hat{\pi}_1.p_1\hat{\pi}_1$$
$$\wedge\bigwedge\{\hat{e}:L\times L\vee\hat{e}:V\times V\vee\sim gyven\hat{e}0_1\,|\,\hat{e}:E\times E\}$$
$$\wedge(\mathfrak{U}q_0\times\mathfrak{U}q_0)\hat{\rho}_0=(\mathfrak{U}q_0\times\mathfrak{U}q_0)\hat{\rho}_1\wedge(\mathfrak{P}q_0\times\mathfrak{P}q_0)0_0=(\mathfrak{P}q_0\times\mathfrak{P}q_0)0_1.$$

Here q_0 is a projection of W into itself which is 'sufficiently large' to discriminate between members of L, V and G, while $\mathfrak{U}$ and $\mathfrak{P}$ are functors such that $\mathfrak{U}W$ and $\mathfrak{P}W$ are composed of environments and stacks respectively. Anticipating 2.4.8 we can now write $c=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge\{a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle\,|\,p\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0\}$.

To ensure that there exists at least one pair $\hat{\pi}$ with

$p\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0=true$ when $fit\hat{\pi}_0\hat{\pi}_0=true$ we have included a truncated version of p in the definition of fit. The codicil constraining the stacks in fit is realistic because although the ψ rules can pair witnessed values which are not both locations such values are coerced by $\mathscr{L}$ into L or by $\mathscr{R}$ into V before use. Thus if $\hat{\rho}$ satisfies $\acute{\rho}[\![I]\!]\downarrow 1:V$ and $\grave{\rho}[\![I]\!]\downarrow 1:L$, for any ψ having $apt\psi\hat{\rho}=true$ $\mathscr{G}[\![I]\!]\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma}$ and $\mathscr{G}[\![\psi[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}$ place a stored value and a hitherto unused location on their respective stacks, but these accretions are then converted into members of L, converted into members of V or discarded; these three possibilities are embodied in I:=E, E:=I and I; E and their transforms under ψ. Notwithstanding this, the interim effect of an expression must also be captured by a suitable relation, and we therefore define

$$set=\lambda\hat{\pi}_0\hat{\pi}_1.(\grave{\upsilon}_0\downarrow 1:L\rightarrow\acute{\upsilon}_0\downarrow 1:L\vee\sim site(\grave{\upsilon}_0\downarrow 1)\grave{\rho}_0((hold(\grave{\upsilon}_0\downarrow 1)\grave{\sigma}_0)\S\grave{\upsilon}_0\dagger 1)\grave{\sigma}_0,\acute{\upsilon}_0\downarrow 1:V)$$
$$\wedge fit\langle\langle\acute{\rho}_0,\acute{\upsilon}_0\dagger 1,\acute{\sigma}_0\rangle,\langle\grave{\rho}_0,\grave{\upsilon}_0\dagger 1,\grave{\sigma}_0\rangle\rangle\hat{\pi}_1.$$

Now the continuations $\acute{\zeta}$ and $\grave{\zeta}$ supplied to I and \$I above must be subject to a constraint which is written as k and is given by $k=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge\{a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle\,|\,p\hat{\pi}\wedge set\hat{\pi}\hat{\pi}_0\}$.

An expression and its transform thus take state vectors $\hat{\pi}$ and $\hat{\pi}$ satisfying $fit\hat{\pi}\hat{\pi}_1=true$ for some $\hat{\pi}_1$ and put elements onto the stacks to obtain new vectors $\hat{\pi}$ and $\hat{\pi}$ having $set\hat{\pi}\hat{\pi}_0=true$ for some $\hat{\pi}_0$. In terms of continuations this means we require a relation o between $\acute{\xi}$ and $\grave{\xi}$ in 0 which in 2.4.8 will be shown to obey $o=\lambda\hat{\xi}\hat{\pi}_0\hat{\pi}_1.\bigwedge\{c\langle\acute{\xi}\acute{\zeta},\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_1\,|\,k\hat{\zeta}\hat{\pi}_0\}$.

Because function closures and label entry points can be witnessed, the arguments adduced in 2.2.1 show that w, p, c, k, o and a are self-referential, and accordingly we build them up by a limiting process similar to that used before. Given a suitable predicate w_n on $W^\circ\times W^\circ$ itself we set

$$p_{n+1}=\lambda\hat{\pi}.p_0\hat{\pi}\wedge\bigwedge\{w_n\hat{\omega}\hat{\pi}\,|\,kent0\hat{\omega}\hat{\pi}\}.$$

Should we link ω_n with a projection $q_n:W\to W$ and a predicate a_{n+1} on $A^\circ\times A^\circ$ we might induce $\mathbb{Z}q_n:Z\to Z$ and $\Theta q_n:O\to O$ to give us

$c_{n+1}=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge\{a_{n+1}\langle \mathbb{Z}q_n\zeta',\mathbb{Z}q_n\zeta''\rangle \mid p_{n+1}\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0\}$,

$k_{n+1}=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge\{a_{n+1}\langle \mathbb{Z}q_n\zeta',\mathbb{Z}q_n\zeta''\rangle \mid p_{n+1}\hat{\pi}\wedge set\hat{\pi}\hat{\pi}_0\}$ and

$o_{n+1}=\lambda\hat{\xi}\hat{\pi}_0\hat{\pi}_1.\bigwedge\{c_{n+1}\langle \Theta q_n\xi',\Theta q_n\xi''\rangle \hat{\pi}_1 \mid k_{n+1}\hat{\zeta}\hat{\pi}_0\}$.

We wish to distinguish the effects of these predicates on proper values from their effects on improper values and thus require that no $\omega:W$ have $q_0\omega=\perp$ unless it be $\perp$ itself. The analogy with 2.2.2 goes beyond this, however, for here also we cannot let $q_0\langle\zeta,\rho,\upsilon\rangle$ be $\langle\perp,\perp,\perp\rangle$. Intuitively *kent* arranges to treat all the accessible witnessed values on the same footing whereas the predicates u and s used before single out for special attention those values which occur at the top of the environment or are the immediate contents of locations. In our present case, therefore, although we want to cut continuations down to size by means of projections we must do so uniformly over all the accessible values. The continuations reached from $\langle\zeta,\rho,\upsilon\rangle$ are ζ and those found by passing down through ρ and υ, so it is by this route that the necessary cuts must be transmitted. All this suggests that we take $q_0\langle\zeta,\rho,\upsilon\rangle$ to be $\langle\perp,\mathfrak{A}q_0\rho,\mathfrak{B}q_0\upsilon\rangle$ and $q_{n+1}\langle\zeta,\rho,\upsilon\rangle$ to be $\langle\mathbb{Z}q_n{}^\circ\zeta,\mathfrak{A}q_{n+1}\rho,\mathfrak{B}q_{n+1}\upsilon\rangle$.

To formalize the construction of q_n we let z, o and w signify projections on Z, O and W respectively and define $\mathfrak{W}$ by

$\mathfrak{W}z\circ w=\lambda\omega.\omega:L\to\omega,\omega:B\to\omega,\omega:L^*\to\omega,$

$\omega:J\to(z^\circ\times\mathfrak{A}w\times\mathfrak{B}w)\omega,$

$\omega:F\to(o^\circ\times\mathfrak{A}w)\omega,$

$\omega:G\to(o^\circ\times\mathfrak{A}w\times\mathfrak{B}w)\omega,$

$\omega:J\to(o^\circ\times\mathfrak{A}w\times\mathfrak{B}w)\omega;$

$\mathfrak{B}w(\omega\mid P)$.

Omitting the natural mappings of W into V, E, D, J and P we have

$\mathfrak{U}\omega=\lambda\rho.\langle\lambda\mathrm{I}.\omega^{\circ}*(\rho[\![\mathrm{I}]\!]),\omega*(\rho[\![\mathrm{res}]\!]),\omega*(\rho[\![\mathrm{rec}]\!])\rangle$, $\mathfrak{P}\omega=\lambda\upsilon.\omega*\upsilon$,

$\mathfrak{S}\omega=\lambda\sigma.\langle\lambda\alpha.\langle(\sigma\downarrow 1)\alpha\downarrow 1,\omega*(\sigma\downarrow 1)\alpha\downarrow 2)\rangle,\omega*(\sigma\downarrow 2),\omega*(\sigma\downarrow 3)\rangle$,

$\mathfrak{p}\omega=\mathfrak{U}\omega\times\mathfrak{P}\omega\times\mathfrak{S}\omega$, $\mathfrak{Z}\omega=\mathfrak{U}\omega\to\mathfrak{P}\omega\to\mathfrak{S}\omega\to\mathfrak{A}\omega$ and $\mathfrak{D}\omega=\mathfrak{Z}\omega\to\mathfrak{Z}\omega$.

Since $\mathfrak{W}z o\omega$ is obviously a projection when z, o and ω are, taking $q_0=fix(\mathfrak{W}(\mathfrak{Z}\perp)(\mathfrak{D}\perp))$ and $q_{n+1}=fix(\mathfrak{W}(\mathfrak{Z}q_n)(\mathfrak{D}q_n))$ when $n\geq 0$ yields a family of projections. The next results are devoted to showing that they preserve as much of the information yielded by *kent* as one might reasonably expect.

2.4.2. Proposition.

The projections q_n defined above form an increasing sequence with join $\lambda\omega.\omega$. Furthermore

$q_0=\lambda\omega.\omega:\mathsf{L}\to\omega,\omega:\mathsf{B}\to\omega,\omega:\mathsf{L}^*\to\omega,$

$\omega:\mathsf{J}\to(\perp^{\circ}\times\mathfrak{U}q_0\times\mathfrak{P}q_0)\omega,$

$\omega:\mathsf{F}\to(\perp^{\circ}\times\mathfrak{U}q_0)\omega,$

$\omega:\mathsf{G}\to(\perp^{\circ}\times\mathfrak{U}q_0\times\mathfrak{S}q_0)\omega,$

$\omega:\mathsf{J}\to(\perp^{\circ}\times\mathfrak{U}q_0\times\mathfrak{P}q_0)\omega,$

$\mathfrak{P}q_0(\omega|\mathsf{P})$

and when $n\geq 0$

$q_{n+1}=\lambda\omega.\omega:\mathsf{L}\to\omega,\omega:\mathsf{B}\to\omega,\omega:\mathsf{L}^*\to\omega,$

$\omega:\mathsf{J}\to(\mathfrak{Z}q_n{}^{\circ}\times\mathfrak{U}q_{n+1}\times\mathfrak{P}q_{n+1})\omega,$

$\omega:\mathsf{F}\to(\mathfrak{D}q_n{}^{\circ}\times\mathfrak{U}q_{n+1})\omega,$

$\omega:\mathsf{G}\to(\mathfrak{D}q_n{}^{\circ}\times\mathfrak{U}q_{n+1}\times\mathfrak{S}q_{n+1})\omega,$

$\omega:\mathsf{J}\to(\mathfrak{Z}q_n{}^{\circ}\times\mathfrak{U}q_{n+1}\times\mathfrak{P}q_{n+1})\omega,$

$\mathfrak{P}q_{n+1}(\omega|\mathsf{P}).$

⊲The latter part of this proposition follows immediately from the construction. Moreover, if $z\sqsubseteq\lambda\zeta.\zeta$, $o\sqsubseteq\lambda\xi.\xi$ and $\omega\sqsubseteq\lambda\omega.\omega$ then $\mathfrak{W}z o\omega\sqsubseteq\lambda\omega.\omega$; hence we may conclude that $q_n\sqsubseteq\lambda\omega.\omega$ for all $n\geq 0$. Certainly $q_0\sqsubseteq q_1$, and if $q_n\sqsubseteq q_{n+1}$ then $\mathfrak{Z}q_n\sqsubseteq\mathfrak{Z}q_{n+1}$, $\mathfrak{D}q_n\sqsubseteq\mathfrak{D}q_{n+1}$ and $q_{n+1}=fix(\mathfrak{W}(\mathfrak{Z}q_n)(\mathfrak{D}q_n))\sqsubseteq fix(\mathfrak{W}(\mathfrak{Z}q_{n+1})(\mathfrak{D}q_{n+1}))=q_{n+2}$, so the sequence

$\{q_n \mid n \geq 0\}$ is increasing.

Because the structure of V in store semantics depends explicitly on that of U and Y it seems perverse to build up the domains in terms of projections on V alone. Yet the use of E instead is no more satisfactory since D is not a natural retraction of it. Even the denoted values do not exhaust the influences on U as there are no members of P among them. In short it is wise to regard W as fundamental and to posit that $\lambda\omega.\omega = fix(\mathfrak{W})$, where the functor $\mathfrak{W}$ is such that $\mathfrak{W}\omega = \mathfrak{W}(\mathfrak{T}\omega)(\mathfrak{S}\omega)\omega$.

Under this assumption, to show that $\lambda\omega.\omega = \bigsqcup q_n$ we have only to verify that for all $n \geq 0$ $r_n \sqsubseteq q_n$ where $r_{n+1} = \mathfrak{W}r_n$ and $r_0 = \bot$. Thus suppose that for some $n \geq 0$ $r_n \sqsubseteq q_n$; then from the definitions of 2.4.1 it follows that $\mathfrak{U}r_n \sqsubseteq \mathfrak{U}q_n$, $\mathfrak{P}r_n \sqsubseteq \mathfrak{P}q_n$, $\mathfrak{S}r_n \sqsubseteq \mathfrak{S}q_n$, $\mathfrak{T}r_n \sqsubseteq \mathfrak{T}q_n$ and $\mathfrak{C}r_n \sqsubseteq \mathfrak{C}q_n$, so $r_{n+1} = \mathfrak{W}r_n = \mathfrak{W}(\mathfrak{T}r_n)(\mathfrak{S}r_n)r_n \sqsubseteq \mathfrak{W}(\mathfrak{T}q_n)(\mathfrak{S}q_n)q_n \sqsubseteq \mathfrak{W}(\mathfrak{T}q_n)(\mathfrak{S}q_n)q_{n+1}$ and, as $\mathfrak{W}(\mathfrak{T}q_n)(\mathfrak{S}q_n)q_{n+1} = \mathfrak{W}(\mathfrak{T}q_n)(\mathfrak{S}q_n)(fix(\mathfrak{W}(\mathfrak{T}q_n)(\mathfrak{S}q_n))) = fix(\mathfrak{W}(\mathfrak{T}q_n)(\mathfrak{S}q_n)) = q_{n+1}$, we have $r_{n+1} \sqsubseteq q_{n+1}$.⊳

Henceforth for brevity we shall not distinguish between q° and q when q is a projection.

2.4.3. Proposition.

For every $n \geq 0$ and for any ν_0, ν_1, $\hat{\omega}_1$ and $\hat{\pi}$ there is some $\hat{\omega}_0$ with $seen\nu_0\nu_1\hat{\omega}_0((q_n \times q_n)\hat{\omega}_1)((\mathfrak{P}q_n \times \mathfrak{P}q_n)\hat{\pi}) = true$ if and only if there are $\hat{\omega}_2$ and $\hat{\omega}_3$ with $(q_n \times q_n)\hat{\omega}_3 = (q_n \times q_n)\hat{\omega}_1$ and $seen\nu_0\nu_1\hat{\omega}_2\hat{\omega}_3\hat{\pi} = true$. Moreover for any ν_0, $\hat{\omega}_0$ and $\hat{\pi}$ $kent\nu_0\hat{\omega}_0((\mathfrak{P}q_n \times \mathfrak{P}q_n)\hat{\pi}) = true$ if and only if $kent\nu_0\hat{\omega}_2\hat{\pi} = true$ for some $\hat{\omega}_2$ with $(q_n \times q_n)\hat{\omega}_2 = \hat{\omega}_0$.

⊲We fix attention on one particular $\hat{\pi}$ with $\hat{\pi}_0 = (\mathfrak{P}q_n \times \mathfrak{P}q_n)\hat{\pi}$ and proceed by induction on ν_1. If $\nu_1 = 0$ and $seen\nu_0\nu_1\hat{\omega}_0((q_n \times q_n)\hat{\omega}_1)\hat{\pi}_0 = true$, $\hat{\omega}_0 = (q_n \times q_n)\hat{\omega}_1$ so that we may take $\hat{\omega}_3 = \hat{\omega}_2 = \hat{\omega}_1$. If $\nu_1 = 0$ and $seen\nu_0\nu_1\hat{\omega}_2\hat{\omega}_3\hat{\pi} = true$ for some $\hat{\omega}_2$ and $\hat{\omega}_3$, we may take $\hat{\omega}_0 = (q_n \times q_n)\hat{\omega}_2$ and $\hat{\omega}_1 = \hat{\omega}_2$.

Assume that the result holds for some particular ν_1, and

suppose that for some $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0(\nu_1+1)\hat{\omega}_0((q_n\times q_n)\hat{\omega}_1)\hat{\pi}_0=true$. When $q_n\acute{\omega}_1:L$ or $q_n\grave{\omega}_1:L$, $access((q_n\times q_n)\hat{\omega}_1)\hat{\pi}_0=(q_n\times q_n)(access\hat{\omega}_1\hat{\pi})$ so $seen\nu_0\nu_1\hat{\omega}_0((q_n\times q_n)(access\hat{\omega}_1\hat{\pi}))\hat{\pi}_0=true$ and by the induction hypothesis $seen\nu_0\nu_1\hat{\omega}_2\hat{\omega}_4\hat{\pi}=true$ for some $\hat{\omega}_2$ and $\hat{\omega}_4$ with $(q_n\times q_n)\hat{\omega}_2=\hat{\omega}_0$ and $(q_n\times q_n)\hat{\omega}_4=(q_n\times q_n)(access\hat{\omega}_1\hat{\pi})$. Let $\hat{\omega}_3=\langle(\acute{\omega}_1:L\to\acute{\omega}_1,\acute{\omega}_4),(\grave{\omega}_1:L\to\grave{\omega}_1,\grave{\omega}_4)\rangle$; then $q_n\acute{\omega}_3=(\acute{\omega}_1:L\to q_n\acute{\omega}_1,q_n\acute{\omega}_4)=(\acute{\omega}_1:L\to q_n\acute{\omega}_1,q_n(access\hat{\omega}_1\hat{\pi}\downarrow 1))=q_n\acute{\omega}_1$ (by the definition of *access*) and similarly $q_n\grave{\omega}_3=q_n\grave{\omega}_1$, whilst $seen\nu_0(\nu_1+1)\hat{\omega}_2\hat{\omega}_3\hat{\pi}=true$. The other cases use the recursive nature of q_n in analogous fashions, so invariably there are $\hat{\omega}_2$ and $\hat{\omega}_3$ with $(q_n\times q_n)\hat{\omega}_3=(q_n\times q_n)\hat{\omega}_1$ and $seen\nu_0(\nu_1+1)\hat{\omega}_2\hat{\omega}_3\hat{\pi}=true$.

On the other hand, if $seen\nu_0(\nu_1+1)\hat{\omega}_2\hat{\omega}_3\hat{\pi}=true$ for some $\hat{\omega}_2$ and $\hat{\omega}_3$ with, say, $\acute{\omega}_3:L$ or $\grave{\omega}_3:L$, writing $\hat{\omega}_0=(q_n\times q_n)\hat{\omega}_2$ gives $seen\nu_0\nu_1\hat{\omega}_0((q_n\times q_n)(access\hat{\omega}_3\hat{\pi}))\hat{\pi}_0=true$ and thus $seen\nu_0\nu_1\hat{\omega}_0(access((q_n\times q_n)\hat{\omega}_3)\hat{\pi}_0)\hat{\pi}_0=true$, with the effect that $seen\nu_0(\nu_1+1)\hat{\omega}_0((q_n\times q_n)\hat{\omega}_3)\hat{\pi}_0=true$. Again the other cases are equally dull.

Thus $seen\nu_0\nu_1\hat{\omega}_0((q_n\times q_n)\hat{\omega}_1)\hat{\pi}_0=true$ if and only if $seen\nu_0\nu_1\hat{\omega}_2\hat{\omega}_3\hat{\pi}=true$ for some $\hat{\omega}_2$ and $\hat{\omega}_3$ with $(q_n\times q_n)\hat{\omega}_3=(q_n\times q_n)\hat{\omega}_1$ whatever the value of ν_1. To obtain the final part of the result note that $yclept((q_n\times q_n)\hat{\omega}_1)\hat{\pi}_0=true$ if and only if $yclept\hat{\omega}_3\hat{\pi}=true$ for some $\hat{\omega}_3$ having $(q_n\times q_n)\hat{\omega}_3=(q_n\times q_n)\hat{\omega}_1$.⊁

When $\alpha:L$ $q_0\alpha=\alpha$ so we can also deduce that for any ρ, υ and σ $site\alpha\rho\upsilon\sigma=true$ if and only if $site\alpha(\mathfrak{A}q_0\rho)(\mathfrak{B}q_0\upsilon)(\mathfrak{C}q_0\sigma)=true$; similar remarks pertain to *plot* and *spot*. Consequently $fit=fit\circ(\mathfrak{P}q_0\times\mathfrak{P}q_0)$ and $set=set\circ(\mathfrak{P}q_0\times\mathfrak{P}q_0)$.

More generally, we can show by the same technique that if $\{\hat{\pi}_m\mid m\geq 0\}$ is any sequence such that $\hat{\pi}_{m+1}\sqsupseteq\hat{\pi}_m$ for all $m\geq 0$ and if $\hat{\omega}_0$ is proper whenever $kent\nu_0\hat{\omega}_0\hat{\pi}_0=true$ then when $kent\nu_0\hat{\omega}(\bigsqcup\hat{\pi}_m)=true$ there is a sequence $\{\hat{\omega}_m\mid m\geq 0\}$ with $\hat{\omega}_{m+1}\sqsupseteq\hat{\omega}_m$ and $kent\nu_0\hat{\omega}_m\hat{\pi}_m=true$ for all $m\geq 0$ and with $\hat{\omega}=\bigsqcup\hat{\omega}_m$.

2.4.4. Discontinuous functions on states.

The proof of 2.4.2 tacitly assumes that there is a lattice W on which can be defined the sequence of projections $\{r_n | n\geq 0\}$. Though this assumption can be justified for *new* store semantics, in which Z is a space of continuous functions, its truth is not immediately obvious when *novel* store equations are considered. Here W will be constructed by a method which is valid even when some members of Z have discontinuities. For any such lattice and for any projection ω on the corresponding space W $\lambda\xi\rho\upsilon\sigma.\mathfrak{A}\omega(\xi(\mathfrak{U}\omega\rho)(\mathfrak{P}\omega\upsilon)(\mathfrak{S}\omega\sigma))$ will be written as $\mathfrak{Z}\omega$ despite the possibility that it need not be a projection.

On the hypothesis that W does indeed exist it can be seen from 2.4.3 that, though in accordance with 2.1.6 there are sequences $\{\rho_n | n\geq 0\}$, $\{\upsilon_n | n\geq 0\}$ and $\{\sigma_n | n\geq 0\}$ which increase with n and for which $\bigsqcup novel\rho_n\upsilon_n\sigma_n$ is not equal to $novel(\bigsqcup\rho_n)(\bigsqcup\upsilon_n)(\bigsqcup\sigma_n)$, for every ρ, υ and σ $\bigsqcup novel(\mathfrak{U}q_n\rho)(\mathfrak{P}q_n\upsilon)(\mathfrak{S}q_n\sigma)$ coincides with $novel\rho\upsilon\sigma$. Likewise if ζ is a member of Z which equals $\bigsqcup\mathfrak{Z}q_n\zeta$ then by the definitions in 2.1.4 $recur\zeta$ equals $\bigsqcup\mathfrak{Z}q_n(recur\zeta)$. Structural induction on the equations of appendix 2 will therefore demonstrate that every ζ having $\zeta=\bigsqcup\mathfrak{Z}q_n\zeta$ is such that for all E:Exp and Δ:Dec $\mathscr{E}[\![E]\!]\zeta=\bigsqcup\mathfrak{Z}q_n(\mathscr{E}[\![E]\!]\zeta)$ and $\mathscr{D}[\![\Delta]\!]\zeta=\bigsqcup\mathfrak{Z}q_n(\mathscr{D}[\![\Delta]\!]\zeta)$; similar remarks apply to all the other valuations. In consequence to ensure that the *novel* store equations are meaningful it is enough to show that there exists a lattice Z comprising precisely those mappings ζ for which $\zeta=\bigsqcup\mathfrak{Z}q_n\zeta$.

From arbitrary lattices Z and O can be constructed a lattice W which is constrained 'up to isomorphism' by the equality $W=\mathfrak{W}ZOW$, where the function $\mathfrak{W}$ is that introduced in 2.4.1. For the present purpose it is necessary to set up domains subject to the relation $W_n=\mathfrak{W}Z_nO_nW_n$ for each $n\geq 0$, so we introduce $\{Z_n | n\geq 0\}$,

$\{O_n \mid n\geq 0\}$ and $\{W_n \mid n\geq 0\}$, sequences of such lattices connected by maps $z_{nm}:Z_n\to Z_m$, $o_{nm}:O_n\to O_m$ and $w_{nm}:W_n\to W_m$ such that if $n\geq m\geq 0$ z_{nm}, o_{nm} and w_{nm} are projections having z_{mn}, o_{mn} and w_{mn} as the unique injections reciprocal to them. These sequences are founded on Z_0, which contains $\perp$ as its sole element; hence whenever $n\geq 0$ $z_{n0}=\lambda\zeta_n.\perp$ and $z_{0n}=\lambda\zeta_0.\perp$. Once Z_n and the set $\{z_{nm} \mid n\geq m\geq 0\}$ have been formed for a certain $n\geq 0$ O_n can be assumed to be the space of continuous functions from Z_n into itself, so that $O_n=Z_n\to Z_n$ and $o_{nm}=\lambda\xi_n.z_{nm}\circ\xi_n\circ z_{mn}$ if $n\geq m\geq 0$. The domain W_n is the smallest solution of the equation $W_n=\mathfrak{W}_n O_n W_n$; similarly if $n\geq m\geq 0$ w_{nm} is the minimal solution of $w_{nm}=\mathfrak{W}z_{nm}o_{nm}w_{nm}$. To extend the sequences Z_{n+1} is taken to comprise all those entities ζ_{n+1} which map elements of $\mathfrak{W}W_n$, $\mathfrak{P}W_n$ and $\mathfrak{S}W_n$ into elements of $\mathfrak{A}W_n$ in such a way that if $n\geq m\geq 0$ then $\zeta_{n+1}\sqsupseteq\mathfrak{Z}(w_{mn}\circ w_{nm})\zeta_{n+1}$. The mappings which connect Z_{n+1} to the other spaces, $z_{(n+1)(m+1)}$ and $z_{(m+1)(n+1)}$, can safely be defined by

$z_{(n+1)(m+1)}=\lambda\zeta_{n+1}\rho_m\upsilon_m\sigma_m.\mathfrak{A}w_{nm}(\zeta(\mathfrak{W}w_{mn}\rho_m)(\mathfrak{P}w_{mn}\upsilon_m)(\mathfrak{S}w_{mn}\sigma_m))$ and $z_{(m+1)(n+1)}=\lambda\zeta_{n+1}\rho_n\upsilon_n\sigma_n.\mathfrak{A}w_{mn}(\zeta(\mathfrak{W}w_{nm}\rho_n)(\mathfrak{P}w_{nm}\upsilon_n)(\mathfrak{S}w_{nm}\sigma_n))$ if $n\geq m\geq 0$, since all ζ_{n+1} subject to the inequality $\zeta_{n+1}\sqsupseteq\mathfrak{Z}(w_{mn}\circ w_{nm})\zeta_{n+1}$ satisfy $\zeta_{n+1}\sqsupseteq(z_{(m+1)(n+1)}\circ z_{(n+1)(m+1)})\zeta_{n+1}$. Even if ζ_{n+1} is not monotonic, when $n\geq m\geq l\geq 0$

$$\mathfrak{Z}(w_{mn}\circ w_{nm})\zeta_{n+1}\sqsupseteq\mathfrak{Z}(w_{mn}\circ w_{nm}\circ w_{ln}\circ w_{nl})\zeta_{n+1}=\mathfrak{Z}(w_{mn}\circ w_{lm}\circ w_{nl})\zeta_{n+1}\sqsupseteq\mathfrak{Z}(w_{ln}\circ w_{nl})\zeta_{n+1}$$

The inverse limit Z_∞ consists of those infinite sequences $\{\zeta_n \mid n\geq 0\}$ for which given any $n\geq 0$ ζ_n is a member of Z_n and $z_{nm}\zeta_n=\zeta_m$ if $n\geq m\geq 0$. The component in Z_n of any ζ_∞ belonging to Z_∞ will be written as $z_{\infty n}\zeta_\infty$; needless to say, $z_{\infty n}:Z_\infty\to Z_n$ is a projection and the corresponding injection, $z_{n\infty}$, is such that for any ζ_n in Z_n $z_{n\infty}\zeta_n$ is the sequence $\{z_{nm}\zeta_n \mid m\geq 0\}$. Analogous notation will be used to describe the inverse limit spaces O_∞ and W_∞ which are endowed with projections $o_{\infty n}:O_\infty\to O_n$ and $w_{\infty n}:W_\infty\to W_n$ for every $n\geq 0$. Easy

calculations [19] show that the mapping $\lambda\xi_\infty.\bigsqcup z_{n\infty}\circ o_{\infty n}\xi_\infty\circ z_{\infty n}$ is an isomorphism of 0_∞ on to $Z_\infty \to Z_\infty$ having $\lambda\xi.\bigsqcup o_{n\infty}(z_{\infty n}\circ\xi\circ z_{n\infty})$ as its inverse and that when W_n is identified with $\mathfrak{W}Z_n 0_n W_n$ for every $n\geq 0$ the mapping $\bigsqcup(\mathfrak{W}z_{n\infty} o_{n\infty} w_{n\infty})\circ w_{\infty n}$ is an isomorphism of W_∞ on to $\mathfrak{W}Z_\infty 0_\infty W_\infty$ having $\bigsqcup w_{n\infty}\circ(\mathfrak{W}z_{\infty n} o_{\infty n} w_{\infty n})$ as its inverse.

More interesting is the result that Z_∞ is isomorphic in a natural way with the set of functions which are not necessarily continuous but which take members of $\mathfrak{W}W_\infty$, $\mathfrak{P}W_\infty$ and $\mathfrak{S}W_\infty$ into members of $\mathfrak{A}W_\infty$ in such a manner that $\zeta=\bigsqcup\mathfrak{Z}(w_{n\infty}\circ w_{\infty n})\zeta$ for all ζ belonging to this set of functions. The appropriate mapping of Z_∞ into this set is $\lambda\zeta_\infty\rho_\infty\cup_\infty\sigma_\infty.\bigsqcup\mathfrak{A}w_{n\infty}(z_{\infty(n+1)}\zeta_\infty(\mathfrak{W}w_{\infty n}\rho_\infty)(\mathfrak{P}w_{\infty n}\cup_\infty)(\mathfrak{S}w_{\infty n}\sigma_\infty))$, while its inverse is $\lambda\zeta.\bigsqcup z_{(n+1)\infty}(\lambda\rho_n\cup_n\sigma_n.\mathfrak{A}w_{\infty n}(\zeta(\mathfrak{W}w_{n\infty}\rho_n)(\mathfrak{P}w_{n\infty}\cup_n)(\mathfrak{S}w_{n\infty}\sigma_n)))$. The proof that these mappings are reciprocal to one another is a computation which hinges on the knowledge that if ζ_∞ is an arbitrary member of Z_∞ and if ζ is equal to $\bigsqcup\mathfrak{Z}(w_{\infty n}\circ w_{n\infty})\zeta$ then $\lambda\rho_\infty\cup_\infty\sigma_\infty.\mathfrak{A}w_{n\infty}(z_{\infty(n+1)}\zeta_\infty(\mathfrak{W}w_{\infty n}\rho_\infty)(\mathfrak{P}w_{\infty n}\cup_\infty)(\mathfrak{S}w_{\infty n}\sigma_\infty))$ and $z_{(n+1)\infty}(\lambda\rho_n\cup_n\sigma_n.\mathfrak{A}w_{\infty n}(\zeta(\mathfrak{W}w_{n\infty}\rho_n)(\mathfrak{P}w_{n\infty}\cup_n)(\mathfrak{S}w_{n\infty}\sigma_n))$ increase in value as n increases. There is little to be gained by embarking on the manipulations of subscripts necessary to confirm this, for the definition of Z_∞ and the nature of ζ provide all the conditions which are necessary for it to hold. Thus henceforth it will be presumed that versions of Z, 0 and W suited to *novel* store semantics are yielded by the spaces Z_∞, 0_∞ and W_∞ constructed in the paragraphs above. Furthermore when these spaces are identified with their respective images under the natural isomorphisms $z_{(n+1)\infty}\circ z_{\infty(n+1)}$ can be regarded as $\mathfrak{Z}q_n$, $o_{(n+1)\infty}\circ o_{\infty(n+1)}$ can be regarded as $\mathfrak{O}q_n$ and $w_{n\infty}\circ w_{\infty n}$ can be regarded as q_n for every $n\geq 0$. Our later work will therefore revert to the initial usage of 2.4.1 by taking o and w (with or without subscripts) to be predicates, not projections.

2.4.5. Some protean predicates.

In order to ensure the existence of a sufficient supply of *novel* locations we demand that the accessible locations be finite in number and that L be infinite (except when discussing garbage collection). Having been set up earlier in the program, the stores attached to accessible members of G are necessarily smaller in area than the 'current' one but those associated with members of P may well be bigger, since if any members of P can be reached through *kent* the current store must be a reincarnation of one attached to a member of G. One fundamental property of state vectors, to be preserved throughout a computation, is thus

$$p_0 = \lambda\hat{\pi}.\, neat\hat{\rho} \wedge \#\acute{\upsilon} = \#\grave{\upsilon} \wedge \#\acute{\sigma}\downarrow 2 = \#\grave{\sigma}\downarrow 2 \wedge \#\acute{\sigma}\downarrow 3 = \#\grave{\sigma}\downarrow 3$$
$$\wedge \bigvee\{\bigwedge\{site\grave{\alpha}_m\acute{\rho}\acute{\upsilon}\acute{\sigma} \wedge site\grave{\alpha}_m\grave{\rho}\grave{\upsilon}\grave{\sigma} \mid 1 \le m \le n\} \rightarrow \bigvee\{\hat{\alpha}_m = \hat{\alpha}_l \mid 1 \le m < l \le n\}, false \mid 2 \le n\}$$
$$\wedge \bigwedge\{(\acute{\omega}_0 : L \wedge \acute{\omega}_0 = \acute{\omega}_1 \rightarrow area\acute{\omega}_0\acute{\sigma} \wedge \grave{\omega}_0 = \grave{\omega}_1, true)$$
$$\wedge (\grave{\omega}_0 : L \wedge \grave{\omega}_0 = \grave{\omega}_1 \rightarrow area\grave{\omega}_0\grave{\sigma} \wedge \acute{\omega}_0 = \acute{\omega}_1, true) \mid kent0\hat{\omega}_0\hat{\pi} \wedge kent1\hat{\omega}_1\hat{\pi} \wedge \hat{\omega}_0 : L \times L\}.$$

We want equivalent programs always to perform assignments in tandem and therefore cannot permit any location set up by one to be paired with two distinct locations by *kent*. Indeed, in p_0 above we even insist that no moiety which is a location be paired with two expressed values. In $apt\psi\hat{\rho}$ it is envisaged that if $hoten\hat{\omega}\hat{\rho} = true$ and $\acute{\omega}$ and $\grave{\omega}$ lie in different summands of W then $\acute{\omega}$:L and $\grave{\omega}$:V, so the initial predicate on witnessed values, w_0, incorporates the same restriction. As accessible members of G and P have stores associated with them which may become current we impose the same condition on them as on the main store. Accordingly w_0, which is akin to the inclusive predicate v_0 of 2.2.8, is given by the equation

$w_0 = \lambda\hat{\omega}\hat{\pi}.\hat{\omega}:B\times B\to b\hat{\omega}$,

$\hat{\omega}:L^*\times L^*\to \#\acute{\omega}=\#\grave{\omega}$,

$\hat{\omega}:J\times J\to neat\langle \acute{\omega}{\downarrow}2,\grave{\omega}{\downarrow}2\rangle \wedge \#\acute{\omega}{\downarrow}3=\#\grave{\omega}{\downarrow}3$

$\wedge\bigwedge\{\hat{\varepsilon}:L\times L\vee\hat{\varepsilon}:V\times V\vee\sim gyven\hat{\varepsilon}\langle \acute{\omega}{\downarrow}3,\grave{\omega}{\downarrow}3\rangle \mid\hat{\varepsilon}:E\times E\}$,

$\hat{\omega}:F\times F\to neat\langle \acute{\omega}{\downarrow}2,\grave{\omega}{\downarrow}2\rangle$,

$\hat{\omega}:J\times J\to neat\langle \acute{\omega}{\downarrow}2,\grave{\omega}{\downarrow}2\rangle \wedge \#\acute{\omega}{\downarrow}3=\#\grave{\omega}{\downarrow}3$

$\wedge\bigwedge\{\hat{\varepsilon}:L\times L\vee\hat{\varepsilon}:V\times V\vee\sim gyven\hat{\varepsilon}\langle \acute{\omega}{\downarrow}3,\grave{\omega}{\downarrow}3\rangle \mid\hat{\varepsilon}:E\times E\}$,

$\hat{\omega}:G\times G\to p_0\langle\langle \acute{\omega}{\downarrow}2,\langle\rangle,\acute{\omega}{\downarrow}3\rangle,\langle \grave{\omega}{\downarrow}2,\langle\rangle,\grave{\omega}{\downarrow}3\rangle\rangle$,

$\hat{\omega}:P\times P\to p_0\hat{\omega}\wedge\bigwedge\{\hat{\varepsilon}:L\times L\vee\hat{\varepsilon}:V\times V\vee\sim gyven\hat{\varepsilon}\langle \acute{\omega}{\downarrow}2,\grave{\omega}{\downarrow}2\rangle \mid\hat{\varepsilon}:E\times E\}$,

$\acute{\omega}:L\wedge\grave{\omega}:E$.

We take $b\hat{\beta}$ to be $\acute{\beta}=\grave{\beta}\wedge\acute{\beta}:B\wedge\grave{\beta}:B$ just as in 2.2.2. Notice that if $\hat{\pi}$ satisfies $\bigwedge\{w_0\hat{\omega}\hat{\pi}\mid kent0\hat{\omega}\hat{\pi}\}=true$ then $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}$ is proper for all ν_0, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ having $yclept\hat{\omega}_1\hat{\pi}=true$, with the effect that 2.1.8 can be used freely; the proof of this is merely an induction on ν_1 using facts such as that if $\hat{\omega}_2:F\times F$ and $w_0\hat{\omega}_2\hat{\pi}=true$ then $\langle \acute{\omega}_2{\downarrow}2,\grave{\omega}_2{\downarrow}2\rangle$ is proper. Furthermore $site\grave{\alpha}\grave{\rho}\grave{\upsilon}\grave{\delta}=true$ if and only if there exists an $\hat{\varepsilon}$ having $kent0\langle \acute{\varepsilon},\grave{\alpha}\rangle\hat{\pi}=true$, and similar remarks hold for *plot* and *spot*; again the proof involves an induction over values taken by *seen* of the kind we shall encounter very frequently.

Taking the definitions of p_{n+1}, c_{n+1}, k_{n+1} and o_{n+1} given in 2.4.1 we set

$w_{n+1}=\lambda\hat{\omega}\hat{\pi}.w_0\hat{\omega}\hat{\pi}$

$\wedge\langle\hat{\omega}:J\times J\to\bigwedge\{c_{n+1}\langle \acute{\omega}{\downarrow}1,\grave{\omega}{\downarrow}1\rangle\hat{\pi}_0\mid\hat{\rho}_0=\langle \acute{\omega}{\downarrow}2,\grave{\omega}{\downarrow}2\rangle\wedge\hat{\upsilon}_0=\langle \acute{\omega}{\downarrow}3,\grave{\omega}{\downarrow}3\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\}$

$\hat{\omega}:F\times F\to\bigwedge\{o_{n+1}\langle\lambda\zeta.(\acute{\omega}{\downarrow}1)(\zeta\circ revert\acute{\rho}_0),\lambda\zeta.(\grave{\omega}{\downarrow}1)(\zeta\circ revert\grave{\rho}_0)\rangle\hat{\pi}_0\hat{\pi}_1$

$\mid\hat{\rho}_1=\langle divert\acute{\rho}_0(\acute{\omega}{\downarrow}2),divert\grave{\rho}_0(\grave{\omega}{\downarrow}2)\rangle$

$\wedge\hat{\upsilon}_0=\langle\hat{\upsilon}_1\dagger1\rangle\wedge fit\hat{\pi}_1\hat{\pi}_1\}$,

$\hat{\omega}:G\times G\to\bigwedge\{o_{n+1}\langle \acute{\omega}{\downarrow}1,\grave{\omega}{\downarrow}1\rangle\hat{\pi}_0\hat{\pi}_1$

$\mid\hat{\rho}_1=\langle(\hat{\omega}{\downarrow}2)[\hat{\pi}_0/\mathrm{rec}],(\grave{\omega}{\downarrow}2)[\hat{\pi}_0/\mathrm{rec}]\rangle$

$\wedge\hat{\upsilon}_1=\langle\langle\rangle,\langle\rangle\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\}$,

$\hat{\omega}:J\times J\to\bigwedge\{c_{n+1}\langle \acute{\omega}{\downarrow}1,\grave{\omega}{\downarrow}1\rangle\hat{\pi}_0\mid\hat{\rho}_0=\langle \acute{\omega}{\downarrow}2,\grave{\omega}{\downarrow}2\rangle\wedge\hat{\upsilon}_0=\langle \acute{\omega}{\downarrow}3,\grave{\omega}{\downarrow}3\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\}$

$true\rangle$.

We now write $\omega=\lambda\hat{\omega}\hat{\pi}.\bigwedge\omega_n\hat{\omega}\hat{\pi}$, $p=\lambda\hat{\pi}.\bigwedge p_{n+1}\hat{\pi}$, $c=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge c_{n+1}\hat{\zeta}\hat{\pi}_0$, $k=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge k_{n+1}\hat{\zeta}\hat{\pi}_0$, $o=\lambda\hat{\xi}\hat{\pi}_0\hat{\pi}_1.\bigwedge o_{n+1}\hat{\xi}\hat{\pi}_0\hat{\pi}_1$ and $a=\lambda\hat{\delta}.\bigwedge a_{n+1}\hat{\delta}$; 2.4.6 will show that these relations provide what is wanted. The notion of the 'inclusive predicate' given in 2.2.3 must plainly be extended to relations taking more than one argument: we shall regard ω, say, as inclusive if for any sequences $\{\hat{\omega}_m \mid m\geq 0\}$ and $\{\hat{\pi}_m \mid m\geq 0\}$ such that $\hat{\omega}_{m+1}\supseteq\hat{\omega}_m$, $\hat{\pi}_{m+1}\supseteq\hat{\pi}_m$ and $\omega\hat{\omega}_m\hat{\pi}_m=true$ for all $m\geq 0$ we have $\omega(\bigsqcup\hat{\omega}_m)(\bigsqcup\hat{\pi}_m)=true$. Likewise o will be inclusive if $\lambda\langle\langle\hat{\xi},\hat{\pi}_0,\hat{\pi}_1\rangle,\langle\hat{\xi},\hat{\pi}_0,\hat{\pi}_1\rangle\rangle.o\hat{\xi}\hat{\pi}_0\hat{\pi}_1$ is inclusive in the earlier sense.

As before we shall assume that $a\langle\bot,\bot\rangle$ and $a\langle\top,\top\rangle$ are both *true*. Now, however, our predicates on expressions compare the outcome of a program with the outcome of all its transforms under suitable $\psi:\mathrm{Ide}\mapsto\mathrm{B}^*$. We therefore set

$E=\lambda\mathrm{E}.\bigwedge\{o\langle\mathscr{E}[\![\mathrm{E}]\!],\mathscr{E}[\![e[\![\mathrm{E}]\!]\psi]\!]\rangle\hat{\pi}\hat{\pi}\mid apt\psi\hat{\rho}\wedge rent[\![\mathrm{E}]\!]\psi\wedge fit\hat{\pi}\hat{\pi}\};$

$L=\lambda\mathrm{E}.\bigwedge\{o\langle\mathscr{L}[\![\mathrm{E}]\!],\mathscr{L}[\![e[\![\mathrm{E}]\!]\psi]\!]\rangle\hat{\pi}\hat{\pi}\mid apt\psi\hat{\rho}\wedge rent[\![\mathrm{E}]\!]\psi\wedge fit\hat{\pi}\hat{\pi}\};$

$R=\lambda\mathrm{E}.\bigwedge\{o\langle\mathscr{R}[\![\mathrm{E}]\!],\mathscr{R}[\![e[\![\mathrm{E}]\!]\psi]\!]\rangle\hat{\pi}\hat{\pi}\mid apt\psi\hat{\rho}\wedge rent[\![\mathrm{E}]\!]\psi\wedge fit\hat{\pi}\hat{\pi}\};$

$G=\lambda\mathrm{E}.\bigwedge\{o\langle\mathscr{G}[\![\mathrm{E}]\!],\mathscr{G}[\![g[\![\mathrm{E}]\!]\psi]\!]\rangle\hat{\pi}\hat{\pi}\mid apt\psi\hat{\rho}\wedge rent[\![\mathrm{E}]\!]\psi\wedge fit\hat{\pi}\hat{\pi}\}$

$\wedge(\mathscr{I}[\![\mathrm{E}]\!]\S\mathscr{X}[\![\mathrm{E}]\!]=\langle\rangle$

$\vee\bigwedge\{\bigwedge\{\omega\langle\mathscr{P}[\![\mathrm{E}]\!]\hat{\zeta}\hat{\rho}\hat{\upsilon}\S\mathscr{Q}[\![\mathrm{E}]\!]\hat{\zeta}\hat{\rho}\hat{\upsilon}\downarrow\nu,$

$swap(\mathscr{I}[\![\mathrm{E}]\!]\S\mathscr{X}[\![\mathrm{E}]\!])(\mathscr{I}[\![g[\![\mathrm{E}]\!]\psi]\!]\S\mathscr{X}[\![g[\![\mathrm{E}]\!]\psi]\!])$

$(\mathscr{P}[\![g[\![\mathrm{E}]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}\S\mathscr{Q}[\![g[\![\mathrm{E}]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon})\downarrow\nu\rangle\hat{\pi}$

$\mid 1\leq\nu\leq\#\mathscr{I}[\![\mathrm{E}]\!]\S\mathscr{X}[\![\mathrm{E}]\!]\}$

$apt\psi\hat{\rho}\wedge torn[\![\mathrm{E}]\!]\psi\wedge fit\hat{\pi}\hat{\pi}\wedge k\hat{\zeta}\hat{\pi}\}).$

Here *swap* (defined as in 1.4.6) aligns the members of $\mathscr{P}[\![g[\![\mathrm{E}]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}\S\mathscr{Q}[\![g[\![\mathrm{E}]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}$ with those of $\mathscr{P}[\![\mathrm{E}]\!]\acute{\zeta}\acute{\rho}\acute{\upsilon}\S\mathscr{Q}[\![\mathrm{E}]\!]\acute{\zeta}\acute{\rho}\acute{\upsilon}$.

The predicates on declarations are

$D=\lambda\Delta.\bigwedge\{c\langle\mathscr{D}[\![\Delta]\!]\acute{\zeta},\mathscr{D}[\![d[\![\Delta]\!]\psi]\!]\grave{\zeta}\rangle\hat{\pi}_0\mid\bigwedge\{c\hat{\zeta}\hat{\pi}_1\mid sewn[\![\Delta]\!]0\psi\hat{\pi}_0\hat{\pi}_1\}\};$

$T=\lambda\Delta.\bigwedge\{c\langle\mathscr{T}[\![\Delta]\!]\acute{\zeta},\mathscr{T}[\![t[\![\Delta]\!]\psi]\!]\grave{\zeta}\rangle\hat{\pi}_0\mid\bigwedge\{c\hat{\zeta}\hat{\pi}_1\mid sewn[\![\Delta]\!]1\psi\hat{\pi}_0\hat{\pi}_1\}\}.$

Here again the conjunctions take into account all the suitable ψ, $\hat{\zeta}$, $\hat{\pi}_0$ and $\hat{\pi}_1$.

The analogue of *knit* required by store semantics is

$$
\begin{aligned}
sewn=\lambda\Delta\nu\psi\hat{\pi}_0\hat{\pi}_1.(\lambda\hat{\pi}_0\hat{\pi}_1.\bigwedge\{I:\mathscr{S}[\![\Delta]\!]\,\S\,\mathscr{H}[\![\Delta]\!]\rightarrow & revert\acute{\rho}_0\acute{\rho}_1[\![I]\!]\dagger\nu=\acute{\rho}_0[\![I]\!]\dagger\nu\\
& \wedge revert\grave{\rho}_0\grave{\rho}_1[\![I]\!]\dagger\nu=\grave{\rho}_0[\![I]\!]\dagger\nu\\
& \wedge(I:\mathscr{S}[\![\Delta]\!]\rightarrow\acute{\rho}_1[\![I]\!]\downarrow 1:L,\acute{\rho}_1[\![I]\!]\downarrow 1:V)\\
& \wedge(\nu=0\vee I:\mathscr{H}[\![\iota[\![\Delta]\!]\psi]\!]\rightarrow true,\\
& \quad \grave{\rho}_1[\![I]\!]\downarrow 1:L\wedge\grave{\rho}_0[\![I]\!]\downarrow 1:L),\\
& revert\acute{\rho}_0\acute{\rho}_1[\![I]\!]=\acute{\rho}_0[\![I]\!]\wedge revert\grave{\rho}_0\grave{\rho}_1[\![I]\!]=\grave{\rho}_0[\![I]\!]\\
& \wedge(\#\acute{\rho}_0[\![I]\!]>0\rightarrow\acute{\rho}_1[\![I]\!]\downarrow 1=\acute{\rho}_0[\![I]\!]\downarrow 1\wedge\grave{\rho}_1[\![I]\!]\downarrow 1=\grave{\rho}_0[\![I]\!]\downarrow 1,\\
& \quad true)\,|\,I:Ide\}\\
& \wedge\acute{\rho}_1[\![res]\!]=\acute{\rho}_0[\![res]\!]\wedge\acute{\rho}_1[\![rec]\!]=\acute{\rho}_0[\![rec]\!]\\
& \wedge\grave{\rho}_1[\![res]\!]=\grave{\rho}_0[\![res]\!]\wedge\grave{\rho}_1[\![rec]\!]=\grave{\rho}_0[\![rec]\!]\\
& \wedge apt\psi\hat{\rho}_0\wedge\hat{0}_1=\hat{0}_0\wedge fit\hat{\pi}_1\hat{\pi}_1\wedge fit\hat{\pi}_0\hat{\pi}_0\\
& \wedge(\lambda\psi'.apt\psi'\hat{\rho}_1\wedge torn[\![\Delta]\!]\psi')\\
& \quad(\nu=0\rightarrow\psi[false^*/\mathscr{S}[\![\Delta]\!]][opts(\mathscr{H}[\![\Delta]\!])\psi/\mathscr{H}[\![\Delta]\!]],\psi))\\
((\mathfrak{P}q_0\times\mathfrak{P}q_0)\hat{\pi}_0)&((\mathfrak{P}q_0\times\mathfrak{P}q_0)\hat{\pi}_1).
\end{aligned}
$$

As promised earlier, we shall now confirm that our predicates have the desired self-referential nature.

2.4.6. Lemma.

Suppose that $a_1(\perp,\perp)=true$, $a_1\hat{\delta}$ is always proper, and if $w_{n+1}\hat{\omega}\hat{\pi}\supset w_n\hat{\omega}\hat{\pi}$ and $w_n\hat{\omega}\hat{\pi}\wedge kent0\hat{\omega}\hat{\pi}\wedge p_1\hat{\pi}\supset w_{n+1}((q_n\times q_n)\hat{\omega})\hat{\pi}$ for all $\hat{\omega}:W^\circ\times W^\circ$ and $\hat{\pi}:P^\circ\times P^\circ$ then $a_{n+2}\hat{\delta}\supset a_{n+1}\hat{\delta}$ and $a_{n+1}\hat{\delta}\supset a_{n+2}((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})$ for all $\hat{\delta}:A^\circ\times A^\circ$. For every member of the relevant domains and for every $n\geq 0$,

(i) $w_{n+1}\hat{\omega}\hat{\pi}\supset w_n\hat{\omega}\hat{\pi}$ and $w_n\hat{\omega}\hat{\pi}\wedge kent0\hat{\omega}\hat{\pi}\wedge p_1\hat{\pi}\supset w_{n+1}((q_n\times q_n)\hat{\omega})\hat{\pi}$;

(ii) $p_{n+2}\hat{\pi}\supset p_{n+1}\hat{\pi}$ and $p_{n+1}\hat{\pi}\supset p_{n+2}((\mathfrak{P}q_n\times\mathfrak{P}q_n)\hat{\pi})$;

(iii) $c_{n+2}\hat{\zeta}\hat{\pi}\supset c_{n+1}\hat{\zeta}\hat{\pi}$ and $c_{n+1}\hat{\zeta}\hat{\pi}\supset c_{n+2}((\mathfrak{Z}q_n\times\mathfrak{Z}q_n)\hat{\zeta})\hat{\pi}$;

(iv) $k_{n+2}\hat{\zeta}\hat{\pi}\supset k_{n+1}\hat{\zeta}\hat{\pi}$ and $k_{n+1}\hat{\zeta}\hat{\pi}\supset k_{n+2}((\mathfrak{Z}q_n\times\mathfrak{Z}q_n)\hat{\zeta})\hat{\pi}$;

(v) $o_{n+2}\hat{\xi}\hat{\pi}_0\hat{\pi}_1\supset o_{n+2}\hat{\xi}\hat{\pi}_0\hat{\pi}_1$ and $o_{n+1}\hat{\xi}\hat{\pi}_0\hat{\pi}_1\supset o_{n+2}((\Phi q_n\times\Phi q_n)\hat{\xi})\hat{\pi}_0\hat{\pi}_1$.

⊲Observe first that by 2.4.3 if $\hat{\omega}_0$ and $\hat{\omega}_1$ satisfy $kent0\hat{\omega}_0((\mathbb{P}q_n\times\mathbb{P}q_n)\hat{\pi})\wedge kent1\hat{\omega}_1((\mathbb{P}q_n\times\mathbb{P}q_n)\hat{\pi})=true$ there are $\hat{\omega}_2$ and $\hat{\omega}_3$ having $kent0\hat{\omega}_2\hat{\pi}\wedge kent1\hat{\omega}_3\hat{\pi}=true$, $\hat{\omega}_0=(q_n\times q_n)\hat{\omega}_2$ and $\hat{\omega}_1=(q_n\times q_n)\hat{\omega}_3$. Furthermore when $\acute{\omega}_0$:L and $\acute{\omega}_0=\acute{\omega}_1$ necessarily $\acute{\omega}_2$:L and $\acute{\omega}_2=\acute{\omega}_3$, so should $p_0\hat{\pi}$ be *true* $area\acute{\omega}_0(\mathbb{P}q_n\acute{\sigma})$ will be *true* while $\grave{\omega}_0$ and $\grave{\omega}_1$ will coincide. As similar assertions hold when $\grave{\omega}_0$:L and $\grave{\omega}_0=\grave{\omega}_1$ we can infer that for all $\hat{\pi}$ and $n\geq 0$ if $p_0\hat{\pi}=true$ then $p_0((\mathbb{P}q_n\times\mathbb{P}q_n)\hat{\pi})=true$.

⊲Suppose that $\hat{\omega}$ and $\hat{\pi}$ are any entities satisfying $w_0\hat{\omega}\hat{\pi}\wedge kent0\hat{\omega}\hat{\pi}\wedge p_1\hat{\pi}=true$. Unless $\hat{\omega}$: J×J, $\hat{\omega}$: F×F, $\hat{\omega}$: G×G or $\hat{\omega}$: J×J the definitions in 2.4.5 make it plain that $w_1((q_0\times q_0)\hat{\omega})\hat{\pi}=true$ Since $a_1\langle\perp,\perp\rangle=true$, we also know that $c_1\langle\perp,\perp\rangle\hat{\pi}_0\vee k_1\langle\perp,\perp\rangle\hat{\pi}_0\wedge o_1\langle\perp,\perp\rangle\hat{\pi}_0\hat{\pi}_1=true$ for all $\hat{\pi}_0$ and $\hat{\pi}_1$ having $fit\hat{\pi}_0\hat{\pi}_0\wedge fit\hat{\pi}_1\hat{\pi}_1=true$. Thus to show that $w_1((q_0\times q_0)\hat{\omega})\hat{\pi}=true$ we have only to establish the existence of $\hat{\pi}_0$ and $\hat{\pi}_1$ for which

$fit\hat{\pi}_0\hat{\pi}_0\wedge fit\hat{\pi}_1\hat{\pi}_1=true$,

$\hat{\pi}_0=(\hat{\omega}$:F×F∨$\hat{\omega}$: G×G→$\hat{\pi}_0,\langle\acute{\omega}\dagger 1\S\langle\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle)$,

$\hat{\pi}_1=(\hat{\omega}$:F×F→$\langle\langle divert\acute{\rho}_0(\acute{\omega}\downarrow 2),\langle\acute{\varepsilon}\rangle\S\acute{\upsilon}_0,\acute{\sigma}_0\rangle,\langle divert\grave{\rho}_0(\grave{\omega}\downarrow 2),\langle\grave{\varepsilon}\rangle\S\grave{\upsilon}_0,\grave{\sigma}_0\rangle\rangle$,

$\hat{\omega}$:G×G→$\langle\langle(\acute{\omega}\downarrow 2)[\hat{\pi}_0/\mathrm{rec}],\langle\rangle,\acute{\omega}\downarrow 3\rangle,\langle(\grave{\omega}\downarrow 2)[\hat{\pi}_0/\mathrm{rec}],\langle\rangle,\grave{\omega}\downarrow 3\rangle\rangle$,

$\hat{\pi}_0\rangle$

and $\hat{\varepsilon}=\langle dummy,dummy\rangle$; this we shall now embark upon.

⊲By 2.1.8 any ν_0, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ such that $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}\wedge kent0\hat{\omega}_1\hat{\pi}=true$ satisfy $kent0\hat{\omega}_0\hat{\pi}=true$; in particular, for any given $\hat{\omega}_1$ and all $\hat{\omega}_0$ $kent0\hat{\omega}_0\hat{\pi}_2\supset kent0\hat{\omega}_0\hat{\pi}$ where $\hat{\pi}_2=(\hat{\omega}_1$:G×G→$\langle\langle\acute{\omega}_1\downarrow 2,\langle\rangle,\acute{\omega}_1\downarrow 3\rangle,\langle\acute{\omega}_1\downarrow 2,\langle\rangle,\grave{\omega}_1\downarrow 3\rangle\rangle,\hat{\omega}_1$:P×P→$\hat{\omega}_1,\hat{\pi})$. As $p_1\hat{\pi}=true$ and $kent0\hat{\omega}_1\hat{\pi}=true$ we have $p_0\hat{\pi}_2=true$ and $\bigwedge\{w_0\hat{\omega}_0\hat{\pi}_2\mid kent0\hat{\omega}_0\hat{\pi}_2\}=true$, which between them suffice to show that $p_1\hat{\pi}_2=true$.

We can prove by induction over the values taken by *seen* that either $kent1\hat{\omega}\hat{\pi}=true$ or there is some $\hat{\omega}_1$ having $kent1\hat{\omega}\hat{\pi}_2\wedge kent0\hat{\omega}_1\hat{\pi}=true$ where $\hat{\pi}_2$ is created from $\hat{\omega}_1$ as above.

Thus we can define

$$\hat{\pi}_0=(\hat{\omega}:J\times J\to\langle\acute{\omega}\dagger 1\S\langle\acute{\sigma}_2\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_2\rangle\rangle,$$
$$\hat{\omega}:G\times G\to\langle\langle\acute{\omega}\downarrow 2,\langle\rangle,\acute{\omega}\downarrow 3\rangle,\langle\grave{\omega}\downarrow 2,\langle\rangle,\grave{\omega}\downarrow 3\rangle\rangle,$$
$$\hat{\omega}:J\times J\to\langle\acute{\omega}\dagger 1\S\langle\acute{\sigma}_2\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_2\rangle\rangle,$$
$$\hat{\pi}_2)$$

and can set up $\hat{\pi}_1$ using it in the manner suggested above.

Because $kent0\hat{\omega}\hat{\pi}=true$ our earlier remarks demonstrate that $p_1\hat{\pi}_0=true$ when $\hat{\omega}:G\times G$. In the remaining cases an induction (to be alluded to in 2.6.4) confirms that $kent\nu\hat{\omega}_2\hat{\pi}_0\supset kent\nu\hat{\omega}_2\hat{\pi}_1$ for all $\hat{\omega}_2$ and $\nu<2$, and the fact that $p_1\hat{\pi}_2=true$ ensures that $p_1\hat{\pi}_0=true$. A similar argument establishes that $p_1\hat{\pi}_1=true$ so the knowledge that $w_0\hat{\omega}\hat{\pi}=true$ allows us to assert that $fit\hat{\pi}_0\hat{\pi}_0\wedge fit\hat{\pi}_1\hat{\pi}_1=true$.➢

As a consequence of this digression we see that $w_0\hat{\omega}\hat{\pi}\wedge kent0\hat{\omega}\hat{\pi}\wedge p_1\hat{\pi}\supset w_1((q_n\times q_n)\hat{\omega})\hat{\pi}$ for all $\hat{\omega}$ and $\hat{\pi}$. From 2.4.5 it is obvious that $w_1\hat{\omega}\hat{\pi}\supset w_0\hat{\omega}\hat{\pi}$ for all $\hat{\omega}$ and $\hat{\pi}$, so we have constructed a suitable foundation for an inductive proof of the result.➢

Assume that (i) is valid when $n=m$ for some $m\geq 0$ and that (ii) to (v) are valid when $m-1\geq n\geq 0$. We shall show that (i) holds when $n=m+1$ and that (ii) to (v) hold when $n=m$.

Firstly, for any $\hat{\pi}$

$$p_{m+2}\hat{\pi}=p_0\hat{\pi}\wedge\bigwedge\{w_{m+1}\hat{\omega}\hat{\pi}\mid kent0\hat{\omega}\hat{\pi}\}\supset p_0\hat{\pi}\wedge\bigwedge\{w_m\hat{\omega}\hat{\pi}\mid kent0\hat{\omega}\hat{\pi}\}=p_{m+1}\hat{\pi},$$

while from 2.4.3 because $m+1\geq 1$ we have

$$\begin{aligned}p_{m+1}\hat{\pi}&=p_0\hat{\pi}\wedge\bigwedge\{w_m\hat{\omega}\hat{\pi}\mid kent0\hat{\omega}\hat{\pi}\}\\&\supset p_0\hat{\pi}\wedge\bigwedge\{w_{m+1}((q_m\times q_m)\hat{\omega})\hat{\pi}\mid kent0\hat{\omega}\hat{\pi}\}\\&=p_0\hat{\pi}\wedge\bigwedge\{w_{m+1}\hat{\omega}\hat{\pi}\mid kent0\hat{\omega}((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi})\}\\&\supset p_0((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi})\wedge\bigwedge\{w_{m+1}\hat{\omega}((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi})\mid kent0\hat{\omega}((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi})\}\\&=p_{m+2}((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi}).\end{aligned}$$

Consequently (ii) is valid when $n=m$, and in particular $p_{m+2}\hat{\pi}\supset p_{m+1}\hat{\pi}\supset p_{m+2}((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi})$ for all $\hat{\pi}$.

Suppose that $c_{m+2}\hat{\zeta}\hat{\pi}_0=true$ for some $\hat{\zeta}$ and $\hat{\pi}_0$. If

$p_{m+1}\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0=true$, $p_{m+2}((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi})\wedge fit((\mathbb{P}q_m\times\mathbb{P}q_m)\hat{\pi})\hat{\pi}_0=true$ and writing $\hat{\theta}=\langle\mathbb{Z}q_m\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\mathbb{Z}q_m\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle$ gives

$\hat{\theta}=\langle\mathbb{Z}q_{m+1}\acute{\zeta}(\mathbb{U}q_m\acute{\rho})(\mathbb{P}q_m\acute{\upsilon})(\mathbb{S}q_m\acute{\sigma}),\mathbb{Z}q_{m+1}\grave{\zeta}(\mathbb{U}q_m\grave{\rho})(\mathbb{P}q_m\grave{\upsilon})(\mathbb{S}q_m\grave{\sigma})\rangle$

and $a_{m+2}\hat{\theta}=true$. By our original premise, $a_{m+2}\hat{\theta}\supset a_{m+1}\hat{\theta}$ so $c_{m+1}\hat{\zeta}\hat{\pi}_0=true$.

Conversely if $c_{m+1}\hat{\zeta}\hat{\pi}_0=true$ and $p_{m+2}\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0=true$, then $p_{m+1}\hat{\pi}=true$ so, writing $\hat{\theta}=\langle\mathbb{Z}q_m\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\mathbb{Z}q_m\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle$, $a_{m+1}\hat{\theta}=true$ and $a_{m+2}\hat{\theta}=true$ (again by the original premise). Hence $c_{m+2}((\mathbb{Z}q_m\times\mathbb{Z}q_m)\hat{\zeta})\hat{\pi}_0=true$ and (iii) holds when $n=m$; the proof of (iv) is almost identical. Note that in addition $c_{m+2}\hat{\zeta}\hat{\pi}_0\supset c_{m+1}\hat{\zeta}\hat{\pi}_0\supset c_{m+1}((\mathbb{Z}q_m\times\mathbb{Z}q_m)\hat{\zeta})\hat{\pi}_0$.

Furthermore, for any $\hat{\xi}$, $\hat{\zeta}$, $\hat{\pi}_0$ and $\hat{\pi}_1$

$$\begin{aligned}o_{m+2}\hat{\xi}\hat{\pi}_0\hat{\pi}_1\wedge k_{m+1}\hat{\zeta}\hat{\pi}_0&\supset o_{m+2}\hat{\xi}\hat{\pi}_0\hat{\pi}_1\wedge k_{m+2}((\mathbb{Z}q_m\times\mathbb{Z}q_m)\hat{\zeta})\hat{\pi}_0\\&\supset c_{m+2}\langle\mathbb{Q}q_{m+1}\acute{\xi}(\mathbb{Z}q_m\acute{\zeta}),\mathbb{Q}q_{m+1}\grave{\xi}(\mathbb{Z}q_m\grave{\zeta})\rangle\,\hat{\pi}_1\\&\supset c_{m+1}\langle\mathbb{Z}q_m(\acute{\xi}(\mathbb{Z}q_m\acute{\zeta})),\mathbb{Z}q_m(\grave{\xi}(\mathbb{Z}q_m\grave{\zeta}))\rangle\,\hat{\pi}_1\\&\supset c_{m+1}\langle\mathbb{Q}q_m\acute{\xi}\acute{\zeta},\mathbb{Q}q_m\grave{\xi}\grave{\zeta}\rangle\,\hat{\pi},\end{aligned}$$

and

$$\begin{aligned}o_{m+1}\hat{\xi}\hat{\pi}_0\hat{\pi}_1\wedge k_{m+2}\hat{\zeta}\hat{\pi}_0&\supset o_{m+1}\hat{\xi}\hat{\pi}_0\hat{\pi}_1\wedge k_{m+1}((\mathbb{Z}q_m\times\mathbb{Z}q_m)\hat{\zeta})\hat{\pi}_0\\&\supset c_{m+1}\langle\mathbb{Q}q_m\acute{\xi}\acute{\zeta},\mathbb{Q}q_m\grave{\xi}\grave{\zeta}\rangle\,\hat{\pi}_1\\&\supset c_{m+2}\langle\mathbb{Q}q_m\acute{\xi}\acute{\zeta},\mathbb{Q}q_m\grave{\xi}\grave{\zeta}\rangle\,\hat{\pi}_1.\end{aligned}$$

From the definition in 2.4.1 it is plain that (v) is valid when $n=m$.

Suppose that $w_{m+2}\hat{\omega}\hat{\pi}=true$ for some $\hat{\omega}$ and $\hat{\pi}$; if $\hat{\omega}:J\times J$, say, by (iii) any $\hat{\sigma}_0$ such that $c_{m+2}\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\langle\acute{\omega}\dagger 1\S\langle\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle=true$ has $c_{m+1}\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\langle\acute{\omega}\dagger 1\S\langle\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle=true$ also. Similar reasoning applies whatever summand of W $\acute{\omega}$ and $\grave{\omega}$ lie in, so ineluctably $w_{m+1}\hat{\omega}\hat{\pi}=true$.

On the other hand, suppose that for some $\hat{\omega}$ and $\hat{\pi}$ $w_{m+1}\hat{\omega}\hat{\pi}=true$. If $\hat{\omega}:J\times J$, writing $\hat{\zeta}=\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle$ we know that any $\hat{\pi}_0$ having $\hat{\pi}_0=\langle\acute{\omega}\dagger 1\S\langle\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle$ and $c_{m+1}\hat{\zeta}\hat{\pi}_0=true$ satisfies also

$c_{m+2}((\mathbb{Z}q_m \times \mathbb{Z}q_m)\hat{\zeta})\hat{\pi}_0 = true$, from the truth of (iii) when $n=m$; because $p_{m+2}\hat{\pi}_1 \wedge fit\hat{\pi}_1((\mathbb{P}q_{m+1} \times \mathbb{P}q_{m+1})\hat{\pi}_0) = true$ for every pair $\hat{\pi}_1$ having $p_{m+2}\hat{\pi}_1 \wedge fit\hat{\pi}_1\hat{\pi}_0 = true$, $c_{m+2}((\mathbb{Z}q_m \times \mathbb{Z}q_m)\hat{\zeta})((\mathbb{P}q_{m+1} \times \mathbb{P}q_{m+1})\hat{\pi}_0) = true$ and $w_{m+2}((q_{m+1} \times q_{m+1})\hat{\omega})((\mathbb{P}q_{m+1} \times \mathbb{P}q_{m+1})\hat{\pi}) = true$. Moreover, for any $\hat{\xi}$, $\hat{\pi}_0$ and $\hat{\pi}_1$ $o_{m+2}\hat{\xi}\hat{\pi}_0\hat{\pi}_1 \supset o_{m+2}\hat{\xi}\hat{\pi}_0\hat{\pi}_2$ whenever $(\mathbb{P}q_{m+1} \times \mathbb{P}q_{m+1})\hat{\pi}_1 \sqsubseteq \hat{\pi}_2 \sqsubseteq \hat{\pi}_1$ (since for any $\hat{\zeta}$ $c_{m+2}\hat{\zeta}\hat{\pi}_1 \supset c_{m+2}\hat{\zeta}\hat{\pi}_2$), and hence similar arguments be adduced to verify that when $\hat{\omega}:F\times F$ $w_{m+2}((q_{m+1} \times q_{m+1})\hat{\omega})((\mathbb{P}q_{m+1} \times \mathbb{P}q_{m+1})\hat{\pi}) = true$. Finally, for every $\hat{\pi}_2$ $p_0\hat{\pi}_2 \supset p_0((\mathbb{P}q_{m+1} \times \mathbb{P}q_{m+1})\hat{\pi}_2)$ so (i) holds when $n=m+1$ even without the assumption that $kent0\hat{\omega}\hat{\pi} \wedge p_1\hat{\pi} = true$.

We may therefore conclude that (i) to (v) hold for all n.⊁

It is this lemma and its underlying motivation which are responsible for our inability to allow $w_n\hat{\omega}\hat{\pi}$ to be *true* if $\acute{\omega}:E$ and $\grave{\omega}:G$. In no sense do $\perp:F$ and $\perp:G$, say, provide equivalent programs so the stipulation that $w_n\langle\perp,\perp\rangle\hat{\pi} = true$ is unreasonable. Moreover q_n acts quite differently on E and on G so $\acute{\omega}:E$ and $\grave{\omega}:G$ would yield information at distinct rates on progressing from w_n to w_{n+1}. Loosely speaking, $w_n\hat{\varepsilon}\hat{\pi}$ is on a par with $w_{n+1}\hat{\delta}\hat{\pi}$ when $\hat{\varepsilon}:E\times E$ and $\hat{\delta}:G\times G$.

2.4.7. Lemma.

Suppose that for any $n\geq 0$ if w_n is inclusive then a_{n+1} is inclusive; then w, p, c, k, o and a, defined as above, are inclusive.

⊀Just as in 2.2.4 we can show by induction that for every $n\geq 0$ w_n, p_{n+1}, c_{n+1}, k_{n+1}, o_{n+1} and a_{n+1} are inclusive.

Suppose, for instance, that w_n is inclusive for some $n\geq 0$, so that a_{n+1} is inclusive. Take any sequence $\{\hat{\pi}_m\}$ with $\hat{\pi}_{m+1} \sqsupseteq \hat{\pi}_m$ for all $m\geq 0$; then if $\bigwedge p_{n+1}\hat{\pi}_m = true$, $\bigwedge p_0\hat{\pi}_m = true$ so $p_0(\bigsqcup\hat{\pi}_m) = true$. Because $p_{n+1}\hat{\pi}_0 = true$, $\hat{\omega}_0$ is proper whenever $kent0\hat{\omega}_0\hat{\pi}_0 = true$ and (as indicated after 2.4.3) when $kent0\hat{\omega}(\bigsqcup\hat{\pi}_m) = true$ there are $\hat{\omega}_m$ with

$\hat{\omega}_{m+1} \sqsupseteq \hat{\omega}_m$ and $kent0\hat{\omega}_m\hat{\pi}_m = true$ for each $m \geq 0$ and with $\hat{\omega} = \bigsqcup \hat{\omega}_m$. Hence $\bigwedge p_{n+1}\hat{\pi}_m \supset \bigwedge w_n\hat{\omega}_m\hat{\pi}_m \supset w_n(\bigsqcup\hat{\omega}_m)(\bigsqcup\hat{\pi}_m)$ and $p_{n+1}(\bigsqcup\hat{\pi}_m) = true$.

When p_{n+1} and a_{n+1} are inclusive it is possible to exhibit the same property for c_{n+1} and k_{n+1}; only the latter will be discussed here, the proof for the former being almost identical. Accordingly, take $\{\hat{\zeta}_m\}$ and $\{\hat{\pi}_m\}$ having $\hat{\zeta}_{m+1} \sqsupseteq \hat{\zeta}_m$, $\hat{\pi}_{m+1} \sqsupseteq \hat{\pi}_m$ and $k_{n+1}\hat{\zeta}_m\hat{\pi}_m = true$ for all $m \geq 0$. When $set\hat{\pi}(\bigsqcup\hat{\pi}_m) = true$, inevitably $set\hat{\pi}\hat{\pi}_0 = set\hat{\pi}\hat{\pi}_1 = set\hat{\pi}\hat{\pi}_2 = \ldots = true$, as by fixed point induction on $\mathfrak{W}_\perp$ $q_0\omega' = q_0\omega''$ if $\perp \sqsubset \omega' \sqsubseteq \omega'' \sqsubset \top$. Should $p\hat{\pi}$ be $true$ in addition, for every $m \geq 0$

$a_{n+1}\langle \mathfrak{E} q_n\hat{\zeta}_m\rho\upsilon\sigma, \mathfrak{E} q_n\hat{\zeta}_m\rho\upsilon\sigma\rangle = true$ and, a_{n+1} being inclusive, $a_{n+1}\langle \mathfrak{E} q_n(\bigsqcup\hat{\zeta}_m)\rho\upsilon\sigma, \mathfrak{E} q_n(\bigsqcup\hat{\zeta}_m)\rho\upsilon\sigma\rangle = true$; hence $k_{n+1}(\bigsqcup\hat{\zeta}_m)(\bigsqcup\hat{\pi}_m) = true$.

The other cases are similar so w_n, p_{n+1}, c_{n+1}, k_{n+1}, o_{n+1} and a_{n+1} are indeed inclusive for every $n \geq 0$; the conjunction of a set of inclusive predicates being inclusive, we may conclude that w, p, c, k, o and a are inclusive.➢

2.4.8. Proposition.

Suppose that $a_1\langle \perp, \perp\rangle = true$ and that for any $n \geq 0$ if w_n is inclusive and every $\hat{\omega}$ and $\hat{\pi}$ having $kent0\hat{\omega}\hat{\pi} \wedge p_1\hat{\pi} = true$ satisfy $w_n((q_n \times q_n)\hat{\omega})\hat{\pi} \supset w((q_n \times q_n)\hat{\omega})\hat{\pi}$, $w_{n+1}\hat{\omega}\hat{\pi} \supset w_n\hat{\omega}\hat{\pi}$ and $w_n\hat{\omega}\hat{\pi} \supset w_{n+1}((q_n \times q_n)\hat{\omega})\hat{\pi}$ as well then a_{n+1} is inclusive and for every $\hat{\sigma}$ $a_1\hat{\sigma}$ is proper, $a_{n+1}((\mathfrak{A} q_n \times \mathfrak{A} q_n)\hat{\sigma}) \supset a((\mathfrak{A} q_n \times \mathfrak{A} q_n)\hat{\sigma})$, $a_{n+2}\hat{\sigma} \supset a_{n+1}\hat{\sigma}$ and $a_{n+1}\hat{\sigma} \supset a_{n+2}((\mathfrak{A} q_n \times \mathfrak{A} q_n)\hat{\sigma})$. Should w, p, c, k and o be defined as in 2.4.5 they will be subject to

(i) $\omega=\lambda\hat{\omega}\hat{\pi}.\omega_0\hat{\omega}\hat{\pi}$

$\wedge(\hat{\omega}:J\times J\to\bigwedge\{c\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\hat{\pi}_0\,|\,\hat{\rho}_0=\langle\acute{\omega}\downarrow 2,\grave{\omega}\downarrow 2\rangle\wedge\hat{o}_0=\langle\acute{\omega}\downarrow 3,\grave{\omega}\downarrow 3\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\}$

$\hat{\omega}:F\times F\to\bigwedge\{o\langle\lambda\zeta.(\acute{\omega}\downarrow 1)(\zeta\circ revert\acute{\rho}_0),\lambda\zeta.(\grave{\omega}\downarrow 1)(\zeta\circ revert\grave{\rho}_0)\rangle\hat{\pi}_0\hat{\pi}$

$|\,\hat{\rho}_1=\langle divert\acute{\rho}_0(\acute{\omega}\downarrow 2),divert\grave{\rho}_0(\grave{\omega}\downarrow 2)\rangle$

$\wedge\hat{o}_0=\langle\acute{o}_1\dagger 1,\grave{o}_1\dagger 1\rangle\wedge fit\hat{\pi}_1\hat{\pi}_1\},$

$\hat{\omega}:G\times G\to\bigwedge\{o\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\hat{\pi}_0\hat{\pi}_1$

$|\,\hat{\rho}_1=\langle(\acute{\omega}\downarrow 2)[\hat{\pi}_0/\mathrm{rec}],(\grave{\omega}\downarrow 2)[\hat{\pi}_0/\mathrm{rec}]\rangle$

$\wedge\hat{o}_1=\langle(),()\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\},$

$\hat{\omega}:J\times J\to\bigwedge\{k\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\hat{\pi}_0\,|\,\hat{\rho}_0=\langle\acute{\omega}\downarrow 2,\grave{\omega}\downarrow 2\rangle\wedge\hat{o}_0=\langle\acute{\omega}\downarrow 3,\grave{\omega}\downarrow 3\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\}$

$true);$

(ii) $p=\lambda\hat{\pi}.p_0\hat{\pi}\wedge\bigwedge\{\omega\hat{\omega}\hat{\pi}\,|\,kent\,0\hat{\omega}\hat{\pi}\};$

(iii) $c=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge\{a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle\,|\,p\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0\};$

(iv) $k=\lambda\hat{\xi}\hat{\pi}_0.\bigwedge\{a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle\,|\,p\hat{\pi}\wedge set\hat{\pi}\hat{\pi}_0\};$

(v) $o=\lambda\hat{\xi}\hat{\pi}_0\hat{\pi}_1.\bigwedge\{c\langle\acute{\xi}\acute{\zeta},\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_1\,|\,k\hat{\zeta}\hat{\pi}_0\}.$

◁Conjunctions distribute over conditional clauses so (i) and (ii) follow from the definitions. Moreover by an induction on n involving 2.4.6, invariably $\omega_{n+1}=\omega\circ(q_{n+1}\times q_{n+1})$, $p_{n+1}=p\circ(\mathfrak{P}q_n\times\mathfrak{P}q_n)$, $c_{n+1}\circ(\mathfrak{Z}q_n\times\mathfrak{Z}q_n)=c\circ(\mathfrak{Z}q_n\times\mathfrak{Z}q_n)$, $k_{n+1}\circ(\mathfrak{Z}q_n\times\mathfrak{Z}q_n)=k\circ(\mathfrak{Z}q_n\times\mathfrak{Z}q_n)$, $o_{n+1}\circ(\mathfrak{O}q_n\times\mathfrak{O}q_n)=o\circ(\mathfrak{O}q_n\times\mathfrak{O}q_n)$ and $a_{n+1}\circ(\mathfrak{A}q_n\times\mathfrak{A}q_n)=a\circ(\mathfrak{A}q_n\times\mathfrak{A}q_n)$. We shall establish only (iii) and (v) from these equalities, leaving (iv) to the imagination.

Suppose that $c\hat{\zeta}\hat{\pi}_0=true$ for some $\hat{\zeta}$ and $\hat{\pi}_0$, and take any $\hat{\pi}$ with $p\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0=true$. Then for every $n\geq 0$, $c_{n+1}\hat{\zeta}\hat{\pi}_0\wedge p_{n+1}\hat{\pi}=true$ so $a_{n+1}\langle\mathfrak{Z}q_n\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\mathfrak{Z}q_n\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle=true$ and $a\langle\mathfrak{Z}q_n\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\mathfrak{Z}q_n\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle=true$. As a is inclusive $a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle=a\langle\bigsqcup(\mathfrak{Z}q_n\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma}),\bigsqcup(\mathfrak{Z}q_n\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma})\rangle=true$.

Conversely, if $\bigwedge\{a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle\,|\,p\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0\}=true$ for some $\hat{\zeta}$ and $\hat{\pi}_0$ let $\hat{\pi}$ be such that for some $n\geq 0$ $p_{n+1}\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0=true$. By 2.4.3 $p_{m+1}((\mathfrak{P}q_n\times\mathfrak{P}q_n)\hat{\pi})=true$ when $m>n$ so $p_{n+1}((\mathfrak{P}q_n\times\mathfrak{P}q_n)\hat{\pi})=true$ and $p((\mathfrak{P}q_n\times\mathfrak{P}q_n)\hat{\pi})=true$. Writing for convenience $\partial=\langle\acute{\zeta}(\mathfrak{U}q_n\acute{\rho})(\mathfrak{P}q_n\acute{\upsilon})(\mathfrak{S}q_n\acute{\sigma}),\grave{\zeta}(\mathfrak{U}q_n\grave{\rho})(\mathfrak{P}q_n\grave{\upsilon})(\mathfrak{S}q_n\grave{\sigma})\rangle$ gives $a\partial=true$, from

which we can deduce in turn that $a_{n+1}\hat{\delta}=true$, $a_{n+2}((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})=true$ and $a_{n+1}((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})=true$. Hence $a_{n+1}\langle\mathfrak{E}q_n\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\mathfrak{E}q_n\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle=true$, $c_{n+1}\hat{\zeta}\hat{\pi}_0=true$ and, n being arbitrary, $c\hat{\zeta}\hat{\pi}_0=true$.

Suppose that $o\hat{\xi}\hat{\pi}_0\hat{\pi}_1=true$ for some $\hat{\xi}$, $\hat{\pi}_0$ and $\hat{\pi}_1$, and take away $\hat{\xi}$ with $k\hat{\zeta}\hat{\pi}_0=true$. For every $n\geq 0$ $k_{n+1}\hat{\zeta}\hat{\pi}_0=true$ so $c\langle\mathfrak{B}q_n\acute{\xi}\acute{\zeta},\mathfrak{B}q_n\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_1=c_{n+1}\langle\mathfrak{B}q_n\acute{\xi}\acute{\zeta},\mathfrak{B}q_n\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_1=true$ and, c being inclusive, $c\langle\acute{\xi}\acute{\zeta},\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_1=true$. On the other hand if $\hat{\xi}$ is such that $c\langle\acute{\xi}\acute{\zeta},\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_1=true$ when $k\hat{\zeta}\hat{\pi}_0=true$, for any $n\geq 0$ and $\hat{\zeta}$ with $k_{n+1}\hat{\zeta}\hat{\pi}_0=true$ $k((\mathfrak{E}q_n\times\mathfrak{E}q_n)\hat{\zeta})\hat{\pi}_0=true$ so $c\langle\acute{\xi}(\mathfrak{E}q_n\acute{\zeta}),\grave{\xi}(\mathfrak{E}q_n\grave{\zeta})\rangle\hat{\pi}_1=true$, $c_{n+1}\langle\acute{\xi}(\mathfrak{E}q_n\acute{\zeta}),\grave{\xi}(\mathfrak{E}q_n\grave{\zeta})\rangle\hat{\pi}_1=true$ and $c_{n+1}\langle\mathfrak{B}q_n\acute{\xi}\acute{\zeta},\mathfrak{B}q_n\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_1=true$, giving $o_{n+1}\hat{\xi}\hat{\pi}_0\hat{\pi}_1=true$ for every $n\geq 0$ and $o\hat{\xi}\hat{\pi}_0\hat{\pi}_1=true$.⊁

2.4.9. Corollary.

Suppose that w, p, c, k, o and a are predicates such that $a\langle\perp,\perp\rangle=true$ and a is inclusive. Suppose further that for any $n\geq 0$ if every $\hat{\omega}$ and $\hat{\pi}$ having $kent0\hat{\omega}\hat{\pi}\wedge p_1\hat{\pi}=true$ satisfy $w\hat{\omega}\hat{\pi}\supset w_n\hat{\omega}\hat{\pi}$ and $w_n\hat{\omega}\hat{\pi}\supset w((q_n\times q_n)\hat{\omega}\hat{\pi})$ also then every $\hat{\delta}$ satisfies $a\hat{\delta}\supset a_{n+1}\hat{\delta}$ and $a_{n+1}\hat{\delta}\supset a((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})$. Should the conclusions of 2.4.8 hold these predicates will coincide with those set up above.

⊀The proof that w is unique in the sense just enunciated follows that of 2.2.6 too closely to have any interest. One value for a which obeys the conditions of this result will be given after 2.7.6.⊁

2.5. Denotation and allocation.

2.5.1. Lemma.

If E:Exp satisfies $G[\![E]\!]=true$ then $E[\![E]\!]=true$.

⊲Take any ψ_2 and $\hat{\pi}$ such that $apt\psi_2\hat{\rho}\wedge rent[\![E]\!]\psi_2\wedge fit\hat{\pi}\hat{\pi}=true$; to show that $o\langle\mathscr{E}[\![E]\!],\mathscr{E}[\![\bullet[\![E]\!]\psi_2]\!]\rangle\hat{\pi}\hat{\pi}=true$ it is enough to verify that whenever $k\hat{\zeta}_2\hat{\pi}\wedge p\hat{\pi}_0\wedge fit\hat{\pi}_0\hat{\pi}=true$ we have $a\langle\mathscr{E}[\![E]\!]\acute{\zeta}_2\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,\mathscr{E}[\![\bullet[\![E]\!]\psi_2]\!]\grave{\zeta}_2\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle=true$.

Let $\psi_1=\psi_2[false^*/\mathscr{I}[\![E]\!]][opts(\mathscr{K}[\![E]\!])\psi_2/\mathscr{K}[\![E]\!]]$,

$\langle\acute{\alpha}^*,\grave{\alpha}^*\rangle=\langle novels(\#\mathscr{I}[\![E]\!])\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,novels(\#\mathscr{I}[\![\bullet[\![E]\!]\psi_2]\!])\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle$,

$\hat{\zeta}_1=\langle\acute{\zeta}_2\circ revert\acute{\rho},\grave{\zeta}_2\circ revert\grave{\rho}\rangle$, $I^*=\mathscr{I}[\![\bullet[\![E]\!]\psi_2]\!]\,\S\mathscr{K}[\![\bullet[\![E]\!]\psi_2]\!]$,

$\hat{\rho}_1=\langle fix(\lambda\rho.\acute{\rho}_0[\acute{\alpha}^*/\mathscr{I}[\![E]\!]][\mathscr{L}[\![E]\!]\acute{\zeta}_1\rho\acute{\upsilon}_0/\mathscr{K}[\![E]\!]])$,

$\quad fix(\lambda\rho.\grave{\rho}_0[\grave{\alpha}^*/\mathscr{I}[\![\bullet[\![E]\!]\psi_2]\!]][\mathscr{L}[\![\bullet[\![E]\!]\psi_2]\!]\grave{\zeta}_1\rho\grave{\upsilon}_0/\mathscr{K}[\![\bullet[\![E]\!]\psi_2]\!]])\rangle$,

$\hat{\upsilon}_1=\hat{\upsilon}_0$ and

$\hat{\sigma}_1=\langle updates\acute{\alpha}^*(\mathscr{P}[\![E]\!]\acute{\zeta}_1\acute{\rho}_1\acute{\upsilon}_0)\acute{\sigma}_0,updates\grave{\alpha}^*(\mathscr{P}[\![\bullet[\![E]\!]\psi_2]\!]\grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_0)\grave{\sigma}_0\rangle$.

Since $p\hat{\pi}_0=true$ and L is infinite, $\acute{\alpha}^*$ and $\grave{\alpha}^*$ are proper and $\langle\mathscr{E}[\![E]\!]\acute{\zeta}_2\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,\mathscr{E}[\![\bullet[\![E]\!]\psi_2]\!]\grave{\zeta}_2\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle=\langle\mathscr{E}[\![E]\!]\acute{\zeta}_1\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{E}[\![\mathscr{I}[\![E]\!]\psi_1]\!]\grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle$. Moreover $G[\![E]\!]=true$ and obviously $apt\psi_1\hat{\pi}_1\wedge torn[\![E]\!]\psi_1=true$, so, this case being typical, to establish that $E[\![E]\!]=true$ we have merely to show that $k\hat{\zeta}_1\hat{\pi}_1\wedge p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_1=true$.

Let $p\hat{\pi}_2\wedge set\hat{\pi}_2\hat{\pi}_1=true$ for some $\hat{\pi}_2$ (on the assumption that such $\hat{\pi}_2$ exist); writing $\hat{\pi}_3=\langle\langle revert\acute{\rho}_0\acute{\rho}_2,\acute{\upsilon}_2,\acute{\sigma}_2\rangle,\langle revert\grave{\rho}_0\grave{\rho}_2,\grave{\upsilon}_2,\grave{\sigma}_2\rangle\rangle$ we have $set\hat{\pi}_3\hat{\pi}=set\hat{\pi}_2\hat{\pi}_1=true$. Furthermore by 2.1.7 and induction on ν_1, for all $\nu_0<2$, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3\wedge kent1\hat{\omega}_1\hat{\pi}_3\supset seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_2$ and, since $yclept\hat{\omega}_1\hat{\pi}_3\supset yclept\hat{\omega}_1\hat{\pi}_2\wedge kent1\hat{\omega}_1\hat{\pi}_3$, $kent\nu_0\hat{\omega}_0\hat{\pi}_3\supset kent\nu_0\hat{\omega}_0\hat{\pi}_2$. Hence $p\hat{\pi}_3=true$, $a\langle\acute{\zeta}_1\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle=a\langle\acute{\zeta}\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3,\grave{\zeta}\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3\rangle=true$ and $k\hat{\zeta}_1\hat{\pi}_1=true$.

⊲It remains to be shown that $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_1=true$. As an abbreviation we introduce the pair of stacks $\hat{\upsilon}_4$, which is

$\langle \mathfrak{A}^{*}\S\mathfrak{Q}[\![E]\!]\xi_1\rho_1\upsilon_0\S\mathfrak{P}[\![E]\!]\zeta_1\rho_1\upsilon_0\S\mathfrak{Q}[\![E]\!]\xi_1\rho_1\upsilon_0,$

$swap(\mathfrak{P}[\![E]\!]\S\mathfrak{R}[\![E]\!])\mathrm{I}^{*}(\mathfrak{A}^{*}\S\mathfrak{Q}[\![\epsilon[\![E]\!]\psi_2]\!]\grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_0)$

$\S swap(\mathfrak{P}[\![E]\!]\S\mathfrak{R}[\![E]\!])\mathrm{I}^{*}(\mathfrak{P}[\![\epsilon[\![E]\!]\psi_2]\!]\grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_0\S\mathfrak{Q}[\![\epsilon[\![E]\!]\psi_2]\!]\grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_0)\rangle$

For any $\hat{\omega}$ $gyven\hat{\omega}0_4 \supset kent1\hat{\omega}\Uparrow_1 \wedge gyven(access\hat{\omega}_1\Uparrow_1)0_4$ and we can now convince ourselves that when $\nu<2$ $kent\nu\hat{\omega}\Uparrow_1 \supset gyven\hat{\omega}0_4 \vee kent\nu\hat{\omega}\Uparrow_0$ as follows.

◁The assertion that

$seen\nu_0\boldsymbol{\nu}_1\hat{\omega}_0\hat{\omega}_1\Uparrow_1 \wedge (gyven\hat{\omega}_1 0_4 \vee kent1\hat{\omega}_1\Uparrow_0) \supset gyven\hat{\omega}_1 0_4 \vee kent\nu_0\hat{\omega}_0\Uparrow_0$ holds for all $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ when $\nu_1<1$. Suppose that it holds for all $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ and for some ν_1; we shall show that it holds for ν_1+1.

Let $seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\Uparrow_1 = true$ and let $hoten\hat{\omega}_1 0_4 = true$ or $kent1\hat{\omega}_1\Uparrow_0 = true$. If $\hat{\omega}_1 : \mathsf{L}$ and $gyven\hat{\omega}_1 0_4 = true$, $site\acute{\omega}_1\beta_0\upsilon_0\delta_0 = false$ so $kent1\hat{\omega}_1\Uparrow_1 = false$ whilst if $\hat{\omega}_1 : \mathsf{L}$ and $kent1\hat{\omega}_1\Uparrow_1 = false$ $gyven\hat{\omega}_1 0_4 = false$; similar remarks pertain when $\grave{\omega}_1 : \mathsf{L}$. Hence if $\acute{\omega}_1 : \mathsf{L}$ or $\grave{\omega}_1 : \mathsf{L}$, $access\hat{\omega}_1\Uparrow_1 = (gyven\hat{\omega}_1 0_4 \rightarrow access\hat{\omega}_1\Uparrow_1, access\hat{\omega}_1\Uparrow_0)$ and $seen\nu_0\nu_1\hat{\omega}_0(access\hat{\omega}_1\Uparrow_1)\Uparrow_1 = true$ where $gyven(access\hat{\omega}_1\Uparrow_1)0_4 = true$ or $kent1(access\hat{\omega}_1\Uparrow_1)\Uparrow_0 = true$. If $\hat{\omega}_1 : \mathsf{J}\times\mathsf{J}$, there is some $\hat{\omega}_2$ with $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2\Uparrow_1 = true$ and $hoten\hat{\omega}_2\langle\acute{\omega}_1\downarrow 2, \grave{\omega}_1\downarrow 2\rangle = true$ or $gyven\hat{\omega}_2\langle\acute{\omega}_1\downarrow 3, \grave{\omega}_1\downarrow 3\rangle = true$; moreover if $gyven\hat{\omega}_1 0_4 = true$ either $hoten\hat{\omega}_2 0_4 = true$ or $yclept\hat{\omega}_2\Uparrow_0 = true$ (from the definition of $\hat{\rho}_1$ as a pair of fixed points) whilst if $kent1\hat{\omega}_1\Uparrow_0 = true$ $kent1\hat{\omega}_2\Uparrow_0 = true$, and accordingly $gyven\hat{\omega}_2 0_4 = true$ or $kent1\hat{\omega}_2\Uparrow_0 = true$. If $\hat{\omega}_1 : \mathsf{F}\times\mathsf{F}$, $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2\Uparrow_1 = true$ for some $\hat{\omega}_2$ such that $hoten\hat{\omega}_2\langle\acute{\omega}_1\downarrow 2, \grave{\omega}_1\downarrow 2\rangle = true$; since the elements of $\acute{\upsilon}_4$ and $\grave{\upsilon}_4$ are all locations or label entry points, $kent1\hat{\omega}_1\Uparrow_0 = true$ and $kent1\hat{\omega}_2\Uparrow_0 = true$. Analogous remarks hold when $\hat{\omega}_1 : \mathsf{L}^{*}\times\mathsf{L}^{*}$ or $\hat{\omega}_1 : \mathsf{J}\times\mathsf{J}$ so, unless possibly $\hat{\omega}_1 : \mathsf{G}\times\mathsf{G}$ or $\hat{\omega}_1 : \mathsf{P}\times\mathsf{P}$, there is some $\hat{\omega}_2$ with $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2\Uparrow_1 = true$ and with $gyven\hat{\omega}_2 0_4 = true$ or $kent1\hat{\omega}_2\Uparrow_0 = true$; by the induction hypothesis $gyven\hat{\omega}_0 0_4 = true$ or $kent\nu_0\hat{\omega}_0\Uparrow_0 = true$. When there is no such $\hat{\omega}_2$ and

when $\hat{\omega}_1$ is a pair in G×G,

$$seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_0=\bigvee\{seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2(\langle\vec{\omega}_1\downarrow 2,\langle\rangle,\vec{\omega}_1\downarrow 3\rangle,\langle\grave{\omega}_1\downarrow 2,\langle\rangle,\grave{\omega}_1\downarrow 3\rangle\rangle$$
$$\wedge yclept\hat{\omega}_2(\langle\vec{\omega}_1\downarrow 2,\langle\rangle,\vec{\omega}_1\downarrow 3\rangle,\langle\grave{\omega}_1\downarrow 2,\langle\rangle,\grave{\omega}_1\downarrow 3\rangle)\mid\hat{\omega}_1:W\times W\}$$
$$=seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_1$$
$$=true$$

and similar remarks apply when $\hat{\omega}_1:P\times P$. From 2.1.8 we can deduce that in these cases also $kent\nu_0\hat{\omega}_0\hat{\pi}=true$, thereby completing the step in the induction.

Consequently the assertion above is valid for all ν_1, and in particular for every $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$, as $yclept\hat{\omega}_1\hat{\pi}_1\supset gyven\hat{\omega}_1 0_4\vee kent1\hat{\omega}_1\hat{\pi}_0$, we can infer that $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_1\wedge yclept\hat{\omega}_1\hat{\pi}_1\supset gyven\hat{\omega}_1 0_4\vee kent\nu_0\hat{\omega}_1\hat{\pi}_0$. From 2.1.6 $kent\nu\hat{\omega}\hat{\pi}_1\supset gyven\hat{\omega}0_4\vee kent\nu\hat{\omega}\hat{\pi}_0$ for all $\nu<2$ and $\hat{\omega}$.⊁

These paragraphs make plain that $p_0\hat{\pi}_1=true$ and that when $kent0\hat{\omega}\hat{\pi}_1=true$ but $gyven\hat{\omega}0_4=false$ $w\hat{\omega}\hat{\pi}_1=true$. Indeed, if $gyven\hat{\omega}0_4=true$ and $\vec{\omega}:L$ or $\grave{\omega}:L$ necessarily

$$\bigwedge\{\sim((\vec{\omega}:L\wedge\vec{\omega}=\vec{\omega}_0)\vee(\grave{\omega}:L\wedge\grave{\omega}=\grave{\omega}_0))\vee(\hat{\omega}:E\times E\wedge\hat{\omega}=\hat{\omega}_0)\mid kent0\hat{\omega}_0\hat{\pi}_1\}=true$$

so only the case when $\hat{\omega}:J\times J$ and $gyven\hat{\omega}0_4=true$ requires further consideration before we may conclude that $p\hat{\pi}_1=true$. Clearly any $\hat{\omega}$ having $gyven\hat{\omega}0_4=true$ satisfies $w_0\hat{\omega}\hat{\pi}_1=true$; coupled with the fact that $p\hat{\pi}_0=true$ this establishes that $p\hat{\pi}_1=true$ and thus that there are indeed pairs $\hat{\pi}_2$ having $p\hat{\pi}_2\wedge set\hat{\pi}_2\hat{\pi}_1=true$. Hence $k\hat{\zeta}_1\hat{\pi}_1=true$, and the premises of the lemma now indicate that any $\hat{\omega}:J\times J$ with $gyven\hat{\omega}0_4=true$ is subject to $c\langle\vec{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\langle\vec{\omega}\dagger 1\S\langle\vec{\sigma}_1\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_1\rangle\rangle=true$; this ensures that $p\hat{\pi}_0=true$.⊁

We have therefore demonstrated that when

$$apt\psi_2\hat{\rho}\wedge rent[\![E]\!]\psi_2\wedge fit\hat{\pi}\hat{\pi}=true\ o\langle\mathscr{E}[\![E]\!],\mathscr{E}[\![\bullet[\![E]\!]\psi_2]\!]\rangle\hat{\pi}\hat{\pi}=true$$

for all ψ_2 and $\hat{\pi}$. In terms of the definitions of 2.4.5 this means that $E[\![E]\!]=true$.⊁

Note that this result does not use any of the properties of *fix* other than that of producing a fixed point. Were there

another means of solving the recursive equations on environments which *fix* deals with above it too could be shown to be exactly equivalent with the technique which stores label entry points.

We are now in a position to resolve the problem raised in 1.3.1 concerning the relation between a while loop and a program containing a conditional expression and a label instead. In standard semantics while E_0 do E_1 and I::E_2, where E_2 is if E_0 then E_1; goto I else dummy, can be shown by induction and the use of 1.5.2 to be identical in their effects so long as I is not free in E_0 or in E_1. If $R[\![E_0]\!]=true$ and E_1 satisfies the hypotheses of the lemma above, I::E_2 and I:E_2 correspond in store semantics (using *new* instead of *novel*). Under the restrictions laid down in 2.6.9 this sort of semantics coincides with standard semantics so the standard translations of I::E_2 and I:E_2, and thus of while E_0 do $E_1$and I:E_2, are such that there are no discernible differences between them.

2.5.2. Lemma.

If E:Exp satisfies $E[\![E]\!]=true$ then $L[\![E]\!]=true$ and $R[\![E]\!]=true$.

⊲Inspecting the definitions of 2.4.5 reveals that the result is entailed by $k\langle mv\acute{\zeta},mv\grave{\zeta}\rangle\hat{\pi}\wedge k\langle sv\acute{\zeta},sv\grave{\zeta}\rangle\hat{\pi}=true$ for every $\hat{\zeta}$ and $\hat{\pi}$ having $k\hat{\zeta}\hat{\pi}=true$. Taking some such $\hat{\zeta}$ and $\hat{\pi}$ it thus suffices to prove that $\hat{\zeta}$ satisfies

$a\langle mv\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,mv\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle\wedge a\langle sv\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,sv\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle=true$

for some typical $\hat{\pi}_0$ such that $p\hat{\pi}_0\wedge set\hat{\pi}_0\hat{\pi}=true$; let $\hat{\varepsilon}_0=\langle\acute{\upsilon}_0\downarrow 1,\grave{\upsilon}_0\downarrow 1\rangle$.

Owing to the fact that $p_0\hat{\pi}_0=true$, if $\acute{\varepsilon}_0$:L $area\acute{\varepsilon}_0\acute{\sigma}_0=true$ whilst if $\acute{\varepsilon}_0$:V $novel\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0$ is proper, and we may sensibly define $\acute{\varepsilon}_1$ and $\acute{\varepsilon}_2$ to be $(\acute{\varepsilon}_0$:L→$\acute{\varepsilon}_0,novel\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0)$ and $(\acute{\varepsilon}_0$:L→$hold\acute{\varepsilon}_0\acute{\sigma}_0,\acute{\varepsilon}_0)$ respectively. Setting $\hat{\pi}_1=\langle\acute{\rho}_0,\langle\acute{\varepsilon}_1\rangle\S\acute{\upsilon}_0\dagger 1,update\acute{\varepsilon}_1\acute{\varepsilon}_2\acute{\sigma}_0\rangle$ and $\hat{\pi}_2=\langle\acute{\rho}_0,\langle\acute{\varepsilon}_2\rangle\S\acute{\upsilon}_0\dagger 1,\acute{\sigma}_0\rangle$, $mv\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0=\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1$ and $sv\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0=\acute{\zeta}\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2$;

analogous definitions apply to $\grave{\pi}_0$, and in terms of them it suffices to prove that $p\hat{\pi}_1 \wedge set\hat{\pi}_1\hat{\pi} = true$ and $p\hat{\pi}_2 \wedge set\hat{\pi}_2\hat{\pi} = true$.

By inductions like that of 2.5.1 we can show that for all $\nu<2$ and $\hat{\omega}$ $kent\nu\hat{\omega}\hat{\pi}_1 \supset (\hat{\omega} = \hat{\varepsilon}_1 \vee kent\nu\hat{\omega}\hat{\pi}_0)$ and $kent\nu\hat{\omega}\hat{\pi}_2 \supset (\hat{\omega} = \hat{\varepsilon}_2 \vee kent\nu\hat{\omega}\hat{\pi}_0)$. Since $access\hat{\varepsilon}_0\hat{\pi} = \hat{\varepsilon}_1$, actually for all $\nu<2$ and $\hat{\omega}$ $kent\nu\hat{\omega}\hat{\pi}_2 \supset kent\nu\hat{\omega}\hat{\pi}_0$ so that $w\hat{\omega}\hat{\pi}_0 \supset w\hat{\omega}\hat{\pi}_2$ for every $\hat{\omega}$ and $p\hat{\pi}_2 = true$. Because $\hat{\varepsilon}_2 : V \times V$ we have $set\hat{\pi}_2\hat{\pi} = true$, $a\langle \acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, \grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle = true$ and $k\langle sv\acute{\zeta}, sv\grave{\zeta}\rangle\hat{\pi} = true$ for every $\hat{\zeta}$ and $\hat{\pi}$ satisfying $k\hat{\zeta}\hat{\pi} = true$.

To establish that $p\hat{\pi}_1 = true$ it is merely necessary to show that $p_0\hat{\pi}_1 = true$. Three conceivable circumstances exist, corresponding to whether $\hat{\varepsilon}_0 : L \times L$, $\hat{\varepsilon}_0 : V \times V$ or $\acute{\varepsilon}_0 : V$ and $\grave{\varepsilon}_0 : L$. In the first of these, $\hat{\varepsilon}_1 = \hat{\varepsilon}_0$ and $\hat{\pi}_1 = \hat{\pi}_0$ so that $p\hat{\pi}_1 = true$. In the second, $\hat{\varepsilon}_1 = \langle novel\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, novel\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle$ so that $site\acute{\varepsilon}_1\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0 \vee site\grave{\varepsilon}_1\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0 = false$ and, as pointed out in 2.4.5, $p_1\hat{\pi}_0$ being $true$ there can be no $\hat{\omega}_0$ with $kent0\hat{\omega}_0\hat{\pi}_0 = true$ and $\acute{\varepsilon}_1 = \acute{\omega}_0$ or $\grave{\varepsilon}_1 = \grave{\omega}_0$; hence $\bigwedge\{\sim((\acute{\omega} : L \wedge \acute{\omega} = \acute{\omega}_0) \vee (\grave{\omega} : L \wedge \grave{\omega} = \grave{\omega}_0)) \vee (\hat{\omega} : E \times E \wedge \hat{\omega} = \hat{\omega}_0) \mid kent0\hat{\omega}_0\hat{\pi}_1\} = true$ when $kent0\hat{\omega}\hat{\pi}_1 = true$ and $\acute{\omega} : L$ or $\grave{\omega} : L$. In the third, $\hat{\varepsilon}_1 = \langle novel\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, \grave{\varepsilon}_0\rangle$ where $site\acute{\varepsilon}_0\acute{\rho}_0(\langle \acute{\varepsilon}_2\rangle \S\acute{\upsilon}_0 \dagger 1)\acute{\sigma}_0 = false$ since $set\hat{\pi}_1\hat{\pi} = true$; $p\langle \hat{\pi}_0, \langle \grave{\rho}_0, \langle \grave{\varepsilon}_2\rangle \S\grave{\upsilon}_0 \dagger 1, \grave{\sigma}_0\rangle\rangle$ being $true$ we may argue as in the second case that $p_0\hat{\pi}_1 = true$. In consequence, $p\hat{\pi}_1 = true$ whatever the nature of $\hat{\varepsilon}_0$, and $set\hat{\pi}_1\hat{\pi} = true$ as $\hat{\varepsilon}_1 : L \times L$, so that $a\langle mv\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, mv\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle = true$ and $k\langle mv\acute{\zeta}, mv\grave{\zeta}\rangle\hat{\pi} = true$. ⊁

This lemma thus validates the contention of 2.4.1 that if a member of V and a member of L are paired at the top of their respective stacks they will quickly be replaced by a pair in either L×L or V×V. In particular, by the time two stacks are preserved as part of comparable label entry points locations will be coupled only with locations.

2.5.3. Lemma.

For all I:Ide and B:Bas $G[\![I]\!] \wedge G[\![B]\!] = true$.

⊲Suppose that ψ, Φ and $\hat{\zeta}$ are such that $apt\psi\hat{\rho} \wedge torn[\![I]\!]\psi \wedge k\hat{\zeta}\hat{\Phi} = true$, and take any Φ_0 having $fit\Phi_0\Phi \wedge p\Phi_0 = true$. We shall show that $a\langle \mathscr{G}[\![I]\!]\zeta\rho_0\upsilon_0\sigma_0, \mathscr{G}[\![g[\![I]\!]\psi]\!]\zeta\rho_0\upsilon_0\sigma_0\rangle = true$. Because $torn[\![I]\!]\psi \wedge apt\psi\hat{\rho}_0 = true$ $\hat{\delta}$, which we define to be $\langle \rho_0[\![I]\!]\downarrow 1, \vartheta_0[\![I]\!]\downarrow 1\rangle$, is proper.

⊲Assume first that $\psi[\![I]\!]\downarrow 1 = true$, so that by the definition of apt δ:V and $\grave{\delta}$:L or $\grave{\delta}$:V while from 1.4.6 $\mathscr{G}[\![g[\![I]\!]\psi]\!] = \mathscr{G}[\![\$I]\!] = \mathscr{A}[\![I]\!] \circ mv$. Writing $\grave{\beta} = (\grave{\delta}:L \rightarrow (area\grave{\delta}\grave{\sigma}_0 \rightarrow hold\grave{\delta}\grave{\sigma}_0, \tau), \grave{\delta})$, $\grave{\alpha} = novel\grave{\rho}_0(\langle \grave{\beta}\rangle \S\vartheta_0)\vartheta_0$ and $\Phi_1 = \langle \langle \rho_0, \langle \delta\rangle \S\upsilon_0, \sigma_0\rangle, \langle \rho_0, \langle \grave{\alpha}\rangle \S\vartheta_0, update\grave{\alpha}\grave{\beta}\grave{\sigma}_0\rangle\rangle$, we know that $\grave{\beta}$ is proper ($p_0\Phi_0$ and $plot\grave{\delta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0$ both being $true$ if $\grave{\delta}$:L) and $\langle \mathscr{G}[\![I]\!]\zeta\rho_0\upsilon_0\sigma_0, \mathscr{G}[\![g[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle = \langle \zeta\rho_1\upsilon_1\sigma_1, \grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle$. We need now verify only that $p\Phi_1 = true$, since from 2.1.6 $site\grave{\alpha}\grave{\rho}_0(\langle \grave{\beta}\rangle \S\vartheta_0)\vartheta_0 = true$ and $set\Phi_1\Phi = true$. In fact we shall show that $kent\nu\hat{\omega}\Phi_1 \supset (\hat{\omega} = \langle \delta, \grave{\alpha}\rangle \vee kent\nu\hat{\omega}\Phi_0)$ for all $\nu < 2$ and $\hat{\omega}$.

⊲Plainly $yclept\hat{\omega}\Phi_1 \supset (\hat{\omega} = \langle \delta, \grave{\alpha}\rangle \vee kent1\hat{\omega}\Phi_0)$, $access\langle \delta, \grave{\alpha}\rangle \Phi_1 = \hat{\delta}$ and $kent1(access\langle \delta, \grave{\alpha}\rangle \Phi_1)\Phi_0 = true$. Suppose that for all $\nu_0 < 2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ and for a certain ν_1 $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\Phi_1 \wedge kent1\hat{\omega}_1\Phi_0 \supset kent1\hat{\omega}_1\Phi_1$, and take any $\nu_0 < 2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ with $seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\Phi_1 \wedge kent1\hat{\omega}_1\Phi_0 = true$.

If $\grave{\omega}_1$:L, $site\grave{\omega}_1\grave{\rho}_0\grave{\upsilon}_0\grave{\vartheta}_0 = true$ so $\grave{\omega}_1$ is not $\grave{\alpha}$ and $hold\grave{\omega}_1\grave{\vartheta}_1$ is $hold\grave{\omega}_1\grave{\vartheta}_0$. Hence when $\acute{\omega}_1$:L or $\grave{\omega}_1$:L $access\hat{\omega}_1\Phi_1 = access\hat{\omega}_1\Phi_0$ and $seen\nu_0\nu_1\hat{\omega}_0(access\hat{\omega}_1\Phi_1)\Phi_1 \wedge kent1(access\hat{\omega}_1\Phi_1)\Phi_0 = true$. When $\hat{\omega}_1$:L*×L*, $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2\Phi_1 = true$ for some $\hat{\omega}_2$ with $hoten\hat{\omega}_2\hat{\omega}_1 = true$ and $kent1\hat{\omega}_2\Phi_0 = true$ by 2.1.8; similarly *when* $\hat{\omega}_1$:J×J, $\hat{\omega}_1$:F×F or $\hat{\omega}_1$:J×J $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2\Phi_1 = true$ for some $\hat{\omega}_2$ having $kent1\hat{\omega}_2\Phi_0 = true$. In all these cases the induction hypothesis permits the conclusion that $kent\nu_0\hat{\omega}_0\Phi_0 = true$. When $\hat{\omega}_1$:P×P necessarily $\nu_0 = 0$ and

$$\begin{aligned} seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\Phi_0 &= \vee\{seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2\hat{\omega}_1 \wedge yclept\hat{\omega}_2\hat{\omega}_1 \mid \hat{\omega}_2 : W\times W\} \\ &= seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\Phi_1 \\ &= true; \end{aligned}$$

likewise when $\hat{\omega}_1 : G\times G$ $seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_0 = true$ so in both cases $kent0\hat{\omega}_0\hat{\pi}_0 = true$ from 2.1.8.

This being so, for all $\nu_0<2$, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_1 \wedge kent1\hat{\omega}_1\hat{\pi}_0 \supset kent1\hat{\omega}_0\hat{\pi}_0$ as this assertion is valid when $\nu_1<1$. Hence for all $\nu<2$ and $\hat{\omega}$

$$kent\nu\hat{\omega}\hat{\pi}_1 \supset (\hat{\omega}=\langle\hat{\delta},\grave{\alpha}\rangle \vee \bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_1 \wedge (\hat{\omega}_1=\hat{\delta}\vee yclept\hat{\omega}_1\hat{\pi}_0)\}\})$$
$$\supset (\hat{\omega}=\langle\hat{\delta},\grave{\alpha}\rangle \vee \bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_0 \wedge kent1\hat{\omega}_1\hat{\pi}_0\}\})$$
$$\supset (\hat{\omega}=\langle\hat{\delta},\grave{\alpha}\rangle \vee kent\nu\hat{\omega}\hat{\pi}_0),$$

which provides what was required.⊁

Because $site\grave{\alpha}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0 = false$ for no ν and $\acute{\varepsilon}$ does $kent\nu\langle\acute{\varepsilon},\grave{\alpha}\rangle\hat{\pi}_0 = true$, and therefore $p_0\hat{\pi}_1 = true$. Moreover unless $\hat{\omega}=\grave{\alpha}$ $kent0\hat{\omega}\hat{\pi}_1 \supset kent0\hat{\omega}\hat{\pi}_0 \supset w\hat{\omega}\hat{\pi}_0 \supset w\hat{\omega}\hat{\pi}_1$; hence $p\hat{\pi}_1 = true$ and $a\langle\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle = true$.⊁

Now assume that $\psi[\![I]\!]\downarrow 1 = false$, so that $\hat{\delta}:L\times L$, $\hat{\delta}:V\times V$ or $\hat{\delta}:G\times G$. In the first two cases,

$\langle\mathscr{S}[\![I]\!]\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, \mathscr{S}[\![\mathscr{I}[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle = \langle\acute{\zeta}\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\grave{\zeta}\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle$ where $\hat{\pi}_2 = \langle\langle\acute{\rho}_0,\langle\acute{\delta}\rangle\S\acute{\upsilon}_0,\acute{\sigma}_0\rangle,\langle\grave{\rho}_0,\langle\grave{\delta}\rangle\S\grave{\upsilon}_0,\grave{\sigma}_0\rangle\rangle$. Because $hoten\hat{\delta}\hat{\rho}_0 = true$, $kent\nu\hat{\omega}\hat{\pi}_2 \supset kent\nu\hat{\omega}\hat{\pi}_0$ for all $\nu<2$ and $\hat{\omega}$ and $p\hat{\pi}_2 = true$. As $set\hat{\pi}_2\hat{\pi}\wedge k\hat{\zeta}\hat{\pi} = true$ we can infer that

$a\langle\mathscr{S}[\![I]\!]\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, \mathscr{S}[\![\mathscr{I}[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle = true$.

When $\hat{\delta}:G\times G$ define $\hat{\xi}=\langle\acute{\delta}\downarrow 1,\grave{\delta}\downarrow 1\rangle$, $\hat{\pi}_3 = \langle\langle(\acute{\delta}\downarrow 2)[\hat{\pi}_0/rec],\langle\rangle,\acute{\delta}\downarrow 3\rangle,\langle(\grave{\delta}\downarrow 2)[\hat{\pi}_0/rec],\langle\rangle,\grave{\delta}\downarrow 3\rangle\rangle$, so that $\langle\mathscr{S}[\![I]\!]\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, \mathscr{S}[\![\mathscr{I}[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle = \langle\acute{\xi}\acute{\zeta}\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3,\grave{\xi}\grave{\zeta}\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3\rangle$. Since $kent1\hat{\delta}\hat{\pi}_0 = true$ and $k\hat{\zeta}\hat{\pi}_0 = true$, $c\langle\acute{\xi}\acute{\zeta},\grave{\xi}\grave{\zeta}\rangle\hat{\pi}_3 = true$ and in order that $a\langle\acute{\xi}\acute{\zeta}\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3,\grave{\xi}\grave{\zeta}\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3\rangle = true$ it is enough that $p\hat{\pi}_3 = true$. Now for any $\nu<2$ and $\hat{\omega}$, writing $\hat{\pi}_4 = \langle\langle\acute{\delta}\downarrow 2,\langle\rangle,\acute{\delta}\downarrow 3\rangle,\langle\grave{\delta}\downarrow 2,\langle\rangle,\grave{\delta}\downarrow 3\rangle\rangle$,

$$kent\nu\hat{\omega}\hat{\pi}_3 = (\bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_3 \wedge yclept\hat{\omega}_1\hat{\pi}_4\}\} \vee \hat{\omega}=\hat{\pi}_0$$
$$\vee(\nu=0\wedge\bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_0 \wedge yclept\hat{\omega}_1\hat{\pi}_0\}\}))$$
$$= (\bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_4 \wedge yclept\hat{\omega}_1\hat{\pi}_4\}\} \vee \hat{\omega}=\hat{\pi}_0 \vee (\nu=0\wedge kent0\hat{\omega}\hat{\pi}_0))$$

since $seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_3 = seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_4$ for all ν_1 and $\hat{\omega}_1$ by definition. In particular $kent0\hat{\omega}\hat{\pi}_3 \supset (\hat{\omega}=\hat{\pi}_0 \vee kent0\hat{\omega}\hat{\pi}_0)$ and

$kent1\hat{\omega}\hat{\pi}_3 \supset (\vee\{seen1\nu_1\hat{\omega}\hat{\delta}\hat{\pi}_0\} \vee \hat{\omega}=\hat{\pi}_0) \supset (\hat{\omega}=\hat{\pi}_0 \vee kent1\hat{\omega}\hat{\pi}_4)$. Since $p\hat{\pi}_0=true$ $w_0\hat{\pi}_0\hat{\pi}_3=true$ and since $hoten\hat{\delta}\hat{\rho}_0=true$ $p_0\hat{\pi}_4=true$; accordingly $kent0\hat{\omega}\hat{\pi}_3 \supset w\hat{\omega}\hat{\pi}_3$ and $p_0\hat{\pi}_3=true$ so that $p\hat{\pi}_3=true$ as required.

Hence whenever ψ, $\hat{\pi}$ and $\hat{\zeta}$ are such that $apt\psi\hat{\rho} \wedge torn[\![I]\!]\psi \wedge k\hat{\zeta}\hat{\pi}=true$ $c\langle \mathscr{G}[\![I]\!]\acute{\zeta}, \mathscr{G}[\![\mathscr{I}[\![I]\!]\psi]\!]\grave{\zeta}\rangle \hat{\pi}=true$, so that $o\langle \mathscr{G}[\![I]\!], \mathscr{G}[\![\mathscr{I}[\![I]\!]\psi]\!]\rangle \hat{\pi}\hat{\pi}=true$ and $G[\![I]\!]=true$. Similarly $o\langle \mathscr{G}[\![B]\!], \mathscr{G}[\![\mathscr{I}[\![B]\!]\psi]\!]\rangle \hat{\pi}\hat{\pi}=true$, for given any $\hat{\pi}_0$ setting $\hat{\beta}=\langle \mathscr{B}[\![B]\!], \mathscr{M}[\![B]\!]\rangle$ yields $b\hat{\beta}=true$ and

$kent\nu\hat{\omega}\langle \langle \acute{\rho}_0, \langle \acute{\beta}\rangle \S\acute{\upsilon}_0, \acute{\sigma}_0\rangle, \langle \grave{\rho}_0, \langle \grave{\beta}\rangle \S\grave{\upsilon}_0, \grave{\sigma}_0\rangle\rangle \supset (\hat{\omega}=\hat{\beta} \vee kent\nu\hat{\omega}\hat{\pi}_0)$

for all ν and $\hat{\omega}$.⊁

Were it not for the fact that the transformation of I into $I makes stored values correspond with fresh locations rather than with their contents it would be possible to eschew the distinction between c and k adopted here.

2.5.4. Lemma.

If $L[\![E]\!]=true$ then $G[\![\mathsf{fn}()E]\!]=true$,

$G[\![\mathsf{fn}I.E]\!] \wedge G[\![\mathsf{fn}I_1,\ldots,I_n.E]\!]=true$ and $G[\![\mathsf{fn}I..E]\!] \wedge G[\![\mathsf{fn}I_1,\ldots,I_n..E]\!]=true$.

⊀Suppose $\hat{\pi}$ and $\hat{\pi}_0$ are pairs satisfying $p\hat{\pi}_0 \wedge fit\hat{\pi}_0\hat{\pi}=true$ and that $\hat{\varepsilon}_0$:F×F is such that $w\hat{\varepsilon}_0\hat{\pi}_0=true$ and $hoten\hat{\omega}\langle \acute{\varepsilon}_0\downarrow 2, \grave{\varepsilon}_0\downarrow 2\rangle \supset hoten\hat{\omega}\hat{\rho}_0$ for all $\hat{\omega}$. Taking $\hat{\pi}_1$ to be $\langle\langle \acute{\rho}_0, \langle \acute{\varepsilon}_0\rangle \S\acute{\upsilon}_0, \acute{\sigma}_0\rangle, \langle \grave{\rho}_0, \langle \grave{\varepsilon}_0\rangle \S\grave{\upsilon}_0, \grave{\sigma}_0\rangle\rangle$ we have for every $\nu<2$ and $\hat{\omega}$

$kent\nu\hat{\omega}\hat{\pi}_1=\vee\{\vee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_1 \wedge (\hat{\omega}_1=\hat{\varepsilon}_0 \vee yclept\hat{\omega}_1\hat{\pi}_0)\}\}$

$=(\vee\{\vee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_1 \wedge (hoten\hat{\omega}_1\langle \acute{\varepsilon}_0\downarrow 2, \grave{\varepsilon}_0\downarrow 2\rangle \vee yclept\hat{\omega}_1\hat{\pi}_0)\}\} \vee \hat{\omega}=\hat{\varepsilon}_0)$

$=(\vee\{\vee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_1 \wedge (hoten\hat{\omega}_1\hat{\rho}_0 \vee yclept\hat{\omega}_1\hat{\pi}_0)\}\} \vee \hat{\omega}=\hat{\varepsilon}_0)$

$=(\vee\{\vee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_0 \wedge yclept\hat{\omega}_1\hat{\pi}_0\}\} \vee \hat{\omega}=\hat{\varepsilon}_0)$

$=(\hat{\omega}=\hat{\varepsilon}_0 \vee kent\nu\hat{\omega}\hat{\pi}_0)$.

The nature of w being such that $w\hat{\omega}\hat{\pi}_0 \supset w\hat{\omega}\hat{\pi}_1$ for all $\hat{\omega}$, $p\hat{\pi}_1 \wedge set\hat{\pi}_1\hat{\pi}=true$.

Given any $\hat{\zeta}$ and $\hat{\pi}_0$, for every abstraction Φ there is a pair

$\hat{\varepsilon}_0$:F×F such that for all $\hat{\omega}$ $hoten\hat{\omega}\langle \acute{\varepsilon}_0\downarrow 2, \grave{\varepsilon}_0\downarrow 2\rangle \supset hoten\hat{\omega}\hat{\rho}_0$ and such that $\mathscr{G}[\![\Phi]\!]\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0 = \acute{\zeta}\acute{\rho}_0(\langle \acute{\varepsilon}_0\rangle \S\acute{\upsilon}_0)\acute{\sigma}_0$ and $\mathscr{G}[\![\gamma[\![\Phi]\!]\psi]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0 = \grave{\zeta}\grave{\rho}_0(\langle \grave{\varepsilon}_0\rangle \S\grave{\upsilon}_0)\grave{\sigma}_0$. In particular if $apt\psi\hat{\rho}\wedge torn[\![\Phi]\!]\psi = true$ and $k\hat{\zeta}\hat{\pi}\wedge p\hat{\pi}_0\wedge fit\hat{\pi}_0\hat{\pi} = true$ it is enough to show that $w\hat{\varepsilon}_0\hat{\pi}_0 = true$ to convince us that, in the notation above, $a\langle \acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, \grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle = true$. Moreover we can even conclude that $G[\![\Phi]\!] = true$ if the fact that $w\hat{\varepsilon}_0\hat{\pi}_0 = true$ does not depend on special properties of $\hat{\pi}_0$.

We shall ignore the possibility that Φ could be of the form fn()E as the necessary proof is less interesting than the following ones. Let the abstraction firstly be fnI.E, and let ψ and $\hat{\rho}$ satisfy $apt\psi\hat{\rho}\wedge torn[\![\text{fnI.E}]\!]\psi = true$. If $\acute{\xi}_0$ and $\grave{\xi}_0$ are $\lambda\zeta\rho\upsilon.\mathscr{L}[\![\text{E}]\!]\zeta\rho[\upsilon\downarrow 1/\text{I}](\upsilon\dagger 1)$ and $\lambda\zeta\rho\upsilon.\mathscr{L}[\![\gamma[\![\text{E}]\!]\psi[false/\text{I}]]\!]\zeta\rho[\upsilon\downarrow 1/\text{I}](\upsilon\dagger 1)$ respectively, for any $\hat{\zeta}$ and $\hat{\pi}_0$
$\mathscr{G}[\![\text{fnI.E}]\!]\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0$ is $\acute{\zeta}\acute{\rho}_0(\langle\langle \acute{\xi}_0, rend[\![\text{fnI.E}]\!]\acute{\rho}_0\rangle\rangle \S\acute{\upsilon}_0)\acute{\sigma}_0$ and
$\mathscr{G}[\![\gamma[\![\text{fnI.E}]\!]\psi]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0$ is $\grave{\zeta}\grave{\rho}_0(\langle\langle \grave{\xi}_0, rend[\![\text{fnI.E}]\!]\grave{\rho}_0\rangle\rangle \S\grave{\upsilon}_0)\grave{\sigma}_0$.
In accordance with the remarks above it suffices to prove that all $\hat{\zeta}_1$, $\hat{\pi}_1$ and $\hat{\pi}_2$ having
$\hat{\rho}_2 = \langle divert\acute{\rho}_1(rend[\![\text{fnI.E}]\!]\acute{\rho}_0), divert\grave{\rho}_1(rend[\![\text{fnI.E}]\!]\grave{\rho}_0)\rangle$,
$0_1 = \langle \acute{\upsilon}_2\dagger 1, \grave{\upsilon}_2\dagger 1\rangle$, $fit\hat{\pi}_2\hat{\pi}_2 = true$ and $k\hat{\zeta}_1\hat{\pi}_1 = true$ satisfy
$c\langle \acute{\xi}_0(\acute{\zeta}_1\circ revert\acute{\rho}_1), \grave{\xi}_0(\grave{\zeta}_1\circ revert\grave{\rho}_1)\rangle\hat{\pi}_2 = true$. Suppose that for some such $\hat{\zeta}_1$, $\hat{\pi}_1$ and $\hat{\pi}_2$ there exists a pair $\hat{\pi}_3$ with $p\hat{\pi}_3\wedge fit\hat{\pi}_3\hat{\pi}_2 = true$; taking $\hat{\pi}_4$ to be $\langle \acute{\rho}_3[\acute{\upsilon}_3\downarrow 1/\text{I}], \acute{\upsilon}_3\dagger 1, \acute{\sigma}_3\rangle$ (and similarly for $\hat{\pi}_4$) gives $apt\psi\hat{\rho}_4\wedge rent[\![\text{E}]\!]\acute{\rho}_4 = true$ and, as $kent\nu\hat{\omega}\hat{\pi}_4 \supset kent\nu\hat{\omega}\hat{\pi}_3$ for all ν and $\hat{\omega}$, $p\hat{\pi}_4\wedge fit\hat{\pi}_4\hat{\pi}_5 = true$ where
$\hat{\pi}_5 = \langle\langle \acute{\rho}_2[\acute{\upsilon}_2\downarrow 1/\text{I}], \acute{\upsilon}_2\dagger 1, \acute{\sigma}_2\rangle, \langle \grave{\rho}_2[\grave{\upsilon}_2\downarrow 1/\text{I}], \grave{\upsilon}_2\dagger 1, \grave{\sigma}_2\rangle\rangle$. In addition $k\hat{\zeta}_1\hat{\pi}_1 = true$ so $k\langle \acute{\zeta}_1\circ revert\acute{\rho}_1, \grave{\zeta}_1\circ revert\grave{\rho}_1\rangle\hat{\pi}_5 = true$ (as was proven for a similar case in 2.5.1) and we may deduce that
$a\langle \acute{\xi}_0(\acute{\zeta}_1\circ revert\acute{\rho}_1)\acute{\rho}_4\acute{\upsilon}_4\acute{\sigma}_4, \grave{\xi}_0(\grave{\zeta}_1\circ revert\grave{\rho}_1)\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4\rangle = true$
$c\langle \acute{\xi}_0(\acute{\zeta}_1\circ revert\acute{\rho}_1), \grave{\xi}_0(\grave{\zeta}_1\circ revert\grave{\rho}_1)\rangle\hat{\pi}_2 = true$.

Now take ψ and $\hat{\pi}$ to be such that
$apt\psi\hat{\rho}\wedge torn[\![\text{fnI..E}]\!]\psi = true$; when $\hat{\zeta}$ satisfies $k\hat{\zeta}\hat{\pi} = true$ and $\hat{\pi}_0$

satisfies $p\hat{\pi}_0 \wedge fit\hat{\pi}_0\hat{\pi} = true$ inevitably

$\mathscr{G}[\![\text{fn}I..E]\!]\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0$ is $\acute{\zeta}\acute{\rho}_0((\acute{\varepsilon}_0) \S \acute{\upsilon}_0)\acute{\sigma}_0$ where $\acute{\varepsilon}_0$ is $\langle sv \circ \acute{\xi}_0, rend[\![\text{fn}I.E]\!]\acute{\rho}_0 \rangle$ and $\acute{\xi}_0$ is as above. On the other hand, $\mathscr{G}[\![\mathscr{I} [\![\text{fn}I..E]\!]\psi]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0$ is ostensibly influenced by the value of $opt[\![I]\!]\psi$; as intimated in 1.4.6, however, $\mathscr{G}[\![(\text{fn}I.\mathscr{E}[\![E]\!]\psi[false/I])\$]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0$ and $\mathscr{G}[\![\text{fn}I..\mathscr{E}[\![E]\!]\psi[false/I]]\!]\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0$ are both $\grave{\zeta}_0\grave{\rho}_0((\grave{\varepsilon}_0) \S \grave{\upsilon}_0)\grave{\sigma}_0$ where $\grave{\varepsilon}_0$ is $\langle sv \circ \grave{\xi}_0, rend[\![\text{fn}I.E]\!]\grave{\rho}_0 \rangle$. The proof of 2.5.2 can be amended readily to show that when $c\hat{\zeta}_2\hat{\pi}_2 = true$ for some $\hat{\zeta}_2$ and $\hat{\pi}_2$ we have $c\langle sv\acute{\zeta}_2, sv\grave{\zeta}_2 \rangle \hat{\pi}_2 = true$, so (remembering that $o\langle \lambda\zeta.\acute{\xi}_0(\zeta \circ revert\acute{\rho}_1), \lambda\zeta.\grave{\xi}_0(\zeta \circ revert\grave{\rho}_1) \rangle \hat{\pi}_1\hat{\pi}_2 = true$ when $\hat{\rho}_2 = \langle divert\acute{\rho}_1(\acute{\varepsilon}_0 \downarrow 2), divert\grave{\rho}_1(\grave{\varepsilon}_0 \downarrow 2) \rangle$ from the paragraph above) $o\langle \lambda\zeta.sv(\acute{\xi}_0(\zeta \circ revert\acute{\rho}_1)), \lambda\zeta.sv(\grave{\xi}_0(\zeta \circ revert\grave{\rho}_1)) \rangle \hat{\pi}_1\hat{\pi}_2 = true$ under these circumstances. This being so for all $\hat{\pi}_1$, $w\hat{\varepsilon}_0\hat{\pi}_0 = true$ and $G[\![\text{fn}I..E]\!] = true$.

In like manner we can show that $G[\![\text{fn}I_1,...,I_n.E]\!] \wedge G[\![\text{fn}I_1,...,I_n..E]\!] = true$, but the necessary proofs diverge slightly from those above because, writing $I^* = \langle I_1,...,I_n \rangle$, $\mathscr{G}[\![(\text{fn}I_1,...,I_n.\mathscr{E}[\![E]\!]\psi[false^*/I^*])\$]\!]\grave{\zeta}\grave{\rho}_0$ does not coincide with $\mathscr{G}[\![\text{fn}I_1,...,I_n..\mathscr{E}[\![E]\!]\psi[false^*/I^*]]\!]\grave{\zeta}\grave{\rho}_0$ although the ultimate effects are the same. The first of these places $\langle sv \circ \grave{\xi}_1, rend[\![\text{fn}I_1,...,I_n.E]\!]\grave{\rho}_0 \rangle$ on the stack whereas the second places $\langle sv \circ \grave{\xi}_2, rend[\![\text{fn}I_1,...,I_n.E]\!]\grave{\rho}_0 \rangle$ on it; here $\grave{\xi}_1$ is

$\lambda\zeta\rho\upsilon\sigma.(\lambda\alpha^*.(\lambda\sigma'.\#\alpha^* = n \rightarrow \mathscr{L}[\![\mathscr{E}[\![E]\!]\psi[false^*/I^*]]\!]\zeta\rho[holds(\upsilon\downarrow 1)\sigma/I^*](\upsilon\dagger 1)\sigma', \tau)$

$(updates\alpha^*(holds(\upsilon\downarrow 1)\sigma)\sigma))(novels(\#\upsilon\downarrow 1 \,|\, L^*)\sigma)$

but $\grave{\xi}_2$ is

$\lambda\zeta\rho\upsilon\sigma.\#\upsilon\downarrow 1 \,|\, L^* = n \rightarrow \mathscr{L}[\![\mathscr{E}[\![E]\!]\psi[false^*/I^*]]\!]\zeta\rho[holds(\upsilon\downarrow 1)\sigma/I^*](\upsilon\dagger 1)\sigma, \tau.$

However, if we define

$\acute{\xi}_1 = \acute{\xi}_2 = \lambda\zeta\rho\upsilon\sigma.\#\upsilon\downarrow 1 \,|\, L^* = n \rightarrow \mathscr{L}[\![E]\!]\zeta\rho[holds(\upsilon\downarrow 1)\sigma/I^*](\upsilon\dagger 1)\sigma, \tau,$

$\mathscr{G}[\![\text{fn}I_1,...,I_n..E]\!]\acute{\zeta}\acute{\rho}_0$ adjoins $\langle sv \circ \acute{\xi}_1, rend[\![\text{fn}I_1,...,I_n.E]\!]\acute{\rho}_0 \rangle$

to $\acute{\upsilon}_0$, and we can verify as before that whenever $\hat{\pi}_1$ and $\hat{\pi}_2$ have $\hat{\rho}_2=\langle divert\acute{\rho}_1(rend[\![\, \mathrm{fn} I_1,\ldots,I_n.E]\!]\acute{\rho}_0), divert\grave{\rho}_1(rend[\![\, \mathrm{fn} I_1,\ldots,I_n.E]\!]\grave{\rho}_0)\rangle$, $\theta_1=\langle \acute{\upsilon}_2\dagger 1,\grave{\upsilon}_2\dagger 1\rangle$ and $fit\hat{\pi}_2\hat{\pi}_2=true$ and whenever $\hat{\zeta}_1$ has $k\hat{\zeta}_1\hat{\pi}_1=true$ $c\langle sv\acute{\xi}_n(\acute{\zeta}_1\circ revert\acute{\rho}_1), sv\grave{\xi}_n(\grave{\zeta}_1\circ revert\grave{\rho}_1)\rangle\hat{\pi}_2=true$ if n is 1 or 2 and if $torn[\![\, \mathrm{fn} I_1,\ldots,I_n.E]\!]\acute{\rho}_0=true$. Hence the result follows by the argument of the opening paragraph.⊁

Here we make significant use for the first time of the conditions on the stacks in the definition of *fit*, in that we arrange to supply an abstraction and its transform with parameters both of which are locations or both of which are stored values. The next lemma requires not only these constraints but also that of 2.4.5 to the effect that if $p\hat{\pi}=true$ then $p_0\hat{\pi}=true$.

2.5.5. Lemma.

If $E[\![E_0]\!]\wedge E[\![E_1]\!]=true$ then $G[\![E_0:=E_1]\!]=true$.

⊀As in the preceding results we can ignore the conditions in the definitions of 2.4.5 involving label entry points as the scopes of labels do not propagate beyond $E_0:=E_1$. We shall assume that *mete* evaluates expressions from left to right since the alternative leads to essentially the same proof.

Suppose that when $k\hat{\zeta}\hat{\pi}=true$ for some $\hat{\zeta}$ and $\hat{\pi}$ then $c\hat{\zeta}_0\hat{\pi}_0=true$ where

$\hat{\zeta}_0=\langle \lambda\zeta\rho\upsilon.\acute{\zeta}\rho(\langle dummy\rangle\S\upsilon\dagger 2)\circ update(\upsilon\downarrow 2)(\upsilon\downarrow 1),$

$\quad \lambda\zeta\rho\upsilon.\grave{\zeta}\rho(\langle dummy\rangle\S\upsilon\dagger 2)\circ update(\upsilon\downarrow 2)(\upsilon\downarrow 1)\rangle$,

$\hat{\rho}_0=\hat{\rho}_1=\hat{\rho}$, $\theta_0=\theta_0=\theta$, $\theta_1=\langle\langle\acute{\alpha}\rangle\S\acute{\upsilon},\langle\grave{\alpha}\rangle\S\grave{\upsilon}\rangle$ and $\theta_0=\langle\langle\acute{\beta}\rangle\S\acute{\upsilon}_1,\langle\grave{\beta}\rangle\S\grave{\upsilon}_1\rangle$ for some $\hat{\alpha}:L\times L$ and $\hat{\beta}:V\times V$ such that $set\hat{\pi}_0\hat{\pi}_1\wedge set\hat{\pi}_1\hat{\pi}=true$. Under these circumstances $k\langle sv\acute{\zeta}_0, sv\grave{\zeta}_0\rangle\hat{\pi}_1=true$ by 2.5.2 so, $E[\![E_0]\!]$ and $E[\![E_1]\!]$ being *true*, $c\langle\mathscr{R}[\![E_1]\!]\acute{\zeta}_0,\mathscr{R}[\![\bullet[\![E_1]\!]\psi]\!]\grave{\zeta}_0\rangle\hat{\pi}_1=true$, $k\langle mv(\mathscr{R}[\![E_1]\!]\acute{\zeta}_0), mv(\mathscr{R}[\![\bullet[\![E_1]\!]\psi]\!]\grave{\zeta}_0)\rangle\hat{\pi}=true$ and $c\langle\mathscr{L}[\![E_0]\!](\mathscr{R}[\![E_1]\!]\acute{\zeta}_0),\mathscr{L}[\![\bullet[\![E_0]\!]\psi]\!](\mathscr{R}[\![\bullet[\![E_1]\!]\psi]\!]\grave{\zeta}_0)\rangle\hat{\pi}=true$ when $apt\psi\hat{\rho}\wedge torn[\![E_0:=E_1]\!]\psi=true$. Thus if for any $\hat{\zeta}$ and $\hat{\pi}$ with $k\hat{\zeta}\hat{\pi}=true$

$c\hat{\zeta}_0\hat{\pi}_0 = true$ when $\hat{\zeta}_0$ and $\hat{\pi}_0$ are as above we shall know that $G[\![E_0 := E_1]\!] = true$.

Take any such $\hat{\zeta}$ and $\hat{\pi}$ together with some $\hat{\pi}_2$ having $p\hat{\pi}_2 \wedge fit\hat{\pi}_2\hat{\pi}_0 = true$ and define $\hat{\pi}_3$ to be $\langle \hat{\rho}_2, \langle dummy \rangle \S \hat{\upsilon}_2 \dagger 2, update(\hat{\upsilon}_2 \downarrow 2)(\hat{\upsilon}_2 \downarrow 1)\hat{\sigma}_2 \rangle$ (and similarly for $\grave{\pi}_3$). Patently $set\hat{\pi}_3\hat{\pi} = true$, so should $p\hat{\pi}_3$ be $true$ $a\langle \acute{\zeta}\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3, \grave{\zeta}\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3 \rangle$ will be $true$ and $a\langle \acute{\zeta}_0\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2, \grave{\zeta}_0\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2 \rangle$ will be $true$. That $p\hat{\pi}_3$ is $true$ follows from the assertion that for all $\nu<2$ and $\hat{\omega}$ $kent\nu\hat{\omega}\hat{\pi}_3 \supset (\hat{\omega} = \langle dummy, dummy \rangle \vee kent\nu\hat{\omega}\hat{\pi}_2)$, which we now ratify.

Suppose that for some ν_1 and all $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3 \wedge kent1\hat{\omega}_1\hat{\pi}_2 \supset kent\nu_0\hat{\omega}_0\hat{\pi}_2$; this is clearly the case if $\nu_1 = 0$. Let $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ be entities having $seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3 \wedge kent1\hat{\omega}_1\hat{\pi}_2 = true$. If $\acute{\omega}_1 : L$ or $\grave{\omega}_1 : L$, $seen\nu_0\nu_1\hat{\omega}_0(access\hat{\omega}_1\hat{\pi}_3)\hat{\pi}_3 = true$ so should $kent1(access\hat{\omega}_1\hat{\pi}_3)\hat{\pi}_2$ be $true$ $kent\nu_0\hat{\omega}_0\hat{\pi}_2$ will be $true$ by the induction hypothesis. When $\acute{\omega}_1 = \hat{\upsilon}_2 \downarrow 2$ or $\grave{\omega}_1 = \grave{\upsilon}_2 \downarrow 2$, since $kent1\hat{\omega}_1\hat{\pi}_2 = kent1\langle \hat{\upsilon}_2 \downarrow 2, \grave{\upsilon}_2 \downarrow 2 \rangle \hat{\pi}_2 = true$, we have $\hat{\omega}_1 = \langle \hat{\upsilon}_2 \downarrow 2, \grave{\upsilon}_2 \downarrow 2 \rangle$, $access\hat{\omega}_1\hat{\pi}_3 = \langle \hat{\upsilon}_2 \downarrow 1, \grave{\upsilon}_2 \downarrow 1 \rangle$ and $kent1\langle \hat{\upsilon}_2 \downarrow 1, \grave{\upsilon}_2 \downarrow 1 \rangle \hat{\pi}_2 = true$; otherwise $access\hat{\omega}_1\hat{\pi}_3 = access\hat{\omega}_1\hat{\pi}_2$ and $kent1(access\hat{\omega}_1\hat{\pi}_2)\hat{\pi}_2 = true$ by 2.1.8. Hence $kent\nu_0\hat{\omega}_0\hat{\pi}_2 = true$ under these circumstances; as we can employ the techniques of 2.5.3 to check that $kent\nu_0\hat{\omega}_0\hat{\pi}_2 = true$ unless $\acute{\omega}_1 : L$ or $\grave{\omega}_1 : L$, we infer that the induction hypothesis may be 'stepped up' freely from ν_1 to ν_1+1 and that for all $\nu_0<2$, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3 \wedge kent1\hat{\omega}_1\hat{\pi}_2 \supset kent\nu_0\hat{\omega}_0\hat{\pi}_2$. Now for any $\nu<2$ and $\hat{\omega}$

$$kent\nu\hat{\omega}\hat{\pi}_3 \supset \bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_3 \wedge (\hat{\omega}_1 = \langle dummy, dummy \rangle \vee yclept\hat{\omega}_1\hat{\pi}_2)\}\}$$
$$\supset (\hat{\omega}_1 = \langle dummy, dummy \rangle \vee \bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_3 \wedge kent1\hat{\omega}_1\hat{\pi}_2\}\})$$
$$\supset (\hat{\omega}_1 = \langle dummy, dummy \rangle \vee kent\nu\hat{\omega}\hat{\pi}_2).$$

In consequence $p\hat{\pi}_3 = true$ and, $\hat{\pi}_2$ being a typical pair having $p\hat{\pi}_2 \wedge fit\hat{\pi}_2\hat{\pi}_0 = true$, $c\hat{\zeta}_0\hat{\pi}_0 = true$ whatever suitable $\hat{\zeta}$ and $\hat{\pi}$ give rise to $\hat{\zeta}_0$ and $\hat{\pi}_0$. From our opening remarks we may therefore conclude that $G[\![E_0 := E_1]\!] = true$.►

2.5.6. Lemma.

If $R[\![E_0]\!] \wedge G[\![E_1]\!] \wedge G[\![E_2]\!] = true$ then $G[\![E_1;\ E_2]\!] = true$, $G[\![\textbf{if } E_0 \textbf{ then } E_1 \textbf{ else } E_2]\!] = true$ and $G[\![\textbf{while } E_0 \textbf{ do } E_1]\!] = true$.

⊲We shall establish only the third part of this result, as the others involve much the same method. Let $\hat{\zeta}$ and $\hat{\pi}$ be any pairs having $k\hat{\zeta}\hat{\pi} = true$ and define

$\xi = \mathscr{R}[\![E_0]\!] \circ (\lambda\zeta\rho\upsilon.\upsilon\downarrow 1 \to \mathscr{G}[\![E_1]\!](\lambda\rho'\upsilon'.\zeta\rho'(\upsilon'\dagger 1))\rho(\upsilon\dagger 1), \zeta\rho(\langle dummy\rangle \S \upsilon\dagger 1))$

and

$\grave{\xi} = \mathscr{R}[\![e[\![E_0]\!]\psi]\!] \circ (\lambda\zeta\rho\upsilon.\upsilon\downarrow 1 \to \mathscr{G}[\![g[\![E_1]\!]\psi]\!](\lambda\rho'\upsilon'.\zeta\rho'(\upsilon'\dagger 1))\rho(\upsilon\dagger 1),$

$\grave{\zeta}\rho(\langle dummy\rangle \S \upsilon\dagger 1))$

for some ψ having $apt\psi\hat{\rho} \wedge rent[\![E_3]\!]\psi = true$, where E_3 is $\textbf{while } E_0 \textbf{ do } E_1$.

If $\hat{\zeta}_0$ satisfies $c\hat{\zeta}_0\hat{\pi} = true$, then $c\hat{\zeta}_1\hat{\pi} = true$ where

$\hat{\zeta}_1 = \langle \mathscr{G}[\![E_1]\!](\lambda\rho\upsilon.\acute{\zeta}_0\rho(\upsilon\dagger 1)), \mathscr{G}[\![g[\![E_1]\!]\psi]\!](\lambda\rho\upsilon.\grave{\zeta}_0\rho(\upsilon\dagger 1))\rangle$, as

$k\langle \lambda\rho\upsilon.\acute{\zeta}_0\rho(\upsilon\dagger 1), \lambda\rho\upsilon.\grave{\zeta}_0\rho(\upsilon\dagger 1)\rangle \hat{\pi} \wedge G[\![E_1]\!] = true$. Moreover, writing

$\acute{\zeta}_2 = \lambda\rho\upsilon.\upsilon\downarrow 1 \to \acute{\zeta}_1\rho(\upsilon\dagger 1), \zeta\rho(\langle dummy\rangle \S \upsilon\dagger 1)$ (and similarly for $\grave{\zeta}_2$),

$k\langle s\upsilon\acute{\zeta}_2, s\upsilon\grave{\zeta}_2\rangle \hat{\pi} = true$ and, as $R[\![E_0]\!] = true$, $c\langle \mathscr{R}[\![E_0]\!]\acute{\zeta}_2, \mathscr{R}[\![e[\![E_0]\!]\psi]\!]\grave{\zeta}_2\rangle \hat{\pi} = true$.

Hence for all $\hat{\zeta}_0$ with $c\hat{\zeta}_0\hat{\pi} = true$, $c\langle \acute{\xi}\acute{\zeta}_0, \grave{\xi}\grave{\zeta}_0\rangle \hat{\pi} = true$; since c is inclusive by 2.4.7 and since $c\langle \perp, \perp\rangle \hat{\pi} = true$ we have

$c\langle \mathscr{G}[\![E_3]\!]\acute{\zeta}, \mathscr{G}[\![g[\![E_3]\!]\psi]\!]\grave{\zeta}\rangle \hat{\pi} = c\langle fix\acute{\xi}, fix\grave{\xi}\rangle \hat{\pi} = true$.

From this we can infer that in fact

$k\langle \lambda\rho\upsilon.fix\acute{\xi}\rho(\upsilon\dagger 1), \lambda\rho\upsilon.fix\grave{\xi}\rho(\upsilon\dagger 1)\rangle \hat{\pi} = true$. Now

$\mathscr{R}[\![E_3]\!]\acute{\zeta}\acute{\rho}\acute{\upsilon} \S \mathscr{Q}[\![E_3]\!]\acute{\zeta}\acute{\rho}\acute{\upsilon} = (\lambda\zeta.\mathscr{R}[\![E_1]\!]\zeta\acute{\rho}\acute{\upsilon} \S \mathscr{Q}[\![E_1]\!]\zeta\acute{\rho}\acute{\upsilon})(\lambda\rho\upsilon.fix\acute{\xi}\rho(\upsilon\dagger 1))$

and analogous remarks hold for the transformed programs; $G[\![E_1]\!]$ being $true$ and $\hat{\zeta}$ and $\hat{\pi}$ being typical of those pairs with $k\hat{\zeta}\hat{\pi} = true$, $G[\![\textbf{while } E_0 \textbf{ do } E_1]\!] = true$.⊳

As a counterpart to 2.5.1 we next outline the proof that a label set by incidence can be satisfactorily transformed into one set by reference whatever the nature of the expression labelled by it. This is in sharp contrast with the situation concerning recursive declarations, which will be analysed in 2.7.6.

2.5.7. Lemma.

If $G[\![E]\!]=true$ then $G[\![I:E]\!]\wedge G[\![I::E]\!]=true$ unless I is a member of $\mathscr{I}[\![E]\!]\S\mathscr{X}[\![E]\!]$.

⊲We shall consider only I::E, the proof for I:E being similar. Let $apt\psi\hat{\rho}\wedge torn[\![I::E]\!]\psi\wedge k\hat{\zeta}\hat{\pi}=true$ for some ψ, $\hat{\pi}$ and $\hat{\zeta}$; then $o\langle\mathscr{G}[\![I::E]\!]\hat{\zeta},\mathscr{G}[\![\mathscr{I}[\![I::E]\!]\psi]\!]\grave{\zeta}\rangle\hat{\pi}=o\langle\mathscr{G}[\![E]\!]\hat{\zeta},\mathscr{G}[\![\mathscr{I}[\![E]\!]\psi]\!]\grave{\zeta}\rangle\hat{\pi}=true$ and $w\langle(\mathscr{G}[\![E]\!]\hat{\zeta},\hat{\rho},\hat{\upsilon}),(\mathscr{G}[\![\mathscr{I}[\![E]\!]\psi]\!]\grave{\zeta},\grave{\rho},\grave{\upsilon})\rangle\hat{\pi}=true$.

Suppose that $\psi[\![I]\!]\downarrow 1=true$; writing

$I^*=\mathscr{I}[\![\mathscr{I}[\![I::E]\!]\psi]\!]\S\mathscr{X}[\![\mathscr{I}[\![I::E]\!]\psi=\langle I\rangle\S\mathscr{I}[\![\mathscr{I}[\![E]\!]\psi]\!]\S\mathscr{X}[\![\mathscr{I}[\![E]\!]\psi]\!]$ and

$\omega^*=\mathscr{P}[\![\mathscr{I}[\![I::E]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}\S\mathscr{Q}[\![\mathscr{I}[\![I::E]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}$

$=\langle(\mathscr{G}[\![\mathscr{I}[\![E]\!]\psi]\!]\grave{\zeta},\grave{\rho},\grave{\upsilon})\rangle\S\mathscr{P}[\![\mathscr{I}[\![E]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}\S\mathscr{Q}[\![\mathscr{I}[\![E]\!]\psi]\!]\grave{\zeta}\grave{\rho}\grave{\upsilon}$,

for every ν with $1\leq\nu\leq\#I^*$ we know from 1.4.6 that

$swap(\mathscr{I}[\![I::E]\!]\S\mathscr{X}[\![I::E]\!])I^*\omega^*\downarrow\nu$ must coincide with

$(\nu=\#\mathscr{I}[\![E]\!]+1\rightarrow\omega^*\downarrow 1,swap(\mathscr{I}[\![E]\!]\S\mathscr{X}[\![E]\!])(I^*\dagger 1)(\omega^*\dagger 1)\downarrow(\nu\leq\#\mathscr{I}[\![E]\!]\rightarrow\nu,\nu-1))$.

Now $w\langle(\mathscr{G}[\![E]\!]\hat{\zeta},\hat{\rho},\hat{\upsilon}),\omega^*\downarrow 1\rangle\hat{\pi}=true$ and if $1\leq\nu\leq\#I^*-1$

$w\langle\mathscr{P}[\![E]\!]\hat{\zeta}\hat{\rho}\hat{\upsilon}\S\mathscr{Q}[\![E]\!]\hat{\zeta}\hat{\rho}\hat{\upsilon}\downarrow\nu,swap(\mathscr{I}[\![E]\!]\S\mathscr{X}[\![E]\!])(I^*\dagger 1)(\omega^*\dagger 1)\downarrow\nu\rangle\hat{\pi}=true$.

Similar remarks are germane when $\psi[\![I]\!]\downarrow 1=false$ so as ψ, $\hat{\pi}$ and $\hat{\zeta}$ are any pairs having $apt\psi\hat{\rho}\wedge torn[\![I::E]\!]\psi\wedge k\hat{\zeta}\hat{\pi}=true$ we can infer that $G[\![I::E]\!]=true$.⊳

We shall leave out all except one of the results about declarations because their proofs are simplified versions of those to be given in the next section. Typical among them is the assertion that if $E[\![E]\!]=true$ then $D[\![\Delta]\!]\wedge T[\![\Delta]\!]=true$ when Δ takes the form I=E or $I_1,\ldots,I_n$=E and $D[\![\Delta]\!]=true$ when Δ takes the form I==E or $I_1,\ldots,I_n$==E; 2.6.5 will verify a somewhat weaker contention about $\mathscr{I}[\![I_1,\ldots,I_n==E]\!]$. Likewise in 2.6.6 we shall show that any Δ_0 and Δ_1 satisfying $D[\![\Delta_0]\!]\wedge D[\![\Delta_1]\!]=true$ have $D[\![\Delta_0$ within $\Delta_1]\!]=true$ whereas any Δ_0 and Δ_1 satisfying $D[\![\Delta_0]\!]\wedge T[\![\Delta_1]\!]=true$ have $T[\![\Delta_0$ within $\Delta_1]\!]=true$ (provided that $\mathscr{X}[\![\Delta_0]\!]\S\mathscr{I}[\![\Delta_1]\!]$ and $\mathscr{I}[\![\Delta_0]\!]\S\mathscr{X}[\![\Delta_0]\!]\S\mathscr{I}[\![\Delta_1]\!]$ are lists without repeated

elements). Multiple declarations will receive a similar treatment in 2.6.7 where we shall confirm that for all $\Delta_1,\ldots,\Delta_n$ $D[\![\Delta_1]\!]\wedge\ldots\wedge D[\![\Delta_n]\!]\supset D[\![\Delta_1$ and...and $\Delta_n]\!]$ and $T[\![\Delta_1]\!]\wedge\ldots\wedge T[\![\Delta_n]\!]\supset T[\![\Delta_1$ and...and $\Delta_n]\!]$.

2.5.8. Lemma.

If $T[\![\Delta]\!]=true$ then $T[\![$rec $\Delta]\!]=true$ and, when $opts(\mathscr{H}[\![\Delta]\!])=\lambda\psi.false^*$ also, $D[\![$rec $\Delta]\!]=true$.

⊀Suppose that ψ_0, $\hat{\pi}_0$, $\hat{\pi}_1$ and $\hat{\zeta}$ are such that $opts(\mathscr{H}[\![\Delta]\!])\psi_0=false^*$, $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}=true$ and $\bigwedge\{c\hat{\zeta}\hat{\pi}\mid sewn[\![$rec $\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}\}=true$; let ψ_1 be $\psi_0[false^*/\mathscr{S}[\![\Delta]\!]][false^*/\mathscr{H}[\![\Delta]\!]]$. Because $p_0\hat{\pi}_1=true$, if $\langle\acute{\alpha}^*,\grave{\alpha}^*\rangle=\langle novels(\#\mathscr{S}[\![\Delta]\!])\acute{\sigma}_1\acute{\upsilon}_1\acute{\sigma}_1,novels(\#\mathscr{S}[\![\Delta]\!])\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle$ and $\theta_2=\langle updates\acute{\alpha}^*dummy^*\acute{\sigma}_1,updates\grave{\alpha}^*dummy^*\grave{\sigma}_1\rangle$, θ_2 is proper. Define $0_2=0_1$ and

$\hat{\rho}_2=\langle fix(\lambda\rho.\acute{\sigma}_1[\acute{\alpha}^*/\mathscr{S}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho\acute{\sigma}_2/\mathscr{H}[\![\Delta]\!]]),$

$\quad fix(\lambda\rho.\grave{\sigma}_1[\grave{\alpha}^*/\mathscr{S}[\![\Delta]\!]][\mathscr{S}[\![\iota[\![\Delta]\!]\psi_1]\!]\rho\grave{\sigma}_2/\mathscr{H}[\![\Delta]\!]])\rangle$.

Plainly, if $\hat{\pi}$ is such that $sewn[\![\Delta]\!]1\psi_1\hat{\pi}_2\hat{\pi}=true$ then $sewn[\![$rec $\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}=true$; thus should $p\hat{\pi}_2$ be $true$ both $a\langle\mathscr{T}[\![\Delta]\!]\acute{\zeta}\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\mathscr{T}[\![\iota[\![\Delta]\!]\psi_1]\!]\grave{\zeta}\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle$ and $a\langle\mathscr{D}[\![$rec $\Delta]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{D}[\![\alpha[\![$rec $\Delta]\!]\psi_0]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle$ will be $true$.

By an argument akin to that of 2.5.1 for all ν and $\hat{\omega}$

$kent\nu\hat{\omega}\hat{\pi}_2\supset(\hat{\omega}=\langle dummy,dummy\rangle\vee gyven\hat{\omega}\langle\acute{\alpha}^*,\grave{\alpha}^*\rangle$

$\quad\vee gyven\hat{\omega}\langle\mathscr{S}[\![\Delta]\!]\acute{\rho}_2\acute{\sigma}_2,\mathscr{S}[\![\iota[\![\Delta]\!]\psi_1]\!]\grave{\rho}_2\grave{\sigma}_2\rangle\vee kent\nu\hat{\omega}\hat{\pi}_1)$.

Thus to show that $p\hat{\pi}_2=true$ it is enough to prove that $gyven\hat{\omega}\langle\mathscr{S}[\![\Delta]\!]\acute{\rho}_2\acute{\sigma}_2,\mathscr{S}[\![\iota[\![\Delta]\!]\psi_1]\!]\grave{\rho}_2\grave{\sigma}_2\rangle\supset\omega\hat{\omega}\hat{\pi}_2$. Accordingly let I be any member of $\mathscr{H}[\![\Delta]\!]$; we wish to verify that for all $\hat{\pi}_3$ having $fit\hat{\pi}_3\hat{\pi}_3=true$ we have

$o\langle\mathscr{T}[\![\Delta]\!]\circ(\lambda\zeta\rho\upsilon.recur\zeta\rho\langle\rho[\![I]\!]\dagger1\mid$E$\rangle)$

$\quad\mathscr{T}[\![\iota[\![\Delta]\!]\psi_1]\!]\circ(\lambda\zeta\rho\upsilon.recur\zeta\rho\langle\rho[\![I]\!]\dagger1\mid$E$\rangle))\hat{\pi}_3\hat{\pi}_4=true$

where $\hat{\pi}_4=\langle\langle(tear[\![\Delta]\!]\acute{\sigma}_2)[\hat{\pi}_3/rec],\langle\rangle,\acute{\sigma}_2\rangle,\langle(tear[\![\Delta]\!]\grave{\rho}_2)[\hat{\pi}_3/rec],\langle\rangle,\grave{\sigma}_2\rangle\rangle$.

This will be so if every $\hat{\zeta}_0$ having $k\hat{\zeta}_0\hat{\pi}_3=true$ satisfies also $\bigwedge\{c\hat{\zeta}_1\hat{\pi}\,|\,sewn[\![\Delta]\!]1\psi_1\hat{\pi}_4\hat{\pi}\}=true$ where $\hat{\zeta}_1=\langle\lambda\rho\upsilon.recur\acute{\zeta}_0\rho\langle\rho[\![I]\!]\downarrow1\,|\,E\rangle,\lambda\rho\upsilon.recur\grave{\zeta}_0\rho\langle\rho[\![I]\!]\downarrow1\,|\,E\rangle\rangle$, because $T[\![\Delta]\!]=true$. Take any $\hat{\pi}_5$ with $p\hat{\pi}_5\wedge sewn[\![\Delta]\!]1\psi_1\hat{\pi}_4\hat{\pi}_5=true$; define $\hat{\varepsilon}_6$, $\hat{\pi}_6$ and $\hat{\pi}_7$ to be $\langle\acute{\rho}_5[\![I]\!]\downarrow1,\grave{\rho}_5[\![I]\!]\downarrow1\rangle$, $\langle\acute{\rho}_5[\![rec]\!]\downarrow1,\grave{\rho}_5[\![rec]\!]\downarrow1\rangle$ and $\langle\langle\acute{\rho}_6,\langle\acute{\varepsilon}_6\rangle\S\acute{\upsilon}_6,replace\acute{\rho}_6\acute{\upsilon}_6\acute{\sigma}_6\acute{\sigma}_5\rangle,\langle\grave{\rho}_6,\langle\grave{\varepsilon}_6\rangle\S\grave{\upsilon}_6,replace\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6\grave{\sigma}_5\rangle\rangle$ respectively, in accordance with the stipulations of 2.1.4. Now $set\hat{\pi}_7\hat{\pi}_3=true$ as $\acute{\varepsilon}_6:V$ and $\grave{\varepsilon}_6:V$, so $a\langle\acute{\zeta}_1\acute{\rho}_5\acute{\upsilon}_5\acute{\sigma}_5,\grave{\zeta}_1\grave{\rho}_5\grave{\upsilon}_5\grave{\sigma}_5\rangle=\langle\acute{\zeta}_0\acute{\rho}_7\acute{\upsilon}_7\acute{\sigma}_7,\grave{\zeta}_0\grave{\rho}_7\grave{\upsilon}_7\grave{\sigma}_7\rangle=true$ if $p\hat{\pi}_7=true$. This we establish below by proving that for all $\hat{\omega}$ $kent0\hat{\omega}\hat{\pi}_7\supset kent0\hat{\omega}\hat{\pi}_5$ and $kent1\hat{\omega}\hat{\pi}_7\supset(\acute{\omega}:L\rightarrow area\acute{\omega}\acute{\sigma}_7,true)\wedge(\grave{\omega}:L\rightarrow area\grave{\omega}\grave{\sigma}_7,true)$.

⊲Suppose that for some ν_1 $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7\wedge kent1\hat{\omega}_1\hat{\pi}_6\supset kent\nu_0\hat{\omega}_0\hat{\pi}_6$ for all $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$, and let ν_0, $\hat{\omega}_0$ and $\hat{\omega}_1$ be such that $seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7\wedge kent1\hat{\omega}_1\hat{\pi}_6=true$. Unless $\acute{\omega}_1:L$ or $\grave{\omega}_1:L$ standard arguments indicate that $kent\nu_0\hat{\omega}_0\hat{\omega}_1\hat{\pi}_6=true$. Since $kent1\hat{\omega}_1\hat{\pi}_6\supset(\acute{\omega}_1:L\rightarrow plot\acute{\omega}_1\acute{\rho}_6\acute{\upsilon}_6\acute{\sigma}_6,true)\wedge(\grave{\omega}_1:L\rightarrow plot\grave{\omega}_1\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6,true)$, however, if $\acute{\omega}_1:L$ or $\grave{\omega}_1:L$ $access\hat{\omega}_1\hat{\pi}_7=access\hat{\omega}_1\hat{\pi}_6$ and $seen\nu_0\nu_1\hat{\omega}_0(access\hat{\omega}_1\hat{\pi}_7)\hat{\pi}_7\wedge kent1(access\hat{\omega}_1\hat{\pi}_7)\hat{\pi}_6=true$ so that by the induction hypothesis $kent\nu_0\hat{\omega}_0\hat{\pi}_7=true$ under these circumstances also. Thus for any $\nu_0<2$, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7\wedge kent1\hat{\omega}_1\hat{\pi}_6\supset kent\nu_0\hat{\omega}_0\hat{\pi}_6$, and for any $\nu<2$ and $\hat{\omega}$

$$\begin{aligned}kent\nu\hat{\omega}\hat{\pi}_7&\supset\bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_7\wedge(\hat{\omega}_1=\hat{\varepsilon}_6\vee yclept\hat{\omega}\hat{\pi}_6)\,|\,\nu_1:N\}\,|\,\hat{\omega}_1:W\times W\}\\&\supset\bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_7\wedge(\hat{\omega}_1=\hat{\varepsilon}_6\vee kent1\hat{\omega}\hat{\pi}_6)\,|\,\nu_1:N\}\,|\,\hat{\omega}_1:W\times W\}\\&\supset\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\varepsilon}_6\hat{\pi}_7\,|\,\nu_1:N\}\vee kent\nu\hat{\omega}\hat{\pi}_6.\end{aligned}$$

From 2.1.8 $kent0\hat{\omega}\hat{\pi}_6\supset kent0\hat{\omega}\hat{\pi}_5$ and from 2.4.5 $p_0\hat{\pi}_6=true$ so it remains to be demonstrated merely that if $\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\varepsilon}_6\hat{\pi}_7\,|\,\nu_1:N\}=true$ then $kent0\hat{\omega}\hat{\pi}_5=true$ and, when $\nu>0$, $(\acute{\omega}:L\rightarrow area\acute{\omega}\acute{\sigma}_7,true)\wedge(\grave{\omega}:L\rightarrow area\grave{\omega}\grave{\sigma}_7,true)=true$.

⊲Suppose that for some ν_1 we are given the validity of the relation

$seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_5 \supset kent\nu_0\hat{\omega}_0\hat{\pi}_6 \vee kent\nu_0\hat{\omega}_0\hat{\pi}_5$ for all $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$, and let $\nu_0<2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ be such that $seen\nu_0(\nu+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_5 = true$. Again the usual arguments show that $kent\nu_0\hat{\omega}_0\hat{\pi}_6 = true$ or $kent\nu_0\hat{\omega}_0\hat{\pi}_5 = true$ unless, perhaps, $\acute{\omega}_1:L$ or $\grave{\omega}_1:L$. If $\acute{\omega}_1:L$ and $plot\acute{\omega}_1\acute{\rho}_6\acute{\upsilon}_6\acute{\sigma}_6 = true$, there is some $\grave{\varepsilon}$ having $kent1\langle\acute{\omega}_1,\grave{\varepsilon}\rangle\hat{\pi}_6 = true$ and $kent0\langle\acute{\omega}_1,\grave{\varepsilon}\rangle\hat{\pi}_6 = true$ so that, $p_0\hat{\pi}_6$ being $true$, $\grave{\varepsilon}=\grave{\omega}_1$, $kent1\hat{\omega}_1\hat{\pi}_6 = true$ and $access\hat{\omega}_1\hat{\pi}_6 = access\hat{\omega}_1\hat{\pi}_5$. If $\grave{\omega}_1:L$ and $plot\grave{\omega}_1\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6 = true$ the same reasoning indicates that $kent1\hat{\omega}_1\hat{\pi}_6 = true$ and $access\hat{\omega}_1\hat{\pi}_6 = access\hat{\omega}_1\hat{\pi}_5$. If $\acute{\omega}_1:L$ but $plot\acute{\omega}_1\acute{\rho}_6\acute{\upsilon}_6\acute{\sigma}_6 = false$ or if $\grave{\omega}_1:L$ but $plot\grave{\omega}_1\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6 = false$, $kent1\hat{\omega}_1\hat{\pi}_5 = true$ so $(\acute{\omega}_1:L\rightarrow area\acute{\omega}_1\acute{\sigma}_5, true)\wedge(\grave{\omega}_1:L\rightarrow area\grave{\omega}_1\grave{\sigma}_5, true) = true$ and $access\hat{\omega}_1\hat{\pi}_6 = access\hat{\omega}_1\hat{\pi}_5$. Thus when $\acute{\omega}_1:L$ or $\grave{\omega}_1:L$ $seen\nu_0\nu_1\hat{\omega}_0(access\hat{\omega}_1\hat{\pi}_7)\hat{\pi}_7 = true$ and either $kent1(access\hat{\omega}_1\hat{\pi}_7)\hat{\pi}_6 = true$ or $kent1(access\hat{\omega}_1\hat{\pi}_7)\hat{\pi}_5 = true$; in the former case $kent\nu_0\hat{\omega}_0\hat{\pi}_6 = true$ by the paragraph above, whereas in the latter case $kent\nu_0\hat{\omega}_0\hat{\pi}_6 = true$ or $kent\nu_0\hat{\omega}_0\hat{\pi}_5 = true$ by the induction hypothesis.

Hence for every $\nu_0<2$, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_5 \supset kent\nu_0\hat{\omega}_0\hat{\pi}_6 \vee kent\nu_0\hat{\omega}_0\hat{\pi}_5$. In particular, as $kent1\hat{\varepsilon}_6\hat{\pi}_5 = true$, if $\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\varepsilon}_6\hat{\pi}_7 \mid \nu_1:N\} = true$ then $kent0\hat{\omega}\hat{\pi}_5 = true$ and, when $\nu>0$, $(\acute{\omega}:L\rightarrow area\acute{\omega}\acute{\sigma}_6, true)\wedge(\grave{\omega}:L\rightarrow area\grave{\omega}\grave{\sigma}_6, true) = true$.➢

Consequently for all $\hat{\omega}$ $kent0\hat{\omega}\hat{\pi}_7 \supset kent0\hat{\omega}\hat{\pi}_5$ and $kent1\hat{\omega}\hat{\pi}_7 \supset (\acute{\omega}:L\rightarrow area\acute{\omega}\acute{\sigma}_7, true)\wedge(\grave{\omega}:L\rightarrow area\grave{\omega}\grave{\sigma}_7, true)$. As $p\hat{\pi}_5 = true$, for all $\hat{\omega}$ $kent0\hat{\omega}\hat{\pi}_7 \supset w\hat{\omega}\hat{\pi}_1$ and $p\hat{\pi}_7 = true$, thereby confirming that $a\langle\acute{\zeta}_0\acute{\rho}_7\acute{\upsilon}_7\acute{\sigma}_7, \grave{\zeta}_0\grave{\rho}_7\grave{\upsilon}_7\grave{\sigma}_7\rangle = true$.➢

For any $\hat{\pi}_5$ having $p\hat{\pi}_5 \wedge sewn[\![\Delta]\!]1\psi_1\hat{\pi}_4\hat{\pi}_5 = true$ we therefore have $a\langle\acute{\xi}_1\acute{\rho}_5\acute{\upsilon}_5\acute{\sigma}_5, \grave{\zeta}_1\grave{\rho}_5\grave{\upsilon}_5\grave{\sigma}_5\rangle = true$, and for any $\hat{\pi}_2$ and $\hat{\zeta}_0$ with $k\hat{\zeta}_0\hat{\pi}_3 = true$ we have

$$c\langle\mathscr{T}[\![\Delta]\!](\lambda\rho\upsilon.recur\acute{\zeta}_0\rho\langle\rho[\![I]\!]+1 \mid E\rangle)$$
$$\mathscr{T}[\![\acute{\iota}[\![\Delta]\!]\psi_1]\!](\lambda\rho\upsilon.recur\grave{\zeta}_0\rho\langle\rho[\![I]\!]+1 \mid E\rangle))\hat{\pi}_4 = true$$

where $\hat{\pi}_4 = \langle\langle(tear[\![\Delta]\!]\acute{\rho}_2)[\hat{\pi}_3/rec],\langle\rangle,\acute{\sigma}_2\rangle, \langle(tear[\![\Delta]\!]\grave{\rho}_2)[\hat{\pi}_3/rec],\langle\rangle,\grave{\sigma}_2\rangle\rangle$.

Now $gyven\hat{\omega}\langle\mathscr{S}[\![\Delta]\!]\delta_2\sigma_2,\mathscr{T}[\![\Delta]\!]\psi_1]\!]\delta_2\vartheta_2\rangle \supset w\hat{\omega}\hat{\pi}_2$, $p\hat{\pi}_2 = true$ and $a\langle\mathscr{D}[\![\text{rec } \Delta]\!]\zeta\delta_1\upsilon_1\sigma_1,\mathscr{D}[\![\mathscr{A}[\![\text{rec } \Delta]\!]\psi_0]\!]\zeta\delta_1\upsilon_1\vartheta_1\rangle = true$. This is so for any ψ_0, $\hat{\pi}_0$, $\hat{\pi}_1$ and $\hat{\zeta}$ having $opts(\mathscr{K}[\![\Delta]\!])\psi_0 = false^*$, $p\hat{\pi}_1 \wedge fit\hat{\pi}_1\hat{\pi}_0 = true$ and $\bigwedge\{c\hat{\zeta}\hat{\pi} \mid sewn[\![\text{rec } \Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}\} = true$, and accordingly $D[\![\text{rec } \Delta]\!] = true$.

That $T[\![\text{rec } \Delta]\!] = true$ is an immediate consequence of the definitions of 2.4.5.⊳

2.5.9. Theorem.

The meanings accorded by *novel* store semantics to a Mal program and its transform under the rules of 1.4.6 are equivalent, provided that the lattice of locations is infinite and every recursive declaration rec Δ_2 embedded in the program is such that any constituent of the form Δ_0 within Δ_1 gives rise to lists $\mathscr{K}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_1]\!]$ and $\mathscr{K}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_1]\!]$ without repeated members while $opts(\mathscr{K}[\![\Delta_2]\!]) = \lambda\psi.false^*$.

⊲If $opts(\mathscr{K}[\![\Delta]\!]) = \lambda\psi.false^*$ for every constituent of the form rec Δ it is possible to apply 2.6.5, which requires that no exit of a recursive declaration be changed by transformations using ψ (so that I==E, for instance, becomes I==$\mathscr{E}[\![E]\!]\psi$ and not I=$\mathscr{E}[\![E]\!]\psi$). The condition on within declarations will be explicated in 2.6.6.

Note that at no point in the preceding proofs do we demand that the *novel* function used by $\hat{\pi}$ be the one used by $\hat{\pi}$: all that we need is that they both obey the postulate of 2.1.1. Moreover if we let *opt* be $\lambda\psi I.false$, so that we consider one program evaluated with the aid of two *novel* functions, $\hat{\pi}$ and $\hat{\pi}$ request fresh locations simultaneously. Thus we can delete the requirement that L be infinite and also the requirement imposed on within declarations (as the proof of 2.6.6 will indicate) to obtain the further result that distinct *novel* functions give a program meanings which are equivalent in the sense suggested by the predicate *a* set up in 2.4.5.⊳

2.6. Connections between storage management techniques.

2.6.1. Additional invariants of computations.

We shall complete the link between standard and store semantics by proving that the store equations alluded to in 2.5.9, which invoke *novel*, frequently coincide in effect with those of 2.3.9, which invoke *new* and approach recursion somewhat differently. The proof closely resembles the one we have just considered, and we need only analyse the cases not dealt with above. Accordingly we shall concern ourselves principally with declarations and shall provide results which after slight amendment will fill the gaps in the foregoing theorem.

Throughout this section we shall presume that the state vector pair $\hat{\pi}$ has arisen through evaluating a program using *novel* store semantics to yield $\hat{\pi}$ and evaluating a transform under some ψ of the program using *new* store semantics to yield $\check{\pi}$. We retain ψ simply so that our results may be placed in the context of the theorem above without much ado. Because storage is no longer allocated to $\check{\pi}$ by means of *novel* we cannot ensure that fresh locations will be inaccessible using *site*, and thus cannot debar a location from being paired with distinct expressed values. However if the locations accessible using *plot* are necessarily in the current area of store we can demand that no fresh location $\check{\alpha}$ satisfy $kent1\langle\check{\epsilon},\check{\alpha}\rangle\hat{\pi}=false$ for all $\check{\epsilon}$. To guarantee the existence of a steady supply of locations we restrict the area of $\check{\sigma}$ to be finite. The constraints imposed on $\hat{\pi}$ remain as they were, since nothing has been changed in its mode of construction. Consequently we now set

$$p_0=\lambda\hat{\pi}.neat\hat{\rho}\wedge\#\hat{\upsilon}=\#\hat{\sigma}\wedge\#\hat{\sigma}\downarrow 2=\#\check{\sigma}\downarrow 2\wedge\#\hat{\sigma}\downarrow 3=\#\check{\sigma}\downarrow 3$$
$$\wedge\vee\{\wedge\{site\check{\alpha}_m\hat{\rho}\hat{\upsilon}\hat{\sigma}\wedge area\check{\alpha}_m\check{\sigma}\mid 1\le m\le n\}\rightarrow\vee\{\hat{\alpha}_m=\hat{\alpha}_l\mid 1\le m<l\le n\},false\mid 2\le n\}$$
$$\wedge\wedge\{(\hat{\omega}_0:L\wedge\hat{\omega}_0=\hat{\omega}_1\wedge\nu=0\rightarrow area\hat{\omega}_0\hat{\sigma}\wedge\check{\omega}_0=\check{\omega}_1)$$
$$\wedge(\check{\omega}_0:L\wedge\check{\omega}_0=\check{\omega}_1\wedge\nu=1\rightarrow area\check{\omega}_0\check{\sigma}\wedge\hat{\omega}_0=\hat{\omega}_1)\mid kent\nu\hat{\omega}_0\hat{\pi}\wedge kent1\hat{\omega}\hat{\pi}\}.$$

For a technical reason which will not emerge until 2.6.5 we now bring into play a property of expressions which hitherto has been irrelevant. The only locations which a program can assign to are those which can be handled by passing through the environment and the store. Though these need not be denoted there must be chains of values leading to them which do not require the presence of either the current stack or one forming part of a label entry point. Accordingly unless a location is accessible in this way or can be supplied by *mv* its content must be left unchanged by the evaluation of an expression. To permit the process of 2.4.6 to be carried out as before in fact we demand that only a projection of this content be preserved, writing

$$
\begin{aligned}
fit = \lambda \hat{\pi}_0 \hat{\pi}_1 \cdot\ & p_1 \hat{\pi}_1 \\
& \wedge \bigwedge \{ \hat{\varepsilon} : L \times L \vee \hat{\varepsilon} : V \times V \vee \sim gyven \hat{\varepsilon} \hat{\sigma}_1 \mid \hat{\varepsilon} : E \times E \} \\
& \wedge (\mathfrak{U} q_0 \times \mathfrak{U} q_0) \hat{\rho}_0 = (\mathfrak{U} q_0 \times \mathfrak{U} q_0) \hat{\rho}_1 \wedge (\mathfrak{V} q_0 \times \mathfrak{V} q_0) \hat{\upsilon}_0 = (\mathfrak{V} q_0 \times \mathfrak{V} q_0) \hat{\upsilon}_1 \\
& \wedge \bigwedge \{ spot \alpha \hat{\rho}_1 \hat{\upsilon}_1 \hat{\sigma}_1 \vee (q_0 (hold \alpha \hat{\sigma}_1) = q_0 (hold \alpha \hat{\sigma}_0) \wedge \sim spot \alpha \hat{\rho}_0 \hat{\upsilon}_0 \hat{\sigma}_0 \\
& \quad \vee \sim gyven \langle \alpha, \alpha \rangle \langle \hat{\upsilon}_1, \hat{\upsilon}_1 \rangle \mid \alpha : L \}.
\end{aligned}
$$

Owing to the adoption of *new* rather than *novel* by the semantic equations our knowledge of the contents of locations is circumscribed by *plot*, not *site*. The values returned by expressions are subject to

$$
\begin{aligned}
set = \lambda \hat{\pi}_0 \hat{\pi}_1 \cdot ((& \hat{\upsilon}_0 \downarrow 1 : L \wedge spot (\hat{\upsilon}_0 \downarrow 1) \hat{\rho}_0 \hat{\upsilon}_0 \hat{\sigma}_0) \\
& \vee (\hat{\upsilon}_0 \downarrow 1 : L \rightarrow \sim plot (\hat{\upsilon}_0 \downarrow 1) \hat{\rho}_0 (\langle hold (\hat{\upsilon}_0 \downarrow 1) \hat{\sigma}_0 \rangle \S \hat{\upsilon}_0 \dagger 1) \hat{\sigma}_0, true)) \\
& \wedge fit (\langle \hat{\rho}_0, \hat{\upsilon}_0 \dagger 1, \hat{\sigma}_0 \rangle, \langle \hat{\rho}_0, \hat{\upsilon}_0 \dagger 1, \hat{\sigma}_0 \rangle) \hat{\pi}_1 .
\end{aligned}
$$

Now that $fit \hat{\pi}_0 \hat{\pi}_1$ expresses a relation between $\hat{\sigma}_0$ and $\hat{\sigma}_1$ must give a more closely confined definition of $w \hat{\omega} \hat{\pi}$ when $\hat{\omega} : F \times F$, as the stores concerned must coincide. In addition if the locations on the stack which cannot be reached using *spot* are not to be assigned to during a call of a function they must not be passed as parameters; this we ensure by using *set* to insist that

the topmost element of the stack does not fall into this category. We continue to demand that a location never be paired with a stored value on the stack.

Stronger conditions must be imposed on $w\hat{\omega}\hat{\pi}$ when $\hat{\omega}:G\times G$ also, as we wish to compare two different approaches to recursion. We take w_0 to be constructed from our present p_0 just as it was built up in 2.4.5 from the earlier version, except that if we have $\hat{\omega}:G\times G$ $w_0\hat{\omega}\hat{\pi}$ is

$p_0\langle\langle\hat{\omega}{\downarrow}2,\langle\rangle,\hat{\omega}{\downarrow}3\rangle,\langle\hat{\omega}{\downarrow}2,\langle\rangle,\hat{\omega}{\downarrow}3\rangle\rangle\wedge hoten((q_0\times q_0)\hat{\omega})\langle q_0\hat{\omega}{\downarrow}2,q_0\hat{\omega}{\downarrow}2\rangle$;

this enables us to demand that

$w=\lambda\hat{\omega}\hat{\pi}.w_0\hat{\omega}\hat{\pi}$

$\wedge(\hat{\omega}:J\times J\to\bigwedge\{o\langle\hat{\omega}{\downarrow}1,\hat{\omega}{\downarrow}1\rangle\hat{\pi}_0\,|\,\hat{\rho}_0=\langle\hat{\omega}{\downarrow}2,\hat{\omega}{\downarrow}2\rangle\wedge\hat{\upsilon}_0=\langle\hat{\omega}{\downarrow}3,\hat{\omega}{\downarrow}3\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\}$,

$\hat{\omega}:F\times F\to\bigwedge\{o\langle\lambda\zeta.(\hat{\omega}{\downarrow}1)(\zeta\circ revert\hat{\rho}_0),\lambda\zeta.(\hat{\omega}{\downarrow}1)(\zeta\circ revert\hat{\rho}_0)\rangle\hat{\pi}_0\hat{\pi}_1$

$|\,\hat{\rho}_1=\langle divert\hat{\rho}_0(\hat{\omega}{\downarrow}2),divert\hat{\rho}_0(\hat{\omega}{\downarrow}2)\rangle$

$\wedge\hat{\sigma}_1=\hat{\sigma}_0\wedge fit\hat{\pi}_1\hat{\pi}_1$

$\wedge set\hat{\pi}_1\langle\langle\hat{\rho}_1,\hat{\upsilon}_0,\hat{\sigma}_0\rangle,\langle\hat{\rho}_1,\hat{\upsilon}_0,\hat{\sigma}_0\rangle\rangle\}$,

$\hat{\omega}:G\times G\to\bigwedge\{o\langle\hat{\omega}{\downarrow}1,\hat{\omega}{\downarrow}1\rangle\hat{\pi}_0\hat{\pi}_1$

$|\,\hat{\rho}_1=\langle(\hat{\omega}{\downarrow}2)[\hat{\pi}_0/\mathrm{rec}],(\hat{\omega}{\downarrow}2)[\hat{\pi}_0/\mathrm{rec}]\rangle$

$\wedge\hat{\upsilon}_0=\langle\langle\rangle,\langle\rangle\rangle\wedge fit\hat{\pi}_1\hat{\pi}_1$

$\wedge hoten((q_0\times q_0)\hat{\omega})((q_0\times q_0)\hat{\rho}_0)\}$,

$\hat{\omega}:J\times J\to\bigwedge\{k\langle\hat{\omega}{\downarrow}1,\hat{\omega}{\downarrow}1\rangle\hat{\pi}_0\,|\,\hat{\rho}_0=\langle\hat{\omega}{\downarrow}2,\hat{\omega}{\downarrow}2\rangle\wedge\hat{\upsilon}_0=\langle\hat{\omega}{\downarrow}3,\hat{\omega}{\downarrow}3\rangle\wedge fit\hat{\pi}_0\hat{\pi}_0\}$,

$true$).

Here c, k and o are presumed to satisfy the equations of 2.4.1 when we use in them the set defined above and a predicate p given again by $p=\lambda\hat{\omega}\hat{\pi}.p_0\hat{\pi}\wedge\bigwedge\{w\hat{\omega}\hat{\pi}\,|\,kent0\hat{\omega}\hat{\pi}\}$.

Thus we must set up w_n, p_{n+1}, c_{n+1}, k_{n+1} and o_{n+1} and obtain w, p, c, k and o as infinite conjunctions which are subject to an analogue of 2.4.8. When $\hat{\omega}:G\times G$, for instance, we take $w_{n+1}\hat{\omega}\hat{\pi}$ to be $true$ if $w_0\hat{\omega}\hat{\pi}$ and $o_{n+1}\langle\hat{\omega}{\downarrow}1,\hat{\omega}{\downarrow}1\rangle\hat{\pi}_0\hat{\pi}_1$ are $true$ whenever $\hat{\rho}_1=\langle(\hat{\omega}{\downarrow}2)[\hat{\pi}_0/\mathrm{rec}],(\hat{\omega}{\downarrow}2)[\hat{\pi}_0/\mathrm{rec}]\rangle$, $\hat{\upsilon}_1=\langle\langle\rangle,\langle\rangle\rangle$, $fit\hat{\pi}_1\hat{\pi}_1=true$ and

$hoten((q_0 \times q_0)\hat{\omega})((\mathfrak{A}q_0 \times \mathfrak{A}q_0)\hat{\rho}_1) = true$ (the last condition being used in 2.6.8 solely to ensure that $plot\alpha(\tilde{\omega}\downarrow 2)\langle\rangle\, empty \supset plot\alpha\tilde{\rho}_1\langle\rangle\, empty$ for all $\alpha:\mathsf{L}$). We shall not provide the proof of this analogue, as it is almost identical to the one given above.

We shall go even further than this by adopting predicates *E*, *L*, *R* and *G* having different content to those of 2.4.5 but precisely the same form. Now, however, they do not exhaust the properties of expressions in which we are interested.

Unlike our earlier theorems, which held for all Mal programs, we are here seeking a connection between two kinds of semantics which do not coincide in their effects on all possible programs, as is shown by 2.1.3. We must therefore deny to recursive declarations the right to return values whence could be reached locations not in scope on entry to the declaration. In practice this means that every exit from an expression E in such a constituent of a recursive declaration as I==E must be a global identifier denoting a member of V, a constant B or an abstraction Φ without local identifiers among its free variables; in the notation of 1.5.3 there will then be some $\psi:\mathsf{Ide}\rightarrow\mathsf{B}^*$ (not to be confused with that below) such that $cramped[\![\mathsf{E}]\!]\psi = true$. To remove the local identifiers from the environment we define $tie = \lambda\psi\rho.\langle(\lambda\mathsf{I}.\#\rho[\![\mathsf{I}]\!] > 0 \rightarrow \langle 0 \leq \psi[\![\mathsf{I}]\!]\downarrow 1 \leq 2 \rightarrow \langle \rho[\![\mathsf{I}]\!]\downarrow 1\rangle, \langle\rangle\rangle, \langle\rangle\rangle, \langle\rangle, \langle\rangle\rangle$. The limitations on the locations accessible from a value are provided by the predicate $field:\mathsf{U}\rightarrow\mathsf{Y}\rightarrow\mathsf{S}\rightarrow\mathsf{T}$, which is

$field = \lambda\rho\upsilon\sigma.\bigwedge\{plot\alpha\rho\langle\rangle\, empty \vee \sim plot\alpha arid\langle \upsilon\downarrow 1:\mathsf{L} \rightarrow hold(\upsilon\downarrow 1)\sigma, \upsilon\downarrow 1\rangle\, empty \mid \alpha:\mathsf{L}\}$

Note that $field = \lambda\rho\upsilon\sigma.field(\mathfrak{A}q_0\rho)(\mathfrak{B}q_0\upsilon)(\mathfrak{C}q_0\sigma)$ by 2.4.3.

Although the predicate underlying $\mathscr{D}$ can remain that used before, that for $\mathscr{T}$ must reflect the fact that locations will not be passed out of scope by a recursive declaration. Accordingly

$D=\lambda\Delta.\wedge\{c\langle\mathscr{G}[\![\Delta]\!]\zeta',\mathscr{D}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\zeta\rangle\,\pi_0\,|\wedge\{c\zeta\pi_1\,|\,sewn[\![\Delta]\!]0\psi\pi_0\pi_1\}\};$

$T=\lambda\Delta.\wedge\{c\langle\mathscr{T}[\![\Delta]\!]\zeta',\mathscr{T}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\zeta\rangle\,\pi_0$

$\quad|\wedge\{c\zeta\pi_1\,|\,sewn[\![\Delta]\!]1\psi\pi_0\pi_1$

$\quad\wedge\wedge\{field(tear[\![\Delta]\!]\rho_0)\langle\rho_1[\![I]\!]\downarrow1\rangle\,\delta_0\,|\,I:\mathscr{H}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\}\}\}.$

Hence $sewn[\![\Delta]\!]\nu\psi\pi_0\pi_1$ is defined exactly as in 2.4.5 except for an additional condition to the effect that *true* equals

$\wedge\{spot\alpha\rho_0\upsilon_0\delta_0\vee(q_0(hold\alpha\delta_0)=q_0(hold\alpha\delta_1)\wedge\sim spot\alpha\rho_0\upsilon_0\delta_0)$

$\quad\vee\sim gyven\langle\alpha,\alpha\rangle\langle\upsilon_0,\upsilon_0\rangle\,|\alpha:L\}.$

Our initial result will be proven at breakneck speed since it is largely a preview of 3.3.9. Strictly speaking, we should incorporate each of its paragraphs in the corresponding later lemma, but tidiness decrees otherwise. This means that the reference to 'constituents' in its statement is a little vague, but clarification will come in the course of the proof.

2.6.2. Proposition.

Let ψ and π be such that for all I:Ide $\psi[\![I]\!]\downarrow1=1$ only if $\rho[\![I]\!]\downarrow1:V$ and $\psi[\![I]\!]\downarrow1=3$ if $I:\mathscr{J}[\![E]\!]\S\mathscr{K}[\![E]\!]$. Suppose that ψ_0 and E:Exp have $cramped[\![E]\!]\psi=true$, $apt\psi_0\rho=true$, $torn[\![E]\!]\psi_0=true$ and $a\langle\zeta\rho_0\upsilon_0\sigma_0,\zeta\rho_0\upsilon_0\sigma_0\rangle=true$ whenever $p\pi_0\wedge set\pi_0\pi=true$ and $field(rend[\![E]\!](tie\psi\rho_0))\upsilon_0\delta_0\wedge(\upsilon_0\downarrow1:L\rightarrow\sim plot(\upsilon_0\downarrow1)\rho_0\upsilon_0\delta_0,true)=true$, and that any constituents E_0 and Δ_0 of E satisfy $G[\![E_0]\!]=true$ and $D[\![\Delta_0]\!]=true$. Then $c\langle\mathscr{G}[\![E]\!]\zeta',\mathscr{G}[\![\mathscr{g}[\![E]\!]\psi_0]\!]\zeta\rangle\,\pi=true$ and $\omega\langle\bar{\omega}^*\downarrow\nu,\tilde{\omega}^*\downarrow\nu\rangle\,\pi=true$ for $1\leq\nu\leq\#\mathscr{J}[\![E]\!]\S\mathscr{K}[\![E]\!]$, where $\bar{\omega}^*=\mathscr{P}[\![E]\!]\zeta\rho\upsilon\S\mathscr{Q}[\![E]\!]\zeta\rho\upsilon$ and $\tilde{\omega}^*=swap(\mathscr{J}[\![E]\!]\S\mathscr{K}[\![E]\!])(\mathscr{J}[\![\mathscr{g}[\![E]\!]\psi_0]\!]\S\mathscr{K}[\![\mathscr{g}[\![E]\!]\psi_0]\!])(\mathscr{P}[\![\mathscr{g}[\![E]\!]\psi_0]\!]\zeta\rho\upsilon\S\mathscr{Q}[\![\mathscr{g}[\![E]\!]\psi_0]\!]\zeta\rho\upsilon)$.

⊲Throughout the proof we shall fix attention on one family of ψ, ψ_0, π, ζ and E such that $cramped[\![E]\!]\psi=true$, $apt\psi_0\rho=true$, $torn[\![E]\!]\psi_0=true$ and $a\langle\zeta\rho_0\upsilon_0\sigma_0,\zeta\rho_0\upsilon_0\sigma_0\rangle=true$ when $p\pi_0\wedge set\pi_0\pi=true$ and $field(rend[\![E]\!](tie\psi\rho_0))\upsilon_0\delta_0\wedge(\upsilon_0\downarrow1:L\rightarrow\sim plot(\upsilon_0\downarrow1)\rho_0\upsilon_0\delta_0,true)=true$. We shall also presume that $\psi[\![I]\!]\downarrow1=1$ only if $\rho[\![I]\!]\downarrow1:V$ and that $\psi[\![I]\!]\downarrow1=3$ if $I:\mathscr{J}[\![E]\!]\S\mathscr{K}[\![E]\!]$. The proof will proceed by induction on

the size of E.

Suppose that E is an identifier, I; for any $\hat{\pi}_0$ having $p\hat{\pi}_0 \wedge fit\hat{\pi}_0\hat{\pi} = true$, define $\hat{\delta} = \langle \acute{\rho} \llbracket I \rrbracket \downarrow 1, \grave{\rho}_0 \llbracket I \rrbracket \downarrow 1 \rangle$, $\grave{\beta} = (\grave{\delta} : L \rightarrow (area\grave{\delta}\grave{\sigma}_0 \rightarrow hold\grave{\delta}\grave{\sigma}_0, \tau), \grave{\delta})$, $\grave{\alpha} = new\omega\grave{\sigma}_0$, $\hat{\pi}_1 = \langle \acute{\rho}_0, \langle \acute{\delta} \rangle \S \acute{\upsilon}_0, \acute{\sigma}_0 \rangle$ and $\hat{\pi}_1 = (\psi_0 \llbracket I \rrbracket \downarrow 1 = false \rightarrow \langle \grave{\rho}_0, \langle \grave{\delta} \rangle \S \grave{\upsilon}_0, \grave{\sigma}_0 \rangle, \langle \grave{\rho}_0, \langle \grave{\alpha} \rangle \S \grave{\upsilon}_0, update\grave{\alpha}\grave{\beta}\grave{\sigma}_0 \rangle)$. Then, just as in 2.5.3, $p\hat{\pi}_1 \wedge set\hat{\pi}_1\hat{\pi} = true$ (where the function *set* has meaning of 2.6.1); also I cannot be in $\mathscr{I} \llbracket E \rrbracket \S \mathscr{X} \llbracket E \rrbracket$ so $field(rend \llbracket E \rrbracket (tie\psi\grave{\rho}_1))\grave{\upsilon}_1\grave{\sigma}_1 = true$. Hence $a\langle \mathscr{G} \llbracket I \rrbracket \acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, \mathscr{G} \llbracket g \llbracket I \rrbracket \psi_0 \rrbracket \grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0 \rangle = a\langle \acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, \grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1 \rangle = true$ and $c\langle \mathscr{G} \llbracket I \rrbracket \acute{\zeta}, \mathscr{G} \llbracket g \llbracket I \rrbracket \psi_0 \rrbracket \grave{\zeta} \rangle \hat{\pi} = true$.

The situation when E is a constant, B, is very similar and need not be considered. Indeed, we shall leave out all the other possible situations except a few key ones.

Suppose that E is an abstraction, Φ; for any $\hat{\pi}_0$ with $p\hat{\pi}_0 \wedge fit\hat{\pi}_0\hat{\pi} = true$ define $\hat{\rho}_1 = \hat{\rho}_0$, $\hat{\upsilon}_1 = \langle \langle \mathscr{F} \llbracket \Phi \rrbracket \acute{\rho}_0 \rangle \S \acute{\upsilon}_0, \langle \mathscr{F} \llbracket g \llbracket \Phi \rrbracket \psi_0 \rrbracket \grave{\rho}_0 \rangle \S \grave{\upsilon}_0 \rangle$ and $\hat{\sigma}_1 = \hat{\sigma}_0$. By 2.5.4 $p\hat{\pi}_1 \wedge set\hat{\pi}_1\hat{\pi} = true$; furthermore for any I having $free \llbracket I \rrbracket \llbracket \Phi \rrbracket = true$ I is not in $\mathscr{I} \llbracket E \rrbracket \S \mathscr{X} \llbracket E \rrbracket$ and so $field(rend \llbracket E \rrbracket (tie\psi\grave{\rho}_1))\grave{\upsilon}_1\grave{\sigma}_1 = true$. Hence the $\hat{\zeta}$ above has $a\langle \mathscr{G} \llbracket \Phi \rrbracket \acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, \mathscr{G} \llbracket g \llbracket \Phi \rrbracket \psi_0 \rrbracket \grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0 \rangle = a\langle \acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, \grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1 \rangle = true$ and $c\langle \mathscr{G} \llbracket \Phi \rrbracket \acute{\zeta}, \mathscr{G} \llbracket g \llbracket \Phi \rrbracket \psi_0 \rrbracket \grave{\zeta} \rangle \hat{\pi} = true$.

Suppose that E is of the form $E_0 := E_1$ and that evaluation takes place from left to right. Defining

$\hat{\zeta}_0 = \langle \lambda\rho\upsilon.\acute{\zeta}\rho((\langle dummy \rangle \S \upsilon \dagger 2) \circ update(\upsilon \downarrow 1)(\upsilon \downarrow 2)),$

$\lambda\rho\upsilon.\grave{\zeta}\rho((\langle dummy \rangle \S \upsilon \dagger 2) \circ update(\upsilon \downarrow 1)(\upsilon \downarrow 2)) \rangle$,

as in 2.5.5 it suffices to show that for suitable $\hat{\alpha}$ and $\hat{\beta}$ we have $c\hat{\zeta}_0\langle \langle \acute{\rho}, \langle \acute{\beta}, \acute{\alpha} \rangle \S \acute{\upsilon}, \acute{\sigma} \rangle, \langle \grave{\rho}, \langle \grave{\beta}, \grave{\alpha} \rangle \S \grave{\upsilon}, \grave{\sigma} \rangle \rangle = true$. Plainly for all $\hat{\pi}_2$ having $p\hat{\pi}_2 \wedge fit\hat{\pi}_2\langle \langle \acute{\rho}, \langle \acute{\beta}, \acute{\alpha} \rangle \S \acute{\upsilon}, \acute{\sigma} \rangle, \langle \grave{\rho}, \langle \grave{\beta}, \grave{\alpha} \rangle \S \grave{\upsilon}, \grave{\sigma} \rangle \rangle = true$ we may assume that $\vee\{plot\alpha arid\langle dummy \rangle empty \mid \alpha : L\} = false$, so it is enough to demonstrate that $p\hat{\pi}_3 \wedge set\hat{\pi}_2\hat{\pi} = true$ where $\hat{\pi}_3$ is $\langle \acute{\rho}_2, \langle dummy \rangle \S \acute{\upsilon}_2 \dagger 2, update(\acute{\upsilon}_2 \downarrow 1)(\acute{\upsilon}_2 \downarrow 2)\acute{\sigma}_2 \rangle$ (and similarly for $\grave{\pi}_3$). Observe, however, that the proof in 2.5.5 that for all ν_0, ν_1,

$\hat{\omega}_0$ and $\hat{\omega}_1$ $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3 \wedge kent1\hat{\omega}_1\hat{\pi}_3 \supset kent\nu_0\hat{\omega}_0\hat{\pi}_2$ requires only that $kent1\hat{\omega}_1\hat{\pi}_3 \wedge (\vec{\omega}_1 = \upsilon_2\downarrow 2 \vee \grave{\omega}_1 = \grave{\upsilon}_2\downarrow 2) \supset (\hat{\omega}_1 = \langle \upsilon_2\downarrow 2, \grave{\upsilon}_2\downarrow 2\rangle)$, not that $kent0\hat{\omega}_1\hat{\pi}_3 \wedge (\vec{\omega}_1 = \upsilon_2\downarrow 2 \vee \grave{\omega}_1 = \grave{\upsilon}_2\downarrow 2) \supset (\hat{\omega}_1 = \langle \upsilon_2\downarrow 2, \grave{\upsilon}_2\downarrow 2\rangle)$, and can therefore be carried across unchanged to the present case, where $p_0\hat{\pi}_2$ is defined as in 2.6.1. Hence $p\hat{\pi}_3 \wedge set\hat{\pi}_3\hat{\pi} = true$, $a\langle \vec{\zeta}_0\vec{\rho}_2\upsilon_2\vec{\sigma}_2, \grave{\zeta}_0\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle = a\langle \vec{\zeta}\vec{\rho}_3\upsilon_3\vec{\sigma}_3, \grave{\zeta}\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3\rangle = true$ and $c\hat{\zeta}_0\hat{\pi} = true$.

Suppose that E has the form Δ_0 inside E_0, so that if $\psi_3 = \psi[3^*/\mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]\S\mathscr{I}[\![E_0]\!]\S\mathscr{K}[\![E_0]\!]]$ we have $cramped[\![E_0]\!]\psi_3 \wedge D[\![\Delta_0]\!] \wedge E[\![E_0]\!] = true$. It follows from the paragraphs below that if $\hat{\zeta}_2$ and $\hat{\pi}_2$ are pairs such that $sewn[\![\Delta]\!]0\hat{\psi}_0\hat{\pi}\hat{\pi}_2 = true$ and $a\langle \vec{\zeta}_2\vec{\rho}_0\upsilon_0\vec{\sigma}_0, \grave{\zeta}_2\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle = true$ for all $\hat{\pi}_0$ having $p\hat{\pi}_0 \wedge set\hat{\pi}_0\hat{\pi}_2 = true$ and

$field(rend[\![E]\!](tie\psi_3\grave{\rho}_0))\grave{\upsilon}_0\grave{\sigma}_0 \wedge (\grave{\upsilon}_0\downarrow 1 : L \rightarrow \sim plot(\grave{\upsilon}_0\downarrow 1)\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0, true) = true$

then, writing $\psi_4 = \psi_0[false^*/\mathscr{I}[\![\Delta]\!]][opts(\mathscr{K}[\![\Delta]\!])\psi_0/\mathscr{K}[\![\Delta]\!]]$, we have $c\langle \mathscr{C}[\![E_0]\!]\vec{\zeta}_2, \mathscr{E}[\![e[\![E_0]\!]\psi_4]\!]\grave{\zeta}_2\rangle\hat{\pi}_2 = true$. However if for any such $\hat{\pi}_0$ we define $\hat{\alpha} = \langle novel\vec{\rho}_0\upsilon_0\vec{\sigma}_0, new\grave{\sigma}_0\rangle$ and

$\hat{\pi}_1 = \langle\langle revert\vec{\rho}\vec{\rho}_0, (\upsilon_0\downarrow 1 : L \rightarrow \upsilon_0, \langle\hat{\alpha}\rangle \S \upsilon_0\dagger 1), (\upsilon_0\downarrow 1 : L \rightarrow \vec{\sigma}_0, update\hat{\alpha}(\upsilon_0\downarrow 1)\vec{\sigma}_0)\rangle,$

$\langle revert\grave{\rho}\grave{\rho}_0, (\grave{\upsilon}_0\downarrow 1 : L \rightarrow \grave{\upsilon}_0, \langle\hat{\alpha}\rangle \S \grave{\upsilon}_0\dagger 1), (\grave{\upsilon}_0\downarrow 1 : L \rightarrow \grave{\sigma}_0, update\hat{\alpha}(\grave{\upsilon}_0\downarrow 1)\grave{\sigma}_0)\rangle\rangle$

then $p\hat{\pi}_1 \wedge set\hat{\pi}_1\hat{\pi} \wedge field(rend[\![E_0]\!](tie\psi_3\grave{\rho}_1))\grave{\upsilon}_1\grave{\sigma}_1 = true$, as $\mathfrak{A}q_0\grave{\rho}_1[\![I]\!]\downarrow 1 = \mathfrak{A}q_0\grave{\rho}[\![I]\!]\downarrow 1 = \mathfrak{A}q_0\grave{\rho}_2[\![I]\!]\downarrow 1 = \mathfrak{A}q_0\grave{\rho}_0[\![I]\!]\downarrow 1$ unless $\#\grave{\rho}_1[\![I]\!] = 0$ or $I : \mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]$. For all such $\hat{\pi}_0$, therefore, $a\langle \vec{\zeta}\vec{\rho}_1\upsilon_1\vec{\sigma}_1, \grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle = true$ and $a\langle mv(\vec{\zeta}\circ revert\vec{\rho})\vec{\rho}_0\upsilon_0\vec{\sigma}_0, mv(\grave{\zeta}\circ revert\grave{\rho})\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle = true$, so that $\langle mv(\vec{\zeta}\circ revert\vec{\rho}), mv(\grave{\zeta}\circ revert\grave{\rho})\rangle$ obeys the conditions laid down for $\hat{\zeta}_2$ above. Thus whenever $sewn[\![\Delta_0]\!]0\psi_0\hat{\pi}\hat{\pi}_2 = true$ we know that $c\langle \mathscr{L}[\![E_0]\!](\vec{\zeta}\circ revert\vec{\rho}), \mathscr{L}[\![e[\![E_0]\!]\psi_0]\!](\grave{\zeta}\circ revert\grave{\rho})\rangle\hat{\pi}_2 = true$, and, $D[\![\Delta_0]\!]$ being $true$, $c\langle \mathscr{G}[\![E]\!]\vec{\zeta}, \mathscr{G}[\![g[\![E]\!]\psi_0]\!]\grave{\zeta}\rangle\hat{\pi} = true$ when E is Δ_0 inside E_0.

We next discuss briefly two forms of expression E for which $\mathscr{I}[\![E]\!]\S\mathscr{K}[\![E]\!]$ need not be vacuous.

Suppose that E is of the form E_0; E_1, and that $cramped[\![E]\!]\psi \wedge G[\![E_0]\!] \wedge G[\![E_1]\!] = true$. From the definition of *cramped*

in 1.5.3 it is clear that $cramped[\![E_1]\!]\psi = true$. Accordingly by the induction hypothesis $c(\mathscr{G}[\![E_1]\!]\acute{\zeta}, \mathscr{G}[\![g\ [\![E_1]\!]\psi_0]\!]\grave{\zeta})\,\hat{\pi} = true$ and $k(\lambda\rho\upsilon.\mathscr{G}[\![E_1]\!]\acute{\zeta}\rho(\upsilon\dagger 1), \lambda\rho\upsilon.\mathscr{G}[\![g\ [\![E_1]\!]\psi_0]\!]\grave{\zeta}\rho(\upsilon\dagger 1))\,\hat{\pi} = true$, giving $c(\mathscr{G}[\![E_0;\ E_1]\!]\acute{\zeta}, \mathscr{G}[\![g\ [\![E_0;\ E_1]\!]\psi_0]\!]\grave{\zeta})\,\hat{\pi} = true$. Furthermore analogous conclusions apply to $\omega(\acute{\omega}*\downarrow\nu, \grave{\omega}*\downarrow\nu)\,\hat{\pi}$ when $\acute{\omega}*$ and $\grave{\omega}*$ are defined as in the statement of the lemma and $1 \le \nu \le \#\mathscr{F}[\![E]\!]\S\mathscr{A}[\![E]\!]$.

Likewise if E is of the form if E_0 then E_1 else E_2 and $cramped[\![E]\!]\psi \wedge R[\![E_0]\!] \wedge G[\![E_1]\!] \wedge G[\![E_2]\!] = true$, then $cramped[\![E_1]\!]\psi \wedge cramped[\![E_2]\!]\psi = true$ so we may apply the induction hypothesis to obtain .
$c(\mathscr{G}[\![E_1]\!]\acute{\zeta}, \mathscr{G}[\![g\ [\![E_1]\!]\psi_0]\!]\grave{\zeta})\,\hat{\pi} \wedge c(\mathscr{G}[\![E_2]\!]\acute{\zeta}, \mathscr{G}[\![g\ [\![E_2]\!]\psi_0]\!]\grave{\zeta})\,\hat{\pi} = true$. When $\hat{\zeta}_1$ and $\hat{\zeta}_2$ satisfy $c\hat{\zeta}_2\hat{\pi} \wedge c\hat{\zeta}_3\hat{\pi} = true$, $k\hat{\zeta}_3\hat{\pi} = true$ where $\acute{\zeta}_3 = s\upsilon(\lambda\rho\upsilon.\upsilon\downarrow 1 \rightarrow \zeta_1\rho(\upsilon\dagger 1), \acute{\zeta}_2\rho(\upsilon\dagger 1))$ (and similarly for $\grave{\zeta}_3$); consequently $c(\mathscr{R}[\![E_0]\!]\acute{\zeta}_3, \mathscr{R}[\![e\ [\![E_0]\!]\psi_0]\!]\grave{\zeta}_3)\,\hat{\pi} = true$ and, taking $\hat{\zeta}_n = (\mathscr{G}[\![E_n]\!]\acute{\zeta}, \mathscr{G}[\![g\ [\![E_n]\!]\psi_0]\!]\grave{\zeta})$ when n is 1 or 2, $c(\mathscr{G}[\![E]\!]\acute{\zeta}, \mathscr{G}[\![g\ [\![E]\!]\psi_0]\!]\grave{\zeta})\,\hat{\pi} = true$. Again we may resort to the same technique for $\acute{\omega}*$ and $\grave{\omega}*$.

Finally, suppose that E satisfies the conclusions of the lemma, and take any ψ_2 and $\hat{\pi}_2$ having $apt\psi_2\hat{\rho}_2 \wedge rent[\![E]\!]\psi_2 \wedge fit\hat{\pi}_2\hat{\pi}_2 = true$ and $\hat{\rho}_2[\![I]\!]\downarrow 1 : \mathbb{V}$ for every I:Ide such that $\psi[\![I]\!]\downarrow 1 = 1$; we know that $cramped[\![E]\!]\psi = true$. Let $\hat{\zeta}_2$ be such that $a(\acute{\zeta}_2\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0, \grave{\zeta}_2\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0) = true$ whenever $\hat{\pi}_0$ satisfies $p\hat{\pi}_0 \wedge set\hat{\pi}_0\hat{\pi}_2 = true$ and $field(rend[\![E]\!](tie\psi\hat{\rho}_0))\dot{\upsilon}_0\dot{\sigma}_0 \wedge (\dot{\upsilon}_0\downarrow 1 : \mathbb{L} \rightarrow \sim plot(\dot{\upsilon}_0\downarrow 1)\dot{\rho}_0\dot{\upsilon}_0\dot{\sigma}_0, true) = true$. Define $\hat{\zeta}_1 = (\acute{\zeta}_2 \circ revert\acute{\rho}_2, \grave{\zeta}_2 \circ revert\grave{\rho}_2)$; should $\hat{\pi}_1$ have $p\hat{\pi}_1 = true$, $field(rend[\![E]\!](tie\psi\dot{\rho}_1))\dot{\upsilon}_1\dot{\sigma}_1 \wedge (\dot{\upsilon}_1\downarrow 1 : \mathbb{L} \rightarrow \sim plot(\dot{\upsilon}_1\downarrow 1)\dot{\rho}_1\dot{\upsilon}_1\dot{\sigma}_1, true) = true$ and $\bigwedge\{\dot{\rho}_1[\![I]\!]\downarrow 1 = \dot{\rho}_0[\![I]\!]\downarrow 1 \vee (\#\psi[\![I]\!] = 0 \rightarrow true, \psi[\![I]\!]\downarrow 1 = 3) \mid I : Ide\} = true$ for some $\hat{\rho}_0$ having $set((\acute{\rho}_0, \acute{\upsilon}_1, \acute{\sigma}_1), (\dot{\rho}_0, \dot{\upsilon}_1, \dot{\sigma}_1))\,\hat{\pi}_2 = true$, we shall have $rend[\![E]\!](tie\psi\dot{\rho}_1) = rend[\![E]\!](tie\psi\dot{\rho}_0)$ and $a(\acute{\zeta}_1\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, \grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1) = a(\acute{\zeta}_2\acute{\rho}_0\acute{\upsilon}_1\acute{\sigma}_1, \grave{\zeta}_2\grave{\rho}_0\grave{\upsilon}_1\grave{\sigma}_1) = true$. As E satisfies the conclusions of the lemma we may infer that

$c\langle \mathscr{S}[\![E]\!]\zeta_1', \mathscr{S}[\![\mathscr{y}[\![E]\!]\psi_0]\!]\zeta_1\rangle\hat{\pi}_1 = true$ for all $\hat{\pi}_1$ such that $apt\psi_0\hat{\rho}_1 = true$, $fit\hat{\pi}_1\hat{\pi}_1 = true$ and $\bigwedge\{\rho_1[\![I]\!]\downarrow 1 = \rho_0[\![I]\!]\downarrow 1 \vee (\#\psi[\![I]\!] = 0 \rightarrow true, \psi[\![I]\!]\downarrow 1 = 3) \mid I : Ide\} = true$, where $\psi_0 = \psi_2[false^*/\mathscr{I}[\![E]\!]][opts(\mathscr{N}[\![E]\!]\psi_2)/\mathscr{N}[\![E]\!]]$. Similar remarks are pertinent to $w\langle \hat{\omega}^*\downarrow\nu, \hat{\omega}^*\downarrow\nu\rangle\hat{\pi}_1$ when $\hat{\omega}^*$ and $\hat{\omega}^*$ are as above and $1 \le \nu \le \#\mathscr{J}[\![E]\!] \S \mathscr{N}[\![E]\!]$, so the argument of 2.5.1 shows that $c\langle \mathscr{E}[\![E]\!]\zeta_2', \mathscr{E}[\![e[\![E]\!]\psi_2]\!]\zeta_2\rangle\hat{\pi}_2 = true$.⊁

Before passing on to the main part of the structural induction we modify the proof of 2.5.3 so that it copes with the new version of w. Suppose that $k\hat{\zeta}\hat{\pi}\wedge p\hat{\pi}\wedge fit\hat{\pi}\hat{\pi} = true$ and that $\rho[\![I]\!]\downarrow 1 : G$ for some $\hat{\zeta}$ and $\hat{\pi}$. Then, writing $\hat{\delta}$ for $\langle \rho[\![I]\!]\downarrow 1, \rho[\![I]\!]\downarrow 1\rangle$, $hoten\hat{\delta}\hat{\rho} = true$ so $w\hat{\delta}\hat{\pi} = true$ and $hoten((q_0 \times q_0)\hat{\delta})((\mathbb{q}_0 \times \mathbb{q}_0)\hat{\rho}) = true$. Consequently $a\langle \mathscr{S}[\![I]\!]\zeta\rho\upsilon\sigma, \mathscr{S}[\![g[\![I]\!]\psi]\!]\zeta\rho\upsilon\sigma\rangle = true$ when $apt\psi\hat{\rho} = true$.

In view of our changes to the definitions of fit and set we must also amend the proof that $G[\![E_0 := E_1]\!] = true$ to ensure that certain locations are not assigned to. All that is required is an elucidation of the remark to the effect that $set\hat{\pi}_3\hat{\pi} = true$ (in the third paragraph of 2.5.5). For any α such that $\alpha = \upsilon_3\downarrow\nu$ for some $\nu > 1$ and $spot\alpha\rho\upsilon\delta = false$ we shall show that $q_0(hold\alpha\delta_3) = q_0(hold\alpha\delta)$ and $spot\alpha\rho_3\upsilon_3\delta_3 = false$. Because $set\hat{\pi}_1\hat{\pi} = true$ we know that $\alpha = \upsilon_1\downarrow\nu$ and $spot\alpha\rho_1\upsilon_1\delta_1 = false$; thus because $set\hat{\pi}_0\hat{\pi}_1 = true$ $spot\alpha\rho_0\upsilon_0\delta_0 = false$, and because $fit\hat{\pi}_3\hat{\pi}_0 = true$ $spot\alpha\rho_2\upsilon_2\delta_2 = false$ and $q_0(hold\alpha\delta_2) = q_0(hold\alpha\delta_0) = q_0(hold\alpha\delta)$. Moreover as $\alpha = \upsilon_1\downarrow\nu$ for some $\nu > 1$ $plot\alpha\rho_1(\langle hold(\upsilon_1\downarrow 1)\delta_1\rangle \S \upsilon_1\dagger 1)\delta_1 = true$ although $spot\alpha\rho_1\upsilon_1\delta_1 = false$; hence as $set\hat{\pi}_1\hat{\pi} = true$ we cannot have $\alpha = \upsilon_1\downarrow 1$. Finally $spot\alpha\rho_3\upsilon_3\delta_3 = false$ and $q_0(hold\alpha\delta_3) = q_0(hold\alpha\delta_2) = q_0(hold\alpha\delta)$, thereby establishing that $set\hat{\pi}_3\hat{\pi} = true$.

A scholium is also necessary in the third paragraph of

2.5.4, as in accordance with 2.6.1 we now presume that $\delta_1=\delta_2$ and that $set\hat{\pi}_2\langle\langle\rho_2,\upsilon_1,\sigma_2\rangle,\langle\grave{\rho}_2,\grave{\upsilon}_1,\grave{\sigma}_2\rangle\rangle=true$. Together with the fact that $fit\hat{\pi}_3\hat{\pi}_2=true$ this ensures that when $\alpha=\grave{\upsilon}_1\downarrow\nu$ for some $\nu>0$ and when $plot\alpha\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2=false$ then α is not $\grave{\upsilon}_3\downarrow 1$ and $plot\alpha\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4=false$ so that $fit\hat{\pi}_4\langle\langle\rho_2[\upsilon_2\downarrow 1/I],\upsilon_2\uparrow 1,\sigma_2\rangle,\langle\grave{\rho}_2[\grave{\upsilon}_2\downarrow 1/I],\grave{\upsilon}_2\uparrow 1,\grave{\sigma}_2\rangle\rangle=true$ as required.

2.6.3. Lemma.

If $E[\![E_0]\!]\wedge E[\![E_1]\!]=true$ then $G[\![E_0E_1]\!]=true$.

⊲By modifying the argument of 2.5.5 it can be seen that $G[\![E_0E_1]\!]=true$ if whenever $\hat{\zeta}$ and $\hat{\pi}$ obey $k\hat{\zeta}\hat{\pi}=true$ then $c\hat{\zeta}_0\hat{\pi}_0=true$ where

$\acute{\zeta}_0=\lambda\rho\upsilon.\upsilon\downarrow 2:F\rightarrow(\upsilon\downarrow 2\downarrow 1)(\acute{\zeta}\circ revert\rho)(divert\rho(\upsilon\downarrow 2\downarrow 2))(\langle\upsilon\downarrow 1\rangle\S\upsilon\uparrow 2),$

$sv(\lambda\rho\upsilon.1\leq\upsilon\downarrow 1\,|\,N\leq\#\upsilon\downarrow 2\,|\,L^*\rightarrow\acute{\zeta}\rho(\langle\upsilon\downarrow 2\downarrow(\upsilon\downarrow 1)\rangle\S\upsilon\uparrow 2),\tau)\rho\upsilon$

and $\acute{\pi}_0=\langle\acute{\rho},\langle\acute{\alpha},\acute{\beta}\rangle\S\acute{\upsilon},\acute{\sigma}\rangle$ for some $\acute{\alpha}:L$ and $\acute{\beta}:V$ (and similarly for $\grave{\zeta}_0$ and $\grave{\pi}_0$). Accordingly, take any suitable $\hat{\zeta}$ and $\hat{\pi}$ and let $\hat{\pi}_1$ be any pair having $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$.

When $\acute{\upsilon}_1\downarrow 2:F$ then $\grave{\upsilon}_1\downarrow 2:F$ also and we may set up

$\hat{\pi}_2=\langle\langle divert\acute{\rho}_1(\acute{\upsilon}_1\downarrow 2\downarrow 2),\langle\acute{\upsilon}_1\downarrow 1\rangle\S\acute{\upsilon}_1\uparrow 2,\acute{\sigma}_1\rangle,\langle divert\grave{\rho}_1(\grave{\upsilon}_1\downarrow 2\downarrow 2),\langle\grave{\upsilon}_1\downarrow 1\rangle\S\grave{\upsilon}_1\uparrow 2,\grave{\sigma}_1\rangle\rangle.$

Because $\omega\langle\acute{\upsilon}_1\downarrow 2,\grave{\upsilon}_1\downarrow 2\rangle\hat{\pi}_1=true$ and $k\hat{\zeta}\langle\langle\acute{\rho}_1,\acute{\upsilon}_1\uparrow 2,\acute{\sigma}_1\rangle,\langle\grave{\rho}_1,\grave{\upsilon}_1\uparrow 2,\grave{\sigma}_1\rangle\rangle=k\hat{\zeta}\hat{\pi}=true$, should $p\hat{\pi}_2$ be $true$ $a\langle(\acute{\upsilon}_1\downarrow 2\downarrow 1)(\acute{\zeta}\circ revert\acute{\rho}_1)\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,(\grave{\upsilon}_1\downarrow 2\downarrow 1)(\grave{\zeta}\circ revert\grave{\rho}_1)\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle$ will be $true$ and $a\langle\acute{\zeta}_0\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle$ will be $true$. However $hoten\hat{\omega}\langle\acute{\upsilon}_1\downarrow 2\downarrow 2,\grave{\upsilon}_1\downarrow 2\downarrow 2\rangle\supset kent1\hat{\omega}\hat{\pi}_1$ for all $\hat{\omega}$ so familiar techniques serve to confirm that $kent\nu\hat{\omega}\hat{\pi}_2\supset kent\nu\hat{\omega}\hat{\pi}_1$ for all $\nu<2$ and $\hat{\omega}$ and thus that $p\hat{\pi}_2=true$. Hence $a\langle\acute{\zeta}_0\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$ and, $\hat{\pi}_1$ being typical of those pairs having $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$, $c\hat{\zeta}_0\hat{\pi}_0=true$.

The proof necessary when $\acute{\upsilon}_1\downarrow 2$ is not a member of follows a predictable pattern and can safely be omitted.⊳

2.6.4. Lemma.

If $E[\![E]\!]=true$ then $G[\![\text{val } E]\!]\wedge G[\![\text{res } E]\!]\wedge G[\![\text{goto } E]\!]=true$.

⊰The proof that $G[\![\text{val } E]\!]=true$ is a greatly simplified version of 2.5.1 and will therefore be left out. The other two proofs are closely connected so only that for $G[\![\text{res } E]\!]=true$ will be discussed further.

The proof that $L[\![E]\!]=true$ given in 2.5.2 is equally apposite for our present version of p_0, so to show that $G[\![\text{res } E]\!]=true$ it suffices to show that for any $\hat{\zeta}$ and $\hat{\pi}_0$ having $k\hat{\zeta}\hat{\pi}_0=true$ if we define

$\acute{\zeta}_0=\lambda\rho\upsilon.(\rho[\![\text{res}]\!]\downarrow 1\downarrow 1)(\rho[\![\text{res}]\!]\downarrow 1\downarrow 2)(\langle\upsilon\downarrow 1\rangle\S\rho[\![\text{res}]\!]\downarrow 1\downarrow 3)$ (and similarly for $\grave{\zeta}_0$) then $k\hat{\zeta}_0\hat{\pi}_0=true$. Accordingly, take any such $\hat{\zeta}$ and $\hat{\pi}_0$ and any $\hat{\pi}_1$ having $p\hat{\pi}_1\wedge set\hat{\pi}_1\hat{\pi}_0=true$; if $\#\rho_1[\![\text{res}]\!]=0$ the result is immediate so suppose that this is not the case and let

$\hat{\pi}_2=\langle\acute{\rho}_1[\![\text{res}]\!]\downarrow 1\downarrow 2,\langle\acute{\upsilon}_1\downarrow 1\rangle\S\acute{\rho}_1[\![\text{res}]\!]\downarrow 1\downarrow 3,\acute{\sigma}_1\rangle$ (and similarly for $\grave{\pi}_2$). Because $kent1\langle\acute{\rho}_1[\![\text{res}]\!]\downarrow 1,\grave{\rho}_1[\![\text{res}]\!]1\rangle\hat{\pi}_1=true$ and $p_1\hat{\pi}_2=true$ (as is shown by 2.4.6) we know that

$k\langle\acute{\rho}_1[\![\text{res}]\!]\downarrow 1\downarrow 1,\grave{\rho}_1[\![\text{res}]\!]\downarrow 1\downarrow 1\rangle\langle\langle\acute{\rho}_2,\acute{\upsilon}_2\dagger 1,\acute{\sigma}_1\rangle,\langle\grave{\rho}_2,\grave{\upsilon}_2\dagger 1,\grave{\sigma}_1\rangle\rangle=true$ and that $kent\nu\hat{\omega}\hat{\pi}_2\supset kent\nu\hat{\omega}\hat{\pi}_1$ for all $\nu<2$ and $\hat{\omega}$. From the latter assertion we deduce that $p\hat{\pi}_2=true$ and that $set\hat{\pi}_2\hat{\pi}_0=true$; from the former, however,

$k\langle\acute{\rho}_1[\![\text{res}]\!]\downarrow 1\downarrow 1,\grave{\rho}_1[\![\text{res}]\!]\downarrow 1\downarrow 1\rangle\hat{\pi}_0=true$ and so we may conclude that $a\langle\acute{\zeta}_0\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$ and that $k\hat{\zeta}_0\hat{\pi}_0=true$.⊱

2.6.5. Lemma.

Let Δ be of the form $I=E$, $I_1,\ldots,I_n=E$, $I==E$ or $I_1,\ldots,I_n==E$ for some I or $I_1,\ldots,I_n$ and some E having $E[\![E]\!]=true$; then $D[\![\Delta]\!]=true$ and, when Δ is $I=E$ or $I_1,\ldots,I_n=E$, $T[\![\Delta]\!]=true$. When Δ is $I==E$ or $I_1,\ldots,I_n==E$ and when ψ, $\hat{\zeta}_0$ and $\hat{\pi}_0$ are any entities such that

$\wedge\{I:\mathscr{M}[\![\Delta]\!]\to\psi[\![I]\!]\downarrow 1=\mathit{false},\mathit{true}\mid I:Ide\}=\mathit{true}$, $\mathit{cramped}[\![\Delta]\!](\lambda I.\langle 2\rangle)=\mathit{true}$ and for all $\hat{\pi}$ if $\mathit{sewn}[\![\Delta]\!]1\psi\hat{\pi}_0\hat{\pi}=\mathit{true}$ and $\wedge\{\mathit{field}(\mathit{tear}[\![\Delta]\!]\hat{\rho}_0)\langle\hat{\rho}[\![I]\!]\downarrow 1\rangle\hat{\sigma}\mid I:\mathscr{M}[\![t[\![\Delta]\!]\psi]\!]\}=\mathit{true}$ then $c\hat{\zeta}_0\hat{\pi}=\mathit{true}$ we have $c\langle\mathscr{T}[\![\Delta]\!]\acute{\zeta}_0,\mathscr{T}[\![t[\![\Delta]\!]\psi]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=\mathit{true}$.

⊲Strictly speaking the hypothesis that $\mathit{cramped}[\![\Delta]\!](\lambda I.\langle 2\rangle)=\mathit{true}$ is not adequate to prove the result, as really what we require is that E (in the case I==E) and E_m when $1\le m\le n$ (in the case $I_1,\ldots,I_n$==E) satisfy the conclusions of 2.6.2. For brevity we do not consider the possibility that $\mathit{cramped}[\![\Delta]\!](\lambda I.\langle 2\rangle)=\mathit{true}$ does not entail this, since in a properly stated induction on the size of Mal programs it would do so, and our lemmata could obviously be stated in this more pedantic form.

As an example of the proof that $D[\![\Delta]\!]=\mathit{true}$ we take Δ to be I==E and consider any ψ, $\hat{\zeta}_0$ and $\hat{\pi}_0$ such that $c\hat{\zeta}_0\hat{\pi}=\mathit{true}$ whenever $\mathit{sewn}[\![\Delta]\!]0\psi\hat{\pi}_0\hat{\pi}=\mathit{true}$. The usual arguments show that $k\langle\lambda\rho\upsilon.\acute{\zeta}_0\rho[\upsilon\downarrow 1/I](\upsilon\dagger 1),\lambda\rho\upsilon.\grave{\zeta}_0\rho[\upsilon\downarrow 1/I](\upsilon\dagger 1)\rangle\hat{\pi}_0=\mathit{true}$ so, as $R[\![E]\!]=\mathit{true}$ by 2.5.2, when $\mathit{opt}[\![I]\!]\psi=\mathit{false}$ we have $c\langle\mathscr{D}[\![\Delta]\!]\acute{\zeta}_0,\mathscr{D}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=\mathit{true}$. Accordingly we turn to the more interesting case that arises when $\mathit{opt}[\![I]\!]=\mathit{true}$; then we can assert that $\langle\mathscr{D}[\![\Delta]\!]\acute{\zeta}_0,\mathscr{D}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\grave{\zeta}_0\rangle=\langle\mathscr{R}[\![E]\!]\acute{\zeta}_1,\mathscr{R}[\![\mathscr{A}[\![E]\!]\psi]\!]\grave{\zeta}_1\rangle$ where $\hat{\zeta}_1=\langle\lambda\rho\upsilon.\acute{\zeta}_0\rho[\upsilon\downarrow 1/I](\upsilon\dagger 1),mv(\lambda\rho\upsilon.\grave{\zeta}_0\rho[\upsilon\downarrow 1/I](\upsilon\dagger 1))\rangle$.

It will be enough to show that $k\langle sv\acute{\zeta}_1,sv\grave{\zeta}_1\rangle\hat{\pi}_0=\mathit{true}$ since $E[\![E]\!]=\mathit{true}$. Take any $\hat{\pi}_1$ having $p\hat{\pi}_1\wedge set\hat{\pi}_1\hat{\pi}_0=\mathit{true}$ and define $\grave{\alpha}=new\grave{\sigma}_1$, $\hat{\varepsilon}=access\langle\acute{\upsilon}_1\downarrow 1,\grave{\upsilon}_1\downarrow 1\rangle\hat{\pi}_1$ and $\hat{\pi}_2=\langle\langle\acute{\rho}_1[\varepsilon/I],\acute{\upsilon}_1\dagger 1,\acute{\sigma}_1\rangle,\langle\grave{\rho}_1[\grave{\alpha}/I],\grave{\upsilon}_1\dagger 1,update\grave{\alpha}\varepsilon\grave{\sigma}_1\rangle\rangle$. Because $p_0\hat{\pi}_1=\mathit{true}$, $\hat{\varepsilon}$ is proper and $\hat{\pi}_2$ is proper; moreover $\mathit{sewn}[\![\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}_2=\mathit{true}$ as we cannot have $\grave{\alpha}=\grave{\upsilon}_1\downarrow\nu$ for some ν. By the technique of 2.5.3 we can readily demonstrate that for all ν and $\hat{\omega}$ $\mathit{kent}\nu\hat{\omega}\hat{\pi}_2\supset(\hat{\omega}=\langle\varepsilon,\grave{\alpha}\rangle\vee\mathit{kent}\nu\hat{\omega}\hat{\pi}_1)$ and thus that $w\hat{\omega}\hat{\pi}_2=\mathit{true}$ when $\mathit{kent}0\hat{\omega}\hat{\pi}_2=\mathit{true}$. Since $\mathit{area}\grave{\alpha}\grave{\sigma}_1=\mathit{false}$ and $p_0\hat{\pi}_1=\mathit{true}$, $\mathit{site}\grave{\alpha}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1=\mathit{false}$

and therefore $p_0\hat{\pi}_2=true$ also. Hence $p\hat{\pi}_2=true$ and $a\langle sv\acute{\zeta}_1\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, sv\grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=a\langle \acute{\zeta}_0\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2, \grave{\zeta}_0\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle=true$; $\hat{\pi}_1$ being any pair with $p\hat{\pi}_1\wedge set\hat{\pi}_1\hat{\pi}_0=true$ we can infer that $k\langle sv\acute{\zeta}_1, sv\grave{\zeta}_1\rangle\hat{\pi}_0=true$ and thus that $c\langle\mathscr{D}[\![\Delta]\!]\acute{\zeta}_0, \mathscr{D}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$. This equality is valid for all suitable ψ, $\hat{\zeta}_0$ and $\hat{\pi}_0$, so $D[\![\Delta]\!]=true$.

⊲We shall only consider the second part of the result when Δ is $I_1,\ldots,I_n$==E. Indeed we shall even restrict ourselves to the proof required when E is $E_1,\ldots,E_n$, that for $\pounds(E_1,\ldots,E_n)$ being almost identical. Accordingly we assume that when $1\leq m\leq n$ the expression E_m satisfies the conclusions of 2.6.2.

⊲Suppose that ψ, $\hat{\zeta}_0$ and $\hat{\pi}_0$ are any entities such that $c\hat{\zeta}_0\hat{\pi}=true$ whenever $\hat{\pi}$ is subject to the constraints $sewn[\![\Delta]\!]1\psi\hat{\pi}_0\hat{\pi}=true$ and $\bigwedge\{field(tear[\![\Delta]\!]\grave{\rho}_0)\langle\grave{\rho}[\![I]\!]\downarrow1\rangle\grave{\sigma}\mid I:\mathscr{R}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\}=true$. If $\psi[\![I_1]\!]\downarrow1\wedge\ldots\wedge\psi[\![I_n]\!]\downarrow1=false$ we shall show that $c\langle\mathscr{T}[\![\Delta]\!]\acute{\zeta}_0, \mathscr{T}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$. To this end we set $I^*=\langle I_1,\ldots,I_n\rangle$, $\hat{\zeta}_1=\langle(\lambda\rho\upsilon\sigma.\#\upsilon\downarrow1\mid L^*=n\rightarrow\acute{\zeta}_0(invert\rho(arid[holds(\upsilon\downarrow1)\sigma/I^*]))(\upsilon\uparrow1)\sigma,\tau),$ $(\lambda\rho\upsilon\sigma.\#\upsilon\downarrow1\mid L^*=n\rightarrow\grave{\zeta}_0(invert\rho(arid[holds(\upsilon\downarrow1)\sigma/I^*]))(\upsilon\uparrow1)\sigma,\tau)\rangle$ and $\hat{\zeta}_2=\langle\lambda\rho\upsilon.\acute{\zeta}_1\rho((\langle\upsilon\downarrow n,\ldots,\upsilon\downarrow1\rangle)\S\upsilon\uparrow n), \lambda\rho\upsilon.\grave{\zeta}_1\rho((\langle\upsilon\downarrow n,\ldots,\upsilon\downarrow1\rangle)\S\upsilon\uparrow n)\rangle$, so that $\langle\mathscr{T}[\![\Delta]\!]\acute{\zeta}_0, \mathscr{T}[\![\mathscr{A}[\![\Delta]\!]\psi]\!]\grave{\zeta}_0\rangle=\langle\mathscr{R}[\![E]\!]\acute{\zeta}_1, \mathscr{R}[\![\mathscr{A}[\![E]\!]\psi]\!]\grave{\zeta}_1\rangle$. For simplicity we presume that *mete* dictates an order of evaluation from left to right, and we define $\hat{\zeta}_{m+1}=\langle\mathscr{L}[\![E_{n-m+2}]\!]\acute{\zeta}_m\mathscr{L}[\![\mathscr{A}[\![E_{n-m+2}]\!]\psi]\!]\grave{\zeta}_m\rangle$ for $2\leq m\leq n+1$; from the equations of appendix 2 $\langle\mathscr{R}[\![E]\!]\acute{\zeta}_1, \mathscr{R}[\![\mathscr{A}[\![E]\!]\psi]\!]\grave{\zeta}_1\rangle=\hat{\zeta}_{n+2}$.

Thus $c\hat{\zeta}_{n+2}\hat{\pi}_0$ will be *true* if any $\hat{\pi}_{n+2}$ satisfying $p\hat{\pi}_{n+2}\wedge fit\hat{\pi}_{n+2}\hat{\pi}_0=true$ has $a\langle\acute{\zeta}_{n+2}\acute{\rho}_{n+2}\acute{\upsilon}_{n+2}\acute{\sigma}_{n+2}, \grave{\zeta}_{n+2}\grave{\rho}_{n+2}\grave{\upsilon}_{n+2}\grave{\sigma}_{n+2}\rangle=true$. By 2.6.2 to verify that this is so for some suitable $\hat{\pi}_{n+2}$ we need only show that when $m=n+1$ any $\hat{\pi}_m$ with $p\hat{\pi}_m\wedge set\hat{\pi}_m\hat{\pi}_{m+1}=true$, $\upsilon_m\downarrow1:L$, $field(rend[\![E_{n-m+2}]\!]\grave{\rho}_{m+1})\langle\upsilon_m\downarrow1\rangle\grave{\sigma}_m=true$ and $spot(\upsilon_m\downarrow1)\grave{\rho}_m\upsilon_m\grave{\sigma}_m=false$ is such that $a\langle\acute{\zeta}_m\acute{\rho}_m\acute{\upsilon}_m\acute{\sigma}_m, \grave{\zeta}_m\grave{\rho}_m\grave{\upsilon}_m\grave{\sigma}_m\rangle=true$. Continuing along this train of reasoning, $c\hat{\zeta}_{n+2}\hat{\pi}_0$ will be *true* if all sequences $\hat{\pi}_{n+2},\ldots,\hat{\pi}_2$ with $\hat{\pi}_{m+1}$ and $\hat{\pi}_m$ related as in the preceding sentence when $2\leq m\leq n+1$ and with $p\hat{\pi}_{n+2}\wedge fit\hat{\pi}_{n+2}\hat{\pi}_0=true$ are subject to

$a\langle \zeta_2\rho_2\upsilon_2\sigma_2, \zeta_2\rho_2\upsilon_2\sigma_2\rangle = true$. When $2\leq m\leq n$, if $\upsilon_{m+1}\downarrow\nu$:L and $spot(\upsilon_{m+1}\downarrow\nu)\rho_{m+1}\upsilon_{m+1}\sigma_{m+1}=false$ then $spot(\upsilon_{m+1}\downarrow\nu)\rho_m\upsilon_m\sigma_m=false$, $q_0(hold(\upsilon_{m+1}\downarrow\nu)\sigma_{m+1})=q_0(hold(\upsilon_{m+1}\downarrow\nu)\sigma_m)$ and $\upsilon_{m+1}\downarrow\nu=\upsilon_m\downarrow(\nu+1)$ as $set\pi_m\pi_{m+1}=true$. Accordingly we can show by induction that for all m with $2\leq m\leq n+1$ $spot(\upsilon_m\downarrow 1)\rho_2\upsilon_2\sigma_2=false$, $\upsilon_m\downarrow 1=\upsilon_2\downarrow(m-1)$ and $q_0(hold(\upsilon_m\downarrow 1)\sigma_m)=q_0(hold(\upsilon_m\downarrow 1)\sigma_2)$; since in addition $\mathfrak{A}q_0\rho_{m+1}=\mathfrak{A}q_0\rho_0$ and for every I:Ide $rend[\![E_{n-m+2}]\!]\rho_{m+1}[\![I]\!]=(\#rend[\![E_{n-m+2}]\!]\rho_{m+1}[\![I]\!]>0\rightarrow tear[\![\Delta]\!]\rho_{m+1}[\![I]\!],\langle\rangle)$ we may deduce that in fact $field(tear[\![\Delta]\!]\rho_0)(\upsilon_2\downarrow(m-1))\sigma_2=true$ when $2\leq m\leq n+1$.

Now we introduce the pair

$\pi_1=\langle\langle invert\rho_2(arid[holds\langle\upsilon_2\downarrow n,\ldots,\upsilon_2\downarrow 1\rangle\sigma_2/I^*]),\upsilon_2\dagger n,\sigma_2\rangle,$

$\langle invert\rho_2(arid[holds\langle\upsilon_2\downarrow n,\ldots,\upsilon_2\downarrow 1\rangle\sigma_2/I^*]),\upsilon_2\dagger n,\sigma_2\rangle\rangle,$

for which $\langle\zeta_2\rho_2\upsilon_2\sigma_2,\zeta_2\rho_2\upsilon_2\sigma_2\rangle=\langle\zeta_0\rho_1\upsilon_1\sigma_1,\zeta_0\rho_1\upsilon_1\sigma_1\rangle$. For all ω $yclept\omega\pi_1\supset kent1\omega\pi_2$ and $gyven\omega 0_1\supset gyven\omega 0_2$; hence for all ν and ω $kent\nu\omega\pi_1\supset kent\nu\omega\pi_2$, and in consequence $p\pi_1\wedge sewn[\![\Delta]\!]1\psi\pi_0\pi_1=true$. Furthermore $\wedge\{field(tear[\![\Delta]\!]\rho_0)\langle\rho_1[\![I]\!]\downarrow 1\rangle\sigma_1\mid I:\mathscr{H}\ell[\![\Delta]\!]\psi[\![\}=true$, so by our initial supposition about ζ_0 $a\langle\zeta_0\rho_1\upsilon_1\sigma_1,\zeta_0\rho_1\upsilon_1\sigma_1\rangle=true$, which in turn shows that $c\zeta_{n+2}\pi_0=true$ and that $c\langle\mathscr{T}[\![\Delta]\!]\zeta_0,\mathscr{T}[\![\ell[\![\Delta]\!]\psi]\!]\zeta_0\rangle\pi_0=true$.⊁

As ψ, ζ_0 and π_0 are typical of the elements such that $\psi[\![I_1]\!]\downarrow 1\wedge\ldots\wedge\psi[\![I_n]\!]\downarrow 1=false$ and $c\zeta_0\pi=true$ whenever $sewn[\![\Delta]\!]1\psi\pi_0\pi=true$ and $\wedge\{field(tear[\![\Delta]\!]\rho_0)\langle\rho[\![I]\!]\downarrow 1\rangle\sigma\mid I:\mathscr{H}\ell[\![\Delta]\!]\psi[\![\}=true$, we can conclude that $T[\![\Delta]\!]=true$.⊁

The proof when Δ is a declaration by reference is a simplified version of this.⊁

Naturally 2.6.2 is irrelevant when this lemma is regarded as a preliminary to 2.5.9, since the restrictions on E are required only in the proof that $\wedge\{field(tear[\![\Delta]\!]\rho_0)\langle\rho_1[\![I]\!]\downarrow 1\rangle\sigma_1\mid I:\mathscr{H}\ell[\![\Delta]\!]\psi[\![\}=true$, which is not needed by the predicates of 2.4.5. Thus for the purposes of

2.5.8 the lemma can be viewed as stating that
$c\langle\mathscr{T}[\![\Delta]\!]\hat{\zeta}_0,\mathscr{T}[\![\iota[\![\Delta]\!]\psi]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$ for all ψ, $\hat{\zeta}_0$ and $\hat{\pi}_0$ having
$\wedge\{I:\mathscr{K}[\![\Delta]\!]\rightarrow\psi[\![I]\!]\downarrow 1=false,true\,|\,I:\mathrm{Ide}\}=true$ and
$\wedge\{c\hat{\zeta}_0\hat{\pi}\,|\,sewn[\![\Delta]\!]1\psi\hat{\pi}_0\hat{\pi}\}=true$.

2.6.6. Lemma.

Let Δ_2 be Δ_0 within Δ_1 for some Δ_0 and Δ_1. If $D[\![\Delta_0]\!]\wedge D[\![\Delta_1]\!]=true$ then $D[\![\Delta_2]\!]=true$, whilst if $D[\![\Delta_0]\!]\wedge T[\![\Delta_1]\!]=true$ then $T[\![\Delta_2]\!]=true$ provided that $cramped[\![\Delta_2]\!](\lambda I.\langle 2\rangle)=true$ and that $\mathscr{K}[\![\Delta_0]\!]\S\mathscr{I}[\![\Delta_1]\!]$ and $\mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_1]\!]$ have no repeated elements.

⊲Suppose that ψ_0, $\hat{\zeta}_0$ and $\hat{\pi}_0$ are entities such that $fit\hat{\pi}_0\hat{\pi}_0=true$ and $c\hat{\zeta}_0\hat{\pi}=true$ whenever $\hat{\pi}$ has $sewn[\![\Delta_2]\!]1\psi_0\hat{\pi}_0\hat{\pi}=true$ and $\wedge\{field(tear[\![\Delta_2]\!]\hat{\rho}_0)\langle\hat{\rho}[\![I]\!]\downarrow 1\rangle\hat{\sigma}\,|\,I:\mathscr{K}[\![\iota[\![\Delta_2]\!]\psi_0]\!]\}=true$; we shall show that $c\langle\mathscr{T}[\![\Delta_2]\!]\hat{\zeta}_0,\mathscr{T}[\![\iota[\![\Delta_2]\!]\psi_0]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$. To this end we define $\psi_1=\psi_0[false*/\mathscr{I}[\![\Delta_0]\!]][opts(\mathscr{K}[\![\Delta_0]\!])\psi/\mathscr{K}[\![\Delta_0]\!]]$,
$\hat{\zeta}_1=\langle\hat{\zeta}_0\circ trim[\![\Delta_1]\!]\hat{\rho}_0,\grave{\zeta}_0\circ trim[\![\Delta_1]\!]\hat{\rho}_0\rangle$ and $\hat{\zeta}_2=\langle\mathscr{T}[\![\Delta_1]\!]\hat{\zeta}_1,\mathscr{T}[\![\iota[\![\Delta_1]\!]\psi_1]\!]\grave{\zeta}_0\rangle$, so that the transformations of 1.4.6 yield $\langle\mathscr{T}[\![\Delta_2]\!]\hat{\zeta}_0,\mathscr{T}[\![\iota[\![\Delta_2]\!]\psi_0]\!]\grave{\zeta}_0\rangle=\langle\mathscr{D}[\![\Delta_0]\!]\hat{\zeta}_2,\mathscr{D}[\![\mathscr{A}[\![\Delta_0]\!]\psi_0]\!]\grave{\zeta}_2\rangle$. As $D[\![\Delta_0]\!]=true$ it suffices to prove that if $\hat{\pi}_1$ satisfies $sewn[\![\Delta_0]\!]0\psi_0\hat{\pi}_0\hat{\pi}_1=true$ then $c\hat{\zeta}_2\hat{\pi}_1=true$. In turn as $T[\![\Delta_1]\!]=true$ this will be established for any such $\hat{\pi}_1$ if $c\hat{\zeta}_1\hat{\pi}_2=true$ whenever $sewn[\![\Delta_1]\!]1\psi_1\hat{\pi}_1\hat{\pi}_2=true$ and $\wedge\{field(tear[\![\Delta_1]\!]\hat{\rho}_1)\langle\hat{\rho}_2[\![I]\!]\downarrow 1\rangle\hat{\sigma}_2\,|\,I:\mathscr{K}[\![\iota[\![\Delta_1]\!]\psi_1]\!]\}=true$.

Accordingly, take any such $\hat{\pi}_1$ and $\hat{\pi}_2$ together with any $\hat{\pi}_3$ having $p\hat{\pi}_3\wedge fit\hat{\pi}_3\hat{\pi}_2=true$, and define $\hat{\pi}_4=\langle\langle trim[\![\Delta_1]\!]\hat{\rho}_0\hat{\rho}_3,\upsilon_3,\sigma_3\rangle,trim[\![\Delta_1]\!]\hat{\rho}_0\hat{\rho}_3,\upsilon_3,\hat{\sigma}_3\rangle$. By the definitions of *fit*, *sewn* and *field* we can assume without loss of generality that $\hat{\pi}_n=(\mathbb{P}q_0\times\mathbb{P}q_0)\hat{\pi}_n$ when $0\leq n\leq 2$. As $neat\hat{\rho}_3\wedge neat\hat{\rho}_0=true$, for every $\hat{\omega}$ $hoten\hat{\omega}\hat{\rho}_4\supset hoten\hat{\omega}\hat{\rho}_3$ and $access\hat{\omega}\hat{\pi}_4=access\hat{\omega}\hat{\pi}_3$; hence for every $\nu_0>2$, ν_1, $\hat{\omega}_0$ and $\hat{\omega}_1$ we know that $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_4\wedge yclept\hat{\omega}_1\hat{\pi}_4\supset seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3\wedge yclept\hat{\omega}_1\hat{\pi}_3$ and $kent\nu_0\hat{\omega}_0\hat{\pi}_4\supset kent\nu_0\hat{\omega}_0\hat{\pi}_3$, and, $p\hat{\pi}_3$ being $true$, $p\hat{\pi}_4$ must be $true$.

It remains to be demonstrated that $sewn⟦\Delta_2⟧1\psi_0\hat{\pi}_0\hat{\pi}_4=true$ and that $\bigwedge\{field(tear⟦\Delta_2⟧\rho_0)\langle\rho_4⟦I⟧\downarrow1\rangle\sigma_4 \mid I:\mathcal{H}⟦\iota⟦\Delta_2⟧\psi_0⟧\}=true$, using the facts that $sewn⟦\Delta_0⟧0\psi_0\hat{\pi}_0\hat{\pi}_1\wedge sewn⟦\Delta_1⟧1\psi_1\hat{\pi}_1\hat{\pi}_2=true$ and $\bigwedge\{field(tear⟦\Delta_1⟧\rho_1)\langle\rho_2⟦I⟧\downarrow1\rangle\sigma_2 \mid I:\mathcal{H}⟦\iota⟦\Delta_1⟧\psi_1⟧\}=true$.

Observe first that $\mathcal{I}⟦\Delta_1⟧=\mathcal{I}⟦\Delta_2⟧$ and that $\mathcal{H}⟦\Delta_1⟧=\mathcal{H}⟦\Delta_2⟧$. For every I:Ide (writing for convenience $\nu=(I:\mathcal{I}⟦\Delta_1⟧\S\mathcal{H}⟦\Delta_1⟧\rightarrow1,0))$ $\#\rho_0⟦I⟧\leq\#\rho_1⟦I⟧$ as $revert\rho_0\rho_1⟦I⟧=\rho_0⟦I⟧$, and $\#\rho_1⟦I⟧\dagger\nu\leq\#\rho_2⟦I⟧\dagger\nu$ as $revert\rho_1\rho_2⟦I⟧\dagger\nu=\rho_1⟦I⟧\dagger\nu$; hence $q_0(revert\rho_0\rho_4)⟦I⟧\dagger\nu=revert\rho_0\rho_2⟦I⟧\dagger\nu=revert\rho_0\rho_1⟦I⟧\dagger\nu=\rho_0⟦I⟧\dagger\nu$ and, similarly, $q_0(revert\rho'_0\rho'_4)⟦I⟧\dagger\nu=\rho'_0⟦I⟧\dagger\nu$. Moreover, unless $I:\mathcal{I}⟦\Delta_2⟧\S\mathcal{H}⟦\Delta_2⟧$ or $\#\rho_0⟦I⟧=0$, $q_0(\rho_4⟦I⟧\downarrow1)=trim⟦\Delta_1⟧\rho_0\rho_2⟦I⟧\downarrow1=revert\rho_0\rho_2⟦I⟧\downarrow1=\rho_0⟦I⟧\downarrow1$ and $q_0(\rho'_4⟦I⟧\downarrow1)=\rho'_0⟦I⟧\downarrow1$. If $\#\rho'_4⟦I⟧=0$ then necessarily $\#\rho'_0⟦I⟧=0$ and, $apt\psi\hat{\rho}_0$ being $true$, $\#\psi_0⟦I⟧=0$. If $I:\mathcal{I}⟦\Delta_2⟧$ then I is not a member of $\mathcal{H}⟦\Delta_0⟧$ so $\psi_1⟦I⟧=\psi_0⟦I⟧$ and, as $apt\psi_1\hat{\rho}_2\wedge fit\hat{\pi}_3\hat{\pi}_2=true$, $\rho'_4⟦I⟧\downarrow1:L$, $\rho_4⟦I⟧\downarrow1:L$ and $\psi_0⟦I⟧\downarrow1=false$. If $I:\mathcal{H}⟦\Delta_2⟧$ then I is not a member of $\mathcal{I}⟦\Delta_0⟧\S\mathcal{H}⟦\Delta_0⟧$ so $\psi_1⟦I⟧=\psi_0⟦I⟧$ and, as $apt\psi\hat{\rho}_2\wedge fit\hat{\pi}_3\hat{\pi}_2=true$, $\rho'_4⟦I⟧\downarrow1:V$ and either $\rho_4⟦I⟧\downarrow1:V$ or $\rho_4⟦I⟧\downarrow1:L$ and $\psi_0⟦I⟧\downarrow1=true$. Hence $apt\psi_0\hat{\rho}_4=true$ and $sewn⟦\Delta_2⟧1\psi_0\hat{\pi}_0\hat{\pi}_4=true$.

Take any $I:\mathcal{H}⟦\iota⟦\Delta_2⟧\psi_0⟧$; as $sewn⟦\Delta_2⟧1\psi_0\hat{\pi}_0\hat{\pi}_4=true$, writing $\varepsilon_4=\rho_4⟦I⟧\downarrow1$ we have $\varepsilon_4:V$ and $field(tear⟦\Delta_2⟧\rho_0)\langle\rho_4⟦I⟧\downarrow1\rangle\sigma_4=field(tear⟦\Delta_2⟧\rho_0)\langle\varepsilon_4\rangle\sigma_2$. Moreover, if I′ is an identifier not in $\mathcal{I}⟦\Delta_0⟧\S\mathcal{H}⟦\Delta_0⟧$ and having $\#\rho_1⟦I'⟧>0$, $tear⟦\Delta_1⟧\rho_1⟦I'⟧=tear⟦\Delta_2⟧\rho_0⟦I'⟧$. We know that $field(tear⟦\Delta_1⟧\rho_1)\langle\varepsilon_4\rangle\sigma_2=true$, but this is not quite powerful enough in general to establish that $field(tear⟦\Delta_2⟧\rho_0)\langle\rho_2⟦I⟧\downarrow1\rangle\sigma_2=true$, as there may be some α:L satisfying $plot\alpha(tear⟦\Delta_2⟧\rho_0)\langle\rangle empty=false$ although $plot\alpha arid\langle\varepsilon_4\rangle empty=plot\alpha arid\langle\rho_1⟦I''⟧\downarrow1\rangle empty=true$ for some I″ in $\mathcal{I}⟦\Delta_0⟧\S\mathcal{H}⟦\Delta_0⟧$. In our case, however, we may presume from 1.5.3 that $cramped⟦\Delta_1⟧(\lambda I''.\langle I'':\mathcal{I}⟦\Delta_0⟧\S\mathcal{H}⟦\Delta_0⟧\rightarrow3,2\rangle)=true$; for such

declarations it can be established by the techniques applied to Δ_0 inside E_0 in 2.6.2 that

$field\langle(\lambda I''.I'':\mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]\to\langle\rangle, tear[\![\Delta_1]\!]\grave{\rho}_1[\![I'']\!]),\langle\rangle,\langle\rangle\rangle\langle\grave{\varepsilon}_4\rangle\grave{\sigma}_2=true$.

Hence for every $\alpha:\mathsf{L}$ if $plot\alpha arid\langle\grave{\varepsilon}_4\rangle empty=true$ then

$plot\alpha\langle(\lambda I''.I'':\mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]\to\langle\rangle, tear[\![\Delta_2]\!]\grave{\rho}_0[\![I'']\!]),\langle\rangle,\langle\rangle\rangle\langle\rangle empty=true$

and consequently $field(tear[\![\Delta_2]\!]\grave{\rho}_0)\langle\grave{\varepsilon}_4\rangle\grave{\sigma}_4=true$.

We have now shown that any pair $\hat{\pi}_3$ constrained by $p\hat{\pi}_3\wedge fit\hat{\pi}_3\hat{\pi}_2=true$ induces a pair $\hat{\pi}_4$ having $p\hat{\pi}_4\wedge sewn[\![\Delta_2]\!]1\psi_0\hat{\pi}_0\hat{\pi}_4=true$; if in addition $cramped[\![\Delta_2]\!](\lambda I.\langle 2\rangle)=true$ and $\wedge\{field(tear[\![\Delta_1]\!]\grave{\rho}_1)\langle\grave{\rho}_2[\![I]\!]+1\rangle\grave{\sigma}_2\,|\,I:\mathscr{K}[\![\Delta_1]\!]\psi_1[\![I]\!]\}=true$ then $\wedge\{field(tear[\![\Delta_2]\!]\grave{\rho}_0)\langle\grave{\rho}_4[\![I]\!]+1\rangle\grave{\sigma}_4\,|\,I:\mathscr{K}[\![\Delta_2]\!]\psi_0[\![I]\!]\}=true$. Hence $a\langle\acute{\zeta}_1\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3,\grave{\zeta}_1\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3\rangle=a\langle\acute{\zeta}_0\acute{\rho}_4\acute{\upsilon}_4\acute{\sigma}_4,\grave{\zeta}_0\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4\rangle=true$ and, $\hat{\pi}_2$ being an arbitrary suitable pair, $c\hat{\xi}_2\hat{\pi}_1=true$. Since $\hat{\pi}_1$ is subject only to $p\hat{\pi}_1\wedge sewn[\![\Delta_0]\!]0\psi_0\hat{\pi}_0\hat{\pi}_1=true$ and $D[\![\Delta_0]\!]=true$, $a\langle\mathscr{D}[\![\Delta_0]\!]\acute{\zeta}_2\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,\mathscr{D}[\![\mathscr{A}[\![\Delta_0]\!]\psi_0]\!]\grave{\zeta}_2\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0\rangle=true$, as was to be established.

The proof that $D[\![\Delta_2]\!]=true$ is very similar.⊁

Note that the assumption that $\mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_1]\!]$ and $\mathscr{I}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_0]\!]\S\mathscr{K}[\![\Delta_1]\!]$ are lists without repeating members is necessary only to ensure that, in the notation used above, $\psi_1[\![I]\!]=\psi_0[\![I]\!]$ if $I:\mathscr{I}[\![\Delta_2]\!]\S\mathscr{K}[\![\Delta_2]\!]$. Were we not to transform programs by means of ψ this assumption could be eliminated from the proof. Accordingly it will play no part in 2.6.9.

The requirement that $cramped[\![\Delta_2]\!](\lambda I.\langle 2\rangle)$ be *true*, on the other hand, is germane only because we wish to avoid sending local variables out from the body of a recursive declaration in order that we may invoke the valuation $\mathscr{S}$ of 2.2.1. Because 2.5.9 refers to only the valuations of appendix 2 it demands no mention of this requirement, although 2.6.9 will do so.

2.6.7. Lemma.

Let Δ_0 be Δ_1 and...and Δ_n for some $\Delta_1,\ldots,\Delta_n$; if $D[\![\Delta_1]\!]\wedge\ldots\wedge D[\![\Delta_n]\!]=true$ then $D[\![\Delta_0]\!]=true$ whilst if $T[\![\Delta_1]\!]\wedge\ldots\wedge T[\![\Delta_n]\!]=true$ then $T[\![\Delta_0]\!]=true$.

⊲The proof of this resembles 2.3.7 very closely. Again the result holds if $n=1$ and we assume that it holds for all sets of m declarations with $n>m$. For the sake of variety we shall describe why $T[\![\Delta_0]\!]=true$ rather than why $D[\![\Delta_0]\!]=true$. Suppose that *deal* corresponds to evaluation from left to right and that Δ_{n+1} is Δ_2 and...and Δ_n.

Take some ψ_0, $\hat{\zeta}_0$ and $\hat{\pi}_0$ such that $fit\hat{\pi}_0\hat{\pi}_0=true$ and $c\hat{\zeta}_0\hat{\pi}=true$ whenever $sewn[\![\Delta_0]\!]1\psi_0\hat{\pi}_0\hat{\pi}=true$ and $\wedge\{field(tear[\![\Delta_0]\!]\grave{\rho}_0)\langle\grave{\rho}[\![I]\!]\downarrow 1\rangle\grave{\sigma}\mid I:\mathscr{H}[\![t[\![\Delta_0]\!]\psi_0]\!]\}=true$; we shall demonstrate that $c\langle\mathscr{T}[\![\Delta_0]\!]\acute{\zeta}_0,\mathscr{T}[\![t[\![\Delta_0]\!]\psi_0]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$, which will establish the contention that $T[\![\Delta_0]\!]=true$ in view of the arbitrary choice of ψ_0, $\hat{\pi}_0$ and $\hat{\zeta}_0$.

Let $\hat{\pi}_1$ be a pair having $sewn[\![\Delta_1]\!]1\psi_0\hat{\pi}_0\hat{\pi}_1=true$ and $\wedge\{field(tear[\![\Delta_1]\!]\grave{\rho}_0)\langle\grave{\rho}_1[\![I]\!]\downarrow 1\rangle\grave{\sigma}_1\mid I:\mathscr{H}[\![t[\![\Delta_1]\!]\psi_0]\!]\}=true$, and define $\hat{\pi}_2=\langle\langle clip[\![\Delta_1]\!]\acute{\rho}_0\acute{\rho}_1,\acute{\upsilon}_1,\acute{\sigma}_1\rangle,\langle clip[\![\Delta_1]\!]\grave{\rho}_0\grave{\rho}_1,\grave{\upsilon}_1,\grave{\sigma}_1\rangle\rangle$, so that $apt\psi_0\hat{\rho}_2=true$. As we have assumed that $T[\![\Delta_{n+1}]\!]=true$, for any $\hat{\zeta}_1$ such that $c\hat{\zeta}_1\hat{\pi}=true$ whenever $sewn[\![\Delta_{n+1}]\!]1\psi_0\hat{\pi}_2\hat{\pi}=true$ and $\wedge\{field(tear[\![\Delta_{n+1}]\!]\grave{\rho}_2)\langle\grave{\rho}[\![I]\!]\downarrow 1\rangle\grave{\sigma}\mid I:\mathscr{H}[\![t[\![\Delta_{n+1}]\!]\psi_0]\!]\}=true$ we have $c\langle\mathscr{T}[\![\Delta_{n+1}]\!]\acute{\zeta}_1,\mathscr{T}[\![t[\![\Delta_{n+1}]\!]\psi_0]\!]\grave{\zeta}_1\rangle\hat{\pi}_2=true$. We shall show that $\langle\acute{\zeta}_0\circ pick[\![\Delta_1]\!]\langle\acute{\rho}_0,\acute{\rho}_2\rangle,\grave{\zeta}_0\circ pick[\![\Delta_1]\!]\langle\grave{\rho}_0,\grave{\rho}_2\rangle\rangle$ is such a pair $\hat{\zeta}_1$; without loss of generality in doing so we shall take $\hat{\pi}_n$ to be $(\mathbb{P}q_0\times\mathbb{P}q_0)\hat{\pi}_n$ when $0\leq n\leq 2$.

Take any $\hat{\pi}_3$ having $p\hat{\pi}_3\wedge sewn[\![\Delta_{n+1}]\!]1\psi_0\hat{\pi}_2\hat{\pi}_3=true$ and $\wedge\{field(tear[\![\Delta_{n+1}]\!]\grave{\rho}_2)\langle\grave{\rho}_3[\![I]\!]\downarrow 1\rangle\grave{\sigma}_3\mid I:\mathscr{H}[\![t[\![\Delta_{n+1}]\!]\psi_0]\!]\}=true$, and set $\hat{\pi}_4=\langle\langle pick[\![\Delta_1]\!]\langle\acute{\rho}_0,\acute{\rho}_2\rangle\acute{\rho}_3,\acute{\upsilon}_3,\acute{\sigma}_3\rangle,\langle pick[\![\Delta_1]\!]\langle\grave{\rho}_0,\grave{\rho}_2\rangle\grave{\rho}_3,\grave{\upsilon}_3,\grave{\sigma}_3\rangle\rangle$. Again $p\hat{\pi}_4=true$ as $kent\nu\hat{\omega}\hat{\pi}_4\supset kent\nu\hat{\omega}\hat{\pi}_3$ for all ν and $\hat{\omega}$. Furthermore the close analogy between *sewn* and *knit* (together with the fact

that $sewn⟦\Delta_1⟧1\psi_0\hat{\pi}_0\hat{\pi}_1 \wedge sewn⟦\Delta_{n+1}⟧1\psi_0\hat{\pi}_2\hat{\pi}_3 = true$) allows us to presume that $sewn⟦\Delta_0⟧0\psi_0\hat{\pi}_0\hat{\pi}_4 = true$; the detailed proof of this follows the lines laid down in 2.3.7 and lacks independent interest. Before we can conclude that $a(\zeta_0\rho_4\upsilon_4\sigma_4, \grave{\zeta}_0\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4) = true$ we must verify that $\bigwedge\{field(tear⟦\Delta_0⟧\grave{\rho}_0)(\grave{\rho}_4⟦I⟧\downarrow 1)\grave{\sigma}_4 \mid I : \mathscr{H}⟦\iota⟦\Delta_0⟧\psi_0⟧\} = true$.

Accordingly take any $I : \mathscr{H}⟦\iota⟦\Delta_0⟧\psi_0⟧$, so that $I : \mathscr{H}⟦\iota⟦\Delta_1⟧\psi_0⟧$ or $I : \mathscr{H}⟦\iota⟦\Delta_{n+1}⟧\psi_0⟧$. In the former case $q_0(\grave{\rho}_4⟦I⟧\downarrow 1) = q_0(revert\grave{\rho}_2\grave{\rho}_3⟦I⟧\downarrow 2) = revert\grave{\rho}_1\grave{\rho}_3⟦I⟧\downarrow 1 = \grave{\rho}_1⟦I⟧\downarrow 1$ as $sewn⟦\Delta_{n+1}⟧1\psi_0\hat{\pi}_2\hat{\pi}_3 = true$, and so $field(tear⟦\Delta_1⟧\grave{\rho}_0)(\grave{\rho}_4⟦I⟧\downarrow 1)\grave{\sigma}_1 = true$; because $\grave{\rho}_4⟦I⟧\downarrow 1 : V$ this means that $field(tear⟦\Delta_0⟧\grave{\rho}_0)(\grave{\rho}_4⟦I⟧\downarrow 1)\grave{\sigma}_4 = true$. In the latter case $\grave{\rho}_4⟦I⟧\downarrow 1 = \grave{\rho}_3⟦I⟧\downarrow 1$ and $field(tear⟦\Delta_{n+1}⟧\grave{\rho}_2)(\grave{\rho}_4⟦I⟧\downarrow 1)\grave{\sigma}_4 = true$; moreover, as $apt\psi_0\hat{\rho}_0 \wedge sewn⟦\Delta_{n+1}⟧1\psi_0\hat{\pi}_2\hat{\pi}_3 = true$, $torn⟦\Delta_{n+1}⟧\grave{\rho}_0 = true$ and, as $sewn⟦\Delta_1⟧1\psi_0\hat{\pi}_0\hat{\pi}_1 = true$, for any $I' : Ide$

$$tear⟦\Delta_{n+1}⟧\grave{\rho}_2⟦I'⟧ = I' : \mathscr{I}⟦\Delta_1⟧ \S \mathscr{H}⟦\Delta_1⟧ \to tear⟦\Delta_{n+1}⟧(revert\grave{\rho}_0\grave{\rho}_1)⟦I'⟧,$$
$$tear⟦\Delta_{n+1}⟧\grave{\rho}_1⟦I'⟧$$
$$= I' : \mathscr{I}⟦\Delta_1⟧ \S \mathscr{H}⟦\Delta_1⟧ \to tear⟦\Delta_{n+1}⟧\grave{\rho}_0⟦I'⟧,$$
$$tear⟦\Delta_{n+1}⟧\grave{\rho}_0⟦I'⟧.$$

Hence $field(tear⟦\Delta_{n+1}⟧\grave{\rho}_0)(\grave{\rho}_4⟦I⟧\downarrow 1)\grave{\sigma}_4 = true$ whether I is in $\mathscr{H}⟦\iota⟦\Delta_1⟧\psi_0⟧$ or in $\mathscr{H}⟦\iota⟦\Delta_{n+1}⟧\psi_0⟧$.

Consequently for all $\hat{\pi}_3$ such that $p\hat{\pi}_3 \wedge sewn⟦\Delta_{n+1}⟧1\psi_0\hat{\pi}_2\hat{\pi}_3 = true$ and $\bigwedge\{field(tear⟦\Delta_{n+1}⟧\grave{\rho}_2)(\grave{\rho}_3⟦I⟧\downarrow 1)\grave{\sigma}_3 \mid I : \mathscr{H}⟦\iota⟦\Delta_{n+1}⟧\psi_0⟧\} = true$ we have $a(\zeta_1\rho_3\upsilon_3\sigma_3, \zeta_1\rho_3\upsilon_3\sigma_3) = true$. From 2.6.1 it now follows that $c(\mathscr{T}⟦\Delta_{n+1}⟧\grave{\zeta}_1, \mathscr{T}⟦\iota⟦\Delta_{n+1}⟧\psi_0⟧\grave{\zeta}_1)\hat{\pi}_2 = true$. Should $\hat{\pi}_5$ satisfy $p\hat{\pi}_5 \wedge fit\hat{\pi}_5\hat{\pi}_1 = true$ and should we define $\hat{\pi}_6 = (\langle clip⟦\Delta_1⟧\hat{\rho}_0\acute{\rho}_5, \acute{\upsilon}_5, \acute{\sigma}_5\rangle, \langle clip⟦\Delta_1⟧\grave{\rho}_0\grave{\rho}_5, \grave{\upsilon}_5, \grave{\sigma}_5\rangle)$ we shall have successively $yclept\hat{\omega}\hat{\pi}_6 \supset yclept\hat{\omega}\hat{\pi}_5$ for all $\hat{\omega}$, $kent\nu\hat{\omega}\hat{\pi}_6 \supset kent\nu\hat{\omega}\hat{\pi}_5$ for all ν and $\hat{\omega}$, and $p\hat{\pi}_6 \wedge fit\hat{\pi}_6\hat{\pi}_2 = true$. Hence we even know that $c(\mathscr{T}⟦\Delta_{n+1}⟧\acute{\zeta}_1 \circ clip⟦\Delta_1⟧\acute{\rho}_0, \mathscr{T}⟦\iota⟦\Delta_{n+1}⟧\psi_0⟧\grave{\zeta}_1 \circ clip⟦\Delta_1⟧\grave{\rho}_0)\hat{\pi}_1 = true$. This being so for all appropriate $\hat{\pi}_1$ we can conclude that $c(\mathscr{T}⟦\Delta_0⟧\acute{\zeta}_0, \mathscr{T}⟦\iota⟦\Delta_0⟧\psi_0⟧\grave{\zeta}_0)\hat{\pi}_0 = true$ because $T⟦\Delta_1⟧ = true$.⊳

Although this result and the preceding one are couched in terms of $T[\![\Delta]\!]$ it is amply evident that they could be phrased in such a way that 2.6.5 could be applied to them immediately without quantifying over all ψ_0. It is in fact this reading of the results to which 2.6.8 really pertains.

2.6.8. Lemma.

If $T[\![\Delta]\!]=true$ then $T[\![\text{rec } \Delta]\!]=true$ and, when $opts(\mathscr{H}[\![\Delta]\!])=\lambda\psi.false^*$ also, $D[\![\text{rec } \Delta]\!]=true$.

⊰The second part of the statement is the only one which is not too trivial to be worthy of proof. To deal with it take any ψ_0, $\hat{\pi}_0$ and $\hat{\zeta}$ having $opts(\mathscr{H}[\![\Delta]\!])\psi_0=false^*$, $fit\hat{\pi}_0\hat{\pi}_0=true$ and $\bigwedge\{c\hat{\zeta}\hat{\pi}\mid sewn[\![\text{rec } \Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}\}=true$; let $\psi_1=\psi_0[false^*/\mathscr{S}[\![\Delta]\!]][false^*/\mathscr{H}[\![\Delta]\!]]$. Assuming without loss of generality that $p\hat{\pi}_0=true$, as $\mathbb{L}$ is infinite we can set up a proper pair $\langle\acute{\alpha}^*,\grave{\alpha}^*\rangle$ which is $\langle novels(\#\mathscr{S}[\![\Delta]\!])\rho_0\upsilon_0\sigma_0, news(\#\mathscr{S}[\![\Delta]\!])\delta_0\rangle$. We can therefore define

$\hat{\rho}_1=\langle fix(\lambda\rho.\rho_0[\acute{\alpha}^*/\mathscr{S}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho\sigma_1/\mathscr{H}[\![\Delta]\!]]),$

$\quad fix(\lambda\rho.\rho_0[\grave{\alpha}^*/\mathscr{S}[\![\Delta]\!]][\mathscr{S}[\![\acute{\Delta}]\!]\psi_1]\rho\delta_1/\mathscr{H}[\![\Delta]\!]])\rangle$,

$\upsilon_1=\upsilon_0$ and $\sigma_1=\langle updates\acute{\alpha}^*dummy^*\sigma_0, updates\grave{\alpha}^*dummy^*\delta_0\rangle$.

Any $\hat{\pi}$ which satisfies $sewn[\![\Delta]\!]1\psi_1\hat{\pi}_1\hat{\pi}=true$ will also be subject to $sewn[\![\text{rec } \Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}=true$; consequently $c\hat{\zeta}\hat{\pi}=true$ for all such $\hat{\pi}$. As $T[\![\Delta]\!]=true$ once we have convinced ourselves that $p\hat{\pi}_1=true$ we shall be able to deduce that both $a\langle\mathscr{T}[\![\Delta]\!]\acute{\zeta}\rho_1\upsilon_1\sigma_1,\mathscr{T}[\![\acute{\Delta}]\!]\psi_1]\grave{\zeta}\rho_1\upsilon_1\delta_1\rangle$ and $a\langle\mathscr{D}[\![\text{rec } \Delta]\!]\acute{\zeta}\rho_0\upsilon_0\sigma_0,\mathscr{D}[\![\acute{\Delta}[\![\text{rec } \Delta]\!]\psi_0]\!]\grave{\zeta}\rho_0\upsilon_0\delta_0\rangle$ are $true$.

To do this we observe first that

$kent\nu\hat{\omega}\hat{\pi}_1\supset(\hat{\omega}=\langle dummy,dummy\rangle \vee gyven\hat{\omega}\langle\acute{\alpha}^*,\grave{\alpha}^*\rangle$

$\quad \vee gyven\hat{\omega}\langle\mathscr{S}[\![\Delta]\!]\rho_1\sigma_1,\mathscr{S}[\![\acute{\Delta}]\!]\psi_1]\!]\grave{\rho}_1\grave{\sigma}_1\rangle \vee kent\nu\hat{\omega}\hat{\pi}_0)$

for every ν and $\hat{\omega}$. Moreover since $p_0\hat{\pi}_0=true$ and since $site\acute{\omega}\rho_0\upsilon_0\sigma_0\vee area\grave{\omega}\delta_0=false$ for any pair $\hat{\omega}$ such that $gyven\hat{\omega}\langle\acute{\alpha}^*,\grave{\alpha}^*\rangle=true$, we already know that $p_0\hat{\pi}_1=true$.

Consequently if we can establish that $\omega\hat{\Omega}\hat{\pi}_1 = true$ when $gyven\hat{\omega}(\mathscr{S}[\![\Delta]\!]\hat{\rho}_1\hat{\sigma}_1, \mathscr{T}[\![\Delta]\!]\psi_1]\!]\hat{\rho}_1\hat{\sigma}_1) = true$ we shall have $p\hat{\pi}_1 = true$.

For some ν_2 let I be $\mathscr{H}[\![\Delta]\!]\downarrow\nu_2$ and let $\hat{\omega}$ be $(\mathscr{S}[\![\Delta]\!]\hat{\rho}_1\hat{\sigma}_1\downarrow\nu_2, \mathscr{T}[\![\Delta]\!]\psi_1]\!]\hat{\rho}_1\hat{\sigma}_1\downarrow\nu_2)$. Reasoning along the lines of 2.5.8 it is enough to show that if $\hat{\zeta}_0$ and $\hat{\pi}_2$ satisfy $k\hat{\zeta}_0\hat{\pi}_2 = true$, $fit\hat{\pi}_2\hat{\pi}_2 = true$ and $hoten((q_0\times q_0)\hat{\Omega})((\mathbb{U}q_0\times\mathbb{U}q_0)\hat{\rho}_2) = true$ we have $c(\mathscr{T}[\![\Delta]\!]\hat{\zeta}_1, \mathscr{T}[\![\Delta]\!]\psi_1]\!]\grave{\zeta}_1)\hat{\pi}_3 = true$, where

$\hat{\pi}_3 = (\hat{\rho}_1[\hat{\pi}_2/rec], \langle\rangle, \hat{\sigma}_1, \hat{\rho}_1[\hat{\pi}_2/rec], \langle\rangle, \hat{\sigma}_1)$ and

$\hat{\zeta}_1 = (\lambda\rho\upsilon.recur\hat{\zeta}_0\rho(\rho[\![I]\!]\downarrow 1 \mid E),$

$\lambda\rho\upsilon\sigma.(\lambda\pi'.\grave{\zeta}_0(\pi'\downarrow 1)((\rho[\![I]\!]\downarrow 1)\,\S\,\pi'\downarrow 2)(\pi'\downarrow 3))(\rho[\![rec]\!]\downarrow 1))$.

Because $T[\![\Delta]\!] = true$ even this reduces to verifying that for $\hat{\pi}_3$ and $\hat{\zeta}_1$ defined in this way $c\hat{\zeta}_1\hat{\pi}_4 = true$ whenever $\hat{\pi}_4$ satisfies $sewn[\![\Delta]\!]1\psi_1\hat{\pi}_3\hat{\pi}_4 \wedge field\hat{\rho}_1(\hat{\rho}_4[\![I]\!]\downarrow 1)\hat{\sigma}_4 = true$.

Accordingly take any such $\hat{\pi}_4$ together with some $\hat{\pi}_5$ such that $p\hat{\pi}_5 \wedge fit\hat{\pi}_5\hat{\pi}_4 = true$; define $\hat{\varepsilon}_6$, $\hat{\pi}_6$ and $\hat{\pi}_7$ to be $(\hat{\rho}_5[\![I]\!]\downarrow 1, \hat{\rho}_5[\![I]\!]\downarrow 1)$, $(\hat{\rho}_5[\![rec]\!]\downarrow 1, \hat{\rho}_5[\![rec]\!]\downarrow 1)$ and $((\hat{\rho}_6, (\hat{\varepsilon}_6)\,\S\,\hat{\upsilon}_6, replace\hat{\rho}_6\hat{\upsilon}_6\hat{\sigma}_6\hat{\sigma}_5), (\hat{\rho}_6, (\grave{\varepsilon}_6)\,\S\,\grave{\upsilon}_6, \grave{\sigma}_6))$. Then $(\hat{\zeta}_1\hat{\rho}_5\hat{\upsilon}_5\hat{\sigma}_5, \grave{\zeta}_1\grave{\rho}_5\grave{\upsilon}_5\grave{\sigma}_5) = (\hat{\zeta}_0\hat{\rho}_7\hat{\upsilon}_7\hat{\sigma}_7, \grave{\zeta}_0\grave{\rho}_7\grave{\upsilon}_7\grave{\sigma}_7)$ and as $set\hat{\pi}_7\hat{\pi}_2 = true$ the result will be proven if $p\hat{\pi}_7 = true$. We shall demonstrate this by showing that for all $\nu > 2$ and $\hat{\omega}$ $kent\nu\hat{\Omega}\hat{\pi}_7 \supset kent0\hat{\Omega}\hat{\pi}_5 \wedge (\nu = 1 \wedge \hat{\omega}: L \rightarrow kent1\hat{\Omega}\hat{\pi}_6, true)$. Just as in 2.5.8 we have $kent\nu\hat{\Omega}\hat{\pi}_7 \supset \bigvee\{seen\nu\nu_1\hat{\Omega}\hat{\varepsilon}_6\hat{\pi}_7 \mid \nu_1 : N\} \vee kent\nu\hat{\Omega}\hat{\pi}_6$, since the proof given before requires only that $access\hat{\Omega}_1\hat{\pi}_7 = access\hat{\Omega}_1\hat{\pi}_6$ when $kent1\hat{\Omega}_1\hat{\pi}_6 = true$, which is still the case even if $\hat{\sigma}_7$ is defined to be $\hat{\sigma}_6$. Thus it remains to be established that $\bigvee\{seen\nu\nu_1\hat{\Omega}\hat{\varepsilon}_6\hat{\pi}_7 \mid \nu_1 : N\} \supset kent0\hat{\Omega}\hat{\pi}_5 \wedge (\nu > 0 \wedge \hat{\omega}: L \rightarrow kent1\hat{\Omega}\hat{\pi}_6, true)$. It is this part of the proof which invokes the assumption that $field\hat{\rho}_1(\hat{\rho}_4[\![I]\!]\downarrow 1)\hat{\sigma}_4 = true$.

Because $sewn[\![\Delta]\!]1\psi_1\hat{\pi}_3\hat{\pi}_4 = true$, $\hat{\rho}_4[\![I]\!]\downarrow 1 : V$; hence as $\psi_1[\![I]\!]\downarrow 1 = false$ and $apt\psi_1\hat{\rho}_4 = true$ $\hat{\rho}_4[\![I]\!]\downarrow 1 : V$. Moreover $fit\hat{\pi}_5\hat{\pi}_4 = true$, so we can infer that $q_0\grave{\varepsilon}_6 = \mathbb{U}q_0\hat{\rho}_4[\![I]\!]\downarrow 1$ and that

$field\grave{\rho}_1\langle\grave{\varepsilon}_6\rangle\grave{\sigma}_5 = true$. As $hoten((q_0\times q_0)\hat{\omega})((\mathfrak{U}q_0\times\mathfrak{U}q_0)\hat{\rho}_2) = true$, $\wedge\{plot\alpha\grave{\rho}_2() empty \vee \sim plot\alpha\grave{\rho}_1() empty \mid \alpha: L\} = true$ and from 2.6.1 $field\grave{\rho}_2\langle\grave{\varepsilon}_6\rangle\grave{\sigma}_5 = true$. Finally, as $fit\hat{\pi}_5\hat{\pi}_4 = true$, $\mathfrak{P}q_0\hat{\pi}_5 = \mathfrak{P}q_0(\grave{\rho}_4[\![rec]\!]\downarrow 1)$, and, as $seen[\![\Delta]\!]1\psi_1\hat{\pi}_3\hat{\pi}_4 = true$, $\mathfrak{P}q_0(\grave{\rho}_4[\![rec]\!]\downarrow 1) = \mathfrak{P}q_0\hat{\pi}_2$; in particular $\mathfrak{U}q_0\grave{\rho}_2 = \mathfrak{U}q_0\grave{\rho}_6$ and by 2.4.3 $\wedge\{plot\alpha\grave{\rho}_6() empty \vee \sim plot\alpha arid\langle\grave{\varepsilon}_6\rangle empty \mid \alpha: L\} = true$.

Defining $\hat{\pi}_8$ to be $\langle\langle arid, \langle\acute{\varepsilon}_6\rangle, empty\rangle, \langle arid, \langle\grave{\varepsilon}_6\rangle, empty\rangle\rangle$, we now know that when $kent1\hat{\omega}\hat{\pi}_8 = true$ for some $\hat{\omega}$ with $\grave{\omega}: L$ we have $plot\grave{\omega}\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6 = true$. Since $p\hat{\pi}_5 = true$ and $hoten\hat{\pi}_6\hat{\rho}_5 = true$ we can even assert that $kent1\langle\acute{\varepsilon},\grave{\omega}\rangle\hat{\pi}_6 = true$ for some $\acute{\varepsilon}$; indeed, $kent1\hat{\omega}\hat{\pi}_8$ being $true$, $kent1\hat{\omega}\hat{\pi}_5$ must be $true$, and, $kent1\langle\acute{\varepsilon},\grave{\omega}\rangle\hat{\pi}_6$ being $true$, $kent0\langle\acute{\varepsilon},\grave{\omega}\rangle\hat{\pi}_5$ must be $true$, so that $\acute{\varepsilon} = \acute{\omega}$ as $p_0\hat{\pi}_6 = true$. Hence whenever $kent1\hat{\omega}\hat{\pi}_8 = true$ and $\grave{\omega}: L$ we have $kent1\hat{\omega}\hat{\pi}_6 = true$.

Assume that for some particular ν_1 and all $\nu_0 < 2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ we know that

$seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_8 \supset kent0\hat{\omega}_0\hat{\pi}_5 \wedge (\nu_0 > 0 \wedge \grave{\omega}_0: L \rightarrow kent1\hat{\omega}_0\hat{\pi}_6, true)$;

the paragraph above shows this to be so when $\nu_1 < 1$. Let $\nu_0 < 2$, $\hat{\omega}_0$ and $\hat{\omega}_1$ be such that $seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_8 = true$. If $\grave{\omega}_1: L$ then $\acute{\omega}_1: E$ as $\omega_0\hat{\omega}_1\hat{\pi}_5 = true$ ($kent0\hat{\omega}_1\hat{\pi}_5$ being $true$) and $kent1\hat{\omega}_1\hat{\pi}_6 = true$ so that $(\acute{\omega}_1: L \rightarrow plot\acute{\omega}_1\acute{\rho}_6\acute{\upsilon}_6\acute{\sigma}_6, true) = true$ and $access\hat{\omega}_1\hat{\pi}_7 = access\hat{\omega}_1\hat{\pi}_6$; under these circumstances $seen\nu_0\nu_1(access\hat{\omega}_1\hat{\pi}_6)\hat{\pi}_7 \wedge kent1(access\hat{\omega}_1\hat{\pi}_6)\hat{\pi}_6 = true$ and $kent\nu_0\hat{\omega}_0\hat{\pi}_6 = true$ by the argument used in 2.5.8. If $\grave{\omega}_1: V$, $\grave{\omega}_1: G$, $\grave{\omega}_1: J$ or $\grave{\omega}_1: P$ we can apply the usual techniques and the induction hypothesis to confirming that $kent0\hat{\omega}_0\hat{\pi}_7 = true$ and, if $\nu_0 > 0$ and $\grave{\omega}_0: L$, that $kent1\hat{\omega}_0\hat{\pi}_6 = true$.

Hence for all ν and $\hat{\omega}$ we have

$$\vee\{seen\nu\nu_1\hat{\omega}\hat{\varepsilon}_6\hat{\pi}_7 \mid \nu_1: N\} \supset \vee\{\vee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_7 \wedge yclept\hat{\omega}_1\hat{\pi}_8 \mid \nu_1: N\} \mid \hat{\omega}_1: W\times W\}$$
$$\supset \vee\{\vee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_8 \mid \nu_1: N\} \mid \hat{\omega}_1: W\times W\}$$
$$\supset kent0\hat{\omega}\hat{\pi}_5 \wedge (\nu > 0 \wedge \grave{\omega}: L \rightarrow kent1\hat{\omega}\hat{\pi}_6, true),$$

and, more generally,

$kent\nu\hat{\omega}\hat{\pi}_7 \supset kent0\hat{\omega}\hat{\pi}_5 \wedge (\nu>0 \wedge \grave{\omega}:L\rightarrow kent1\hat{\omega}\hat{\pi}_6, true)$. As $p_0\hat{\pi}_5$, $\bigwedge\{w\hat{\omega}\hat{\pi}_5 \mid kent0\hat{\omega}\hat{\pi}_5\}$ and $kent0\hat{\pi}_6\hat{\pi}_5$ are all *true*, $p_0\hat{\pi}_7$ and $\bigwedge\{w\hat{\omega}\hat{\pi}_7 \mid kent0\hat{\omega}\hat{\pi}_7\}$ are *true*; thus $p\hat{\pi}_7=true$. Retracing the steps of the proof, it follows that $c\langle\mathscr{T}[\![\Delta]\!]\acute{\zeta}_1, \mathscr{T}[\![\ell]\!][\![\Delta]\!]\psi_1[\![\grave{\zeta}_1\rangle\hat{\pi}_3=true$ for all suitable $\hat{\pi}_1$, $\hat{\pi}_2$ and $\hat{\zeta}_0$ inducing $\hat{\pi}_3$ and $\hat{\zeta}_1$ in the manner above and thus that $w\hat{\omega}\hat{\pi}_1=true$ when $gyven\hat{\omega}\langle\mathscr{S}[\![\Delta]\!]\acute{\rho}_1\acute{\sigma}_1, \mathscr{S}[\![\ell]\!][\![\Delta]\!]\psi_1[\![\grave{\rho}_1\grave{\sigma}_1\rangle=true$. This being so whenever ψ_1 and $\hat{\pi}_1$ are defined in terms of any ψ_0 and $\hat{\pi}_0$ having $opts(\mathscr{K}[\![\Delta]\!])\psi_0=false^*$ and $fit\hat{\pi}_0\hat{\pi}_0=true$, the lemma must be valid.⊳

It is in the proof above that we require the rather curious definition of $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}$ when $\hat{\omega}_1:G\times G$. In the sixth paragraph above we made use of the fact that if I′ is such that $\grave{\rho}_2[\![I']\!]\downarrow 1$ is in G then any α which is subject to $plot\alpha(\grave{\rho}_2[\![I']\!]\downarrow 1\downarrow 2)\langle\rangle empty=true$ has $plot\alpha\grave{\rho}_2\langle\rangle empty=true$ also. We could not, however, simply modify the algorithm in 2.1.6 so that when $\hat{\omega}_3:G\times G$ $seen1(\nu_1+1)\hat{\omega}_2\hat{\omega}_3\hat{\pi}$ would be *true* only if $seen1\nu_1\hat{\omega}_2\hat{\omega}_4\langle\langle arid,\langle\rangle, empty\rangle, \langle arid,\langle\rangle, empty\rangle\rangle$ were *true* for some $\hat{\omega}_4$ having $hoten\hat{\omega}_4\langle\acute{\omega}_3\downarrow 2, \grave{\omega}_3\downarrow 2\rangle=true$: in the eighth paragraph above we demand that $kent1\hat{\omega}_1\hat{\pi}_6 \supset kent1(access\hat{\omega}_1\hat{\pi}_6)\hat{\pi}_6$, which would not be correct were this form to be taken.

Even in a practical implementation we cannot regard this aspect of the tracing procedure as redundant. If $\grave{\rho}_2[\![I']\!]\downarrow 1$ is in G as a consequence of a declaration of I′ earlier in the program we may presume that any locations reached by passing through the result returned by $\grave{\rho}_2[\![I']\!]\downarrow 1$ will have been already adjoined to the area of store during the declaration. If I′ represents a library function, however, no explicit declaration will have been given and unless $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}$ takes its form from 2.1.6 the definition of $p_0\hat{\pi}_2$ may lead to anomalies when I′ is activated.

2.6.9. Theorem.

The meanings of a Mal program provided by *novel* store semantics and by *new* store semantics are comparable so long as every constituent of the form rec Δ has $cramped[\![\Delta]\!](\lambda I.\langle 2\rangle)=true$.

⊲If we suppose that $opt=\lambda I\psi.false$ we can dispense with the requirement in 2.6.6 that certain constituents of the form Δ_0 within Δ_1 are such that $\mathscr{I}[\![\Delta_1]\!]$ has no elements in common with $\mathscr{H}[\![\Delta_0]\!]$ and $\mathscr{H}[\![\Delta_1]\!]$ has no elements in common with $\mathscr{I}[\![\Delta_0]\!]\S\mathscr{H}[\![\Delta_0]\!]$. A structural induction using 2.6.2 and the definition of *cramped* given in 1.5.3 thus establishes that the meanings are comparable with respect to the predicate a of 2.6.1.⊳

2.7. An extension to cover recursion.

2.7.1. Removing continuations.

Although 2.5.9 ensures that many of the transformations of 1.4.6 can be performed on a program without modifying its meaning, we have yet to establish the equivalence of recursive declarations by incidence and those by reference. In this section we shall achieve this end by a rather devious route which will involve switching between two types of semantic equation by means of 2.3.9. Taking over the notions of 1.5.4 we shall prove that $D[\![\,\mathsf{rec}\ \Delta\,]\!]=\mathit{true}$ for certain declarations outside the class we have already catered for. The predicates underlying this assertion will be those of 2.6.1 (except that $\mathit{area}\acute{\alpha}_m\acute{\sigma}$ and $\mathit{kent1}\hat{\omega}_0\hat{\pi}$ will supersede $\mathit{site}\acute{\alpha}_m\beta\acute{\upsilon}\acute{\sigma}$ and $\mathit{kent}\upsilon\hat{\omega}_0\hat{\pi}$ in $p_0\hat{\pi}$), but all programs will be deemed to be evaluated using *new* instead of *novel*.

Conjugate valuations can be set up for store semantics in much the same way as for standard semantics; moreover the only property of their forerunners which they do not have is 1.5.8, which depends on the fact that free variable lists are not split off from the code in standard function closures. It is, however, precisely this property that allows us to replace members of G in the environment by members of V, thereby providing the rationale for *apt* in 1.4.6. Accordingly we shall need to compare these types of conjugate in order to transfer this property to store semantics; 2.7.3 will effect this comparison while 2.7.5 will outline the proof that two related recursive declarations give rise to equivalent state vectors.

Again we take the conjugates of $\mathscr{F}$ and $\mathscr{G}$ to be $\mathscr{F}$ and $\mathscr{G}$, writing them as $\P\mathscr{F}$ and $\P\mathscr{G}$. Because we do not consider labels we can continue to conflate $\P\mathscr{C}$ and $\P\mathscr{E}$, which this time are in Exp→P→P. We could in fact eliminate the stack component from all state vectors to which these valuations are applied, as the

arguments of 2.1.1 which led to its introduction become nugatory when jumps are entirely removed. To avoid confusion with the standard conjugates we retain it and demand that $\P\mathscr{G}[\![I]\!]$, $\P\mathscr{G}[\![\Phi]\!]$ and $\P\mathscr{G}[\![B]\!]$ be $\lambda\langle\rho,\upsilon,\sigma\rangle.\langle\rho,\langle\rho[\![I]\!]\downarrow 1\,|\,E\rangle\,\S\upsilon,\sigma\rangle$, $\lambda\langle\rho,\upsilon,\sigma\rangle.\langle\rho,\langle\mathscr{F}[\![\Phi]\!]\rho\rangle\,\S\upsilon,\sigma\rangle$ and $\lambda\langle\rho,\upsilon,\sigma\rangle.\langle\rho,\langle\mathscr{B}[\![B]\!]\rangle\,\S\upsilon,\sigma\rangle$ for any I:Ide, Φ:Abs and B:Bas. For every E:Exp we define $\P\mathscr{L}[\![E]\!]$ and $\P\mathscr{A}[\![E]\!]$ to be

$(\lambda\langle\rho,\upsilon,\sigma\rangle.\upsilon\downarrow 1:\mathsf{L}\rightarrow\langle\rho,\upsilon,\sigma\rangle,\langle\rho,\langle new\sigma\rangle\,\S\upsilon\dagger 1,update(new\sigma)(\upsilon\downarrow 1)\sigma\rangle)\circ\P\mathscr{E}[\![E]\!]$

and

$(\lambda\langle\rho,\upsilon,\sigma\rangle.(\lambda\varepsilon.\varepsilon:\mathsf{L}\rightarrow(area\varepsilon\sigma\rightarrow\langle\rho,\langle hold\varepsilon\sigma\rangle\,\S\upsilon\dagger 1,\sigma\rangle,\tau),\langle\rho,\upsilon,\sigma\rangle)(\upsilon\downarrow 1))\circ\P\mathscr{E}[\![E]\!]$

respectively.

It would be possible to avoid returning an environment as a result on applying $\P\mathscr{E}$ or $\P\mathscr{G}$, but this liberty is not open to us when we consider $\P\mathscr{D}$ and $\P\mathscr{T}$, which are now in Dec→P→P. We take $\P\mathscr{D}[\![I=E]\!]$ to be $(\lambda\langle\rho,\upsilon,\sigma\rangle.\langle\rho[\upsilon\downarrow 1/I],\upsilon\dagger 1,\sigma\rangle)\circ\P\mathscr{E}[\![E]\!]$ and $\P\mathscr{D}[\![I==E]\!]$ to be $(\lambda\langle\rho,\upsilon,\sigma\rangle.\langle\rho[\upsilon\downarrow 1/I],\upsilon\dagger 1,\sigma\rangle)\circ\P\mathscr{A}[\![E]\!]$, and adopt similar equations, based on those of appendix 2, for the other forms of declaration. Consequently $\P\mathscr{D}[\![\Delta_0\ \text{within}\ \Delta_1]\!]$ and $\P\mathscr{T}[\![\Delta_0\ \text{within}\ \Delta_1]\!]$ become

$\lambda\pi'.(\lambda\pi''.\langle trim[\![\Delta_1]\!](\pi'\downarrow 1)(\pi''\downarrow 1)\rangle\,\S\pi''\dagger 1)(\P\mathscr{D}[\![\Delta_1]\!](\P\mathscr{D}[\![\Delta_0]\!]\pi'))$ and

$\lambda\pi'.(\lambda\pi''.\langle trim[\![\Delta_1]\!](\pi'\downarrow 1)(\pi''\downarrow 1)\rangle\,\S\pi''\dagger 1)(\P\mathscr{D}[\![\Delta_1]\!](\P\mathscr{T}[\![\Delta_0]\!]\pi'))$,

while $\P\mathscr{D}[\![\text{rec}\ \Delta]\!]$ and $\P\mathscr{T}[\![\text{rec}\ \Delta]\!]$ become

$\lambda\langle\rho,\upsilon,\sigma\rangle.(\lambda\alpha^*.(\lambda\sigma'.(\lambda\rho'.\alpha^*:\mathsf{E}\rightarrow\P\mathscr{T}[\![\Delta]\!]\langle\rho',\upsilon,\sigma'\rangle,\tau)$

$(fix(\lambda\rho''.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]]\mathscr{D}[\![\Delta]\!]\rho''\sigma'/\mathscr{H}[\![\Delta]\!]])))$

$(updates\alpha^*dummy^*\sigma))(news(\#\mathscr{I}[\![\Delta]\!])\sigma)$

and $\P\mathscr{T}[\![\Delta]\!]$ respectively. To simplify our notation we introduce a member of Dec→U→S→E*, $\P\mathscr{S}$, such that for any Δ:Dec $\P\mathscr{S}[\![\Delta]\!]$ is

$\lambda\rho\sigma.fix(\lambda\phi I^*.I^*=\langle\rangle\rightarrow\langle\rangle,\langle(\P\mathscr{T}[\![\Delta]\!]\langle\rho,\langle\rangle,\sigma\rangle\downarrow 1)[\![I^*\downarrow 1]\!]\downarrow 1\rangle\,\S\phi(I^*\dagger 1))(\mathscr{H}[\![\Delta]\!])$.

For brevity we define $crowded$:[Exp+Dec]→[Ide↛B*]→T, so that $crowded[\![E]\!]\psi=true$ if $crushed[\![E]\!]\psi=true$ and each constituent of E of the form Δ_0 inside E_0 satisfies $crowded[\![\Delta_0]\!]\psi=true$, while

crowded⟦Δ⟧ψ=*true* if *crushed*⟦Δ⟧ψ=*true*, Δ contains no recursive declarations by incidence, no identifier I having ψ⟦I⟧↓1=2 is in $\mathscr{F}$⟦Δ⟧§$\mathscr{K}$⟦Δ⟧, every constituent Δ_1 of Δ obeys *crowded*⟦Δ_1⟧ψ=*true* and every expression E_1 embedded in Δ satisfies *crowded*⟦E_1⟧ψ=*true*.

2.7.2. Proposition.

Let ψ and ρ be such that for all I:Ide ψ⟦I⟧↓1=2 if ρ⟦I⟧↓1:G. For every proper υ and σ and every E:Exp subject to *crushed*⟦E⟧ψ=*true* either λζ.$\mathscr{E}$⟦E⟧ζρυσ is improper or ¶$\mathscr{E}$⟦E⟧⟨ρ,υ,σ⟩ is proper and λζ.$\mathscr{E}$⟦E⟧ζρυσ=λζ.(λ⟨ρ',υ',σ'⟩.ζρ'υ'σ')(¶$\mathscr{E}$⟦E⟧⟨ρ,υ,σ⟩); analogous conclusions hold for $\mathscr{L}$, $\mathscr{R}$ and $\mathscr{G}$ also. For every proper υ and σ and every Δ:Dec subject to *crushed*⟦Δ⟧ψ=*true* either λζ.$\mathscr{D}$⟦Δ⟧ζρυσ is improper or ¶$\mathscr{D}$⟦Δ⟧⟨ρ,υ,σ⟩ is proper, λζ.$\mathscr{D}$⟦Δ⟧ζρυσ=λζ.(λ⟨ρ',υ',σ'⟩.ζρ'υ'σ')(¶$\mathscr{D}$⟦Δ⟧⟨ρ,υ,σ⟩) and for all I:Ide (¶$\mathscr{D}$⟦Δ⟧⟨ρ,υ,σ⟩↓1)⟦I⟧↓1:G only if ρ⟦I⟧↓1:G and I is not a member of $\mathscr{F}$⟦Δ⟧§$\mathscr{K}$⟦Δ⟧; analogous conclusions hold for $\mathscr{T}$.

⊲The proof involves a structural induction which differs from that of 1.5.5 solely because the valuations are those appropriate to store semantics and therefore lie in Exp→P→P and Dec→P→P instead of Exp→U→S→[E×S] and Dec→U→S→[U×S]. As it has absolutely no additional interest it can safely be left to the imagination.⊳

2.7.3. Proposition.

Let ψ, υ, $\hat{\rho}$ and $\hat{\sigma}$ be proper entities such that in the notation of 2.2.7 $u\hat{\rho}\wedge s\hat{\sigma}$=*true* and for all I:Ide ψ⟦I⟧↓1=2 if $\hat{\rho}$⟦I⟧↓1:G. For every E:Exp satisfying *crushed*⟦E⟧ψ=*true* and *rent*⟦E⟧$\hat{\rho}$=*true* either λκ.$\mathscr{E}$⟦E⟧$\hat{\rho}$κσ and λζ.$\mathscr{E}$⟦E⟧ζ$\hat{\rho}$υ$\hat{\sigma}$ are both ⊥ or ⊤ or $\hat{\rho}_0=\hat{\rho}$, $\hat{\upsilon}_0\dagger 1=\upsilon$, $e\langle\hat{\varepsilon}_0,\hat{\upsilon}_0\downarrow 1\rangle$=*true* and $s\hat{\sigma}_0$=*true*, where $\langle\hat{\varepsilon}_0,\hat{\sigma}_0\rangle$=¶$\mathscr{E}$⟦E⟧$\hat{\rho}\sigma$ and $\hat{\pi}_0$=¶$\mathscr{E}$⟦E⟧⟨$\hat{\rho}$,υ,$\hat{\sigma}$⟩; analogous conclusions hold

for $\mathscr{L}$, $\mathscr{R}$ and $\mathscr{G}$ also. For every Δ:Dec satisfying $crushed[\![\Delta]\!]\psi=true$ and $rent[\![\Delta]\!]\hat{\rho}=true$ either $\lambda\chi.\mathscr{D}[\![\Delta]\!]\hat{\rho}\chi\hat{\sigma}$ and $\lambda\zeta.\mathscr{D}[\![\Delta]\!]\zeta\hat{\rho}\upsilon\hat{\theta}$ are both $\perp$ or $\top$ or $knit[\![\Delta]\!]0\hat{\rho}\hat{\rho}_0=true$, $\hat{\upsilon}_0=\upsilon$, $u\hat{\rho}_0=true$ and $s\hat{\sigma}_0=true$, where $\langle\hat{\rho}_0,\hat{\sigma}_0\rangle=\P\mathscr{D}[\![\Delta]\!]\hat{\rho}\hat{\sigma}$ and $\hat{\pi}_0=\P\mathscr{D}[\![\Delta]\!]\langle\hat{\rho},\upsilon,\theta\rangle$. For every Δ:Dec satisfying $crushed[\![\Delta]\!]\psi=true$ and $torn[\![\Delta]\!]\hat{\rho}=true$ either $\lambda\chi.\mathscr{T}[\![\Delta]\!]\hat{\rho}\chi\hat{\sigma}$ and $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\hat{\rho}\upsilon\hat{\theta}$ are both $\perp$ or $\top$ or $knit[\![\Delta]\!]1\hat{\rho}\hat{\rho}_0=true$, $\hat{\upsilon}_0=\upsilon$, $u\hat{\rho}_0=true$ and $s\hat{\sigma}_0=true$, where $\langle\hat{\rho}_0,\hat{\upsilon}_0\rangle=\P\mathscr{T}[\![\Delta]\!]\hat{\rho}\hat{\sigma}$ and $\hat{\pi}_0=\P\mathscr{T}[\![\Delta]\!]\langle\hat{\rho},\upsilon,\theta\rangle$.

≮Because the conjugate equations are built up by removing the continuations from the usual versions but leaving the primitive functions like *update* intact, lemmata such as 2.3.5 can be transferred wholesale to the present situation. Indeed 2.3.2 requires almost no alterations whatever, since we have chosen to identify $\P\mathscr{F}$ with $\mathscr{F}$ and $\P\mathscr{B}$ with $\mathscr{B}$. Thus the proof of 2.3.9 in effect includes that of this result, which we shall dwell on no longer.≯

Suppose that ψ, υ, $\hat{\rho}$ and $\hat{\sigma}$ satisfy the conditions of the proposition and let Δ:Dec be such that $torn[\![\Delta]\!]\hat{\rho}=true$. When $\lambda\chi.\mathscr{T}[\![\Delta]\!]\hat{\rho}\chi\hat{\sigma}$ is improper so is $\P\mathscr{T}[\![\Delta]\!]\hat{\rho}\hat{\sigma}$ and when $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\hat{\rho}\upsilon\hat{\theta}$ is improper so is $\P\mathscr{T}[\![\Delta]\!]\langle\hat{\rho},\langle\rangle,\theta\rangle$. Writing $\hat{\rho}_0$ for $\langle\P\mathscr{T}[\![\Delta]\!]\hat{\rho}\hat{\sigma}\downarrow1,\P\mathscr{T}[\![\Delta]\!]\hat{\rho}\langle\rangle\sigma\downarrow1\rangle$, either $\hat{\rho}_0$ and $\hat{\rho}_0$ take the same improper value or $knit[\![\Delta]\!]1\psi\hat{\rho}\hat{\rho}_0\wedge u\hat{\rho}_0=true$; for both possibilities $d\langle\hat{\rho}_0[\![I]\!]\downarrow1,\hat{\rho}_0[\![I]\!]\downarrow1\rangle=true$ when $I:\mathscr{H}[\![\Delta]\!]$. Now assume that $\hat{\rho}$ is $\langle\hat{\rho}_1[\perp^*/\mathscr{H}[\![\Delta]\!]],\hat{\rho}_1[\perp^*/\mathscr{H}[\![\Delta]\!]]\rangle$ for some $\hat{\rho}_1$, and define

$fun=\lambda\nu.\langle\hat{\rho}_1[\nu=0\to\perp^*,\P\mathscr{S}[\![\Delta]\!](fun(\nu-1)\downarrow1)\hat{\sigma}/\mathscr{H}[\![\Delta]\!]],$

$\hat{\rho}_1[\nu=0\to\perp^*,\P\mathscr{S}[\![\Delta]\!](fun(\nu-1)\downarrow2)\hat{\theta}/\mathscr{H}[\![\Delta]\!]]\rangle$.

If $u(fun\nu)=true$ for some ν, when $1\le\nu'\le\#\mathscr{H}[\![\Delta]\!]$ we know that $d\langle\P\mathscr{S}[\![\Delta]\!](fun\nu\downarrow1)\hat{\sigma}\downarrow\nu',\P\mathscr{S}[\![\Delta]\!](fun\nu\downarrow2)\hat{\theta}\downarrow\nu'\rangle=true$ and $u(fun(\nu+1))=true$. Since $u(fun0)=true$ we can confirm by induction that $u\langle fix(\lambda\rho.\hat{\rho}_1[\P\mathscr{S}[\![\Delta]\!]\rho\hat{\sigma}/\mathscr{H}[\![\Delta]\!]]),fix(\lambda\rho.\hat{\rho}_1[\P\mathscr{S}[\![\Delta]\!]\rho\hat{\theta}/\mathscr{H}[\![\Delta]\!]])\rangle=true$. This would not be the case were we to substitute Ide↦D* for Ide↦D°* in U, as may be seen by taking Δ to be n==1.

2.7.4. Lemma.

Suppose that $\hat{\pi}_0$, $\hat{\pi}_1$, α^*, I^* and ψ_0 satisfy $\hat{\pi}_1=\langle \hat{\pi}_0,\langle \hat{\rho}_0,\hat{\upsilon}_0,\langle \lambda\alpha.\langle area\alpha\delta_0\wedge\sim\alpha:\alpha^*,hold\alpha\delta_0\rangle\rangle \S\delta_0\dagger 1\rangle\rangle$, $\bigwedge\{\omega_0\hat{\omega}\hat{\pi}_1\vee(\acute{\omega}:G\wedge\grave{\omega}:\alpha^*)\,|\,kent0\hat{\omega}\hat{\pi}_1\}=true$, $\bigwedge\{\acute{\omega}_0:L\wedge\acute{\omega}_0=\acute{\omega}_1\rightarrow area\acute{\omega}_0\acute{\sigma}_0\wedge\grave{\omega}_0=\grave{\omega}_1,true\,|\,kent1\hat{\omega}_0\hat{\pi}_1\wedge kent1\hat{\omega}_1\hat{\pi}_1\}=true$, $\bigwedge\{\grave{\omega}_0:L\wedge\grave{\omega}_0=\grave{\omega}_1\rightarrow area\grave{\omega}_0\delta_0\wedge(\grave{\omega}_0=\grave{\omega}_1\vee\grave{\omega}_1:\alpha^*),true\,|\,kent1\hat{\omega}_0\hat{\pi}_1\wedge kent1\hat{\omega}_0\hat{\pi}_1\}=true$, $apt\psi_0\langle \acute{\rho}_1[dummy^*/I^*],\hat{\rho}_1\rangle=true$ and $fit\hat{\pi}_0\hat{\pi}_0=true$; presume also that $\acute{\sigma}_1$ and δ_1 are of finite area and that $\alpha^*=\langle \hat{\rho}_1[\![I^*\downarrow 1]\!]\downarrow 1,\ldots,\hat{\rho}_1[\![I^*\downarrow\#I^*]\!]\downarrow 1\rangle$. For some ψ and E:Exp suppose that $crowded[\![E]\!]\psi=true$, that $\psi[\![I]\!]\downarrow 1=2$ for all I having $\acute{\sigma}_0[\![I]\!]\downarrow 1:G$ or I:I*, and that $rent[\![E]\!]\psi_0=true$. Either $\lambda\zeta.\mathscr{E}[\![E]\!]\zeta\acute{\rho}_0\xi_0\acute{\sigma}_0$ and $\lambda\zeta.\mathscr{E}[\![\grave{\ }[\![E]\!]\psi_0]\!]\zeta\hat{\rho}_0\hat{\upsilon}_0\delta_0$ are proper or they are both $\perp$ or $\top$; moreover the choice of alternative does not depend on $\hat{\upsilon}_0$ or on the value of $\acute{\rho}_0[\![I]\!]\downarrow 1$ when $\psi[\![I]\!]\downarrow 1=2$. If the former situation obtains then, writing $\hat{\pi}_2=\langle \P\mathscr{E}[\![E]\!]\hat{\pi}_0,\P\mathscr{E}[\![\grave{\ }[\![E]\!]\psi_0]\!]\hat{\pi}_0\rangle$ and $\hat{\pi}_3=\langle \hat{\pi}_2,\langle \hat{\rho}_2,\hat{\upsilon}_2,\langle \lambda\alpha.\langle area\alpha\delta_2\wedge\sim\alpha:\alpha^*,hold\alpha\delta_2\rangle\rangle \S\delta_2\dagger 1\rangle\rangle$, $set\hat{\pi}_2\hat{\pi}_0=true$, $\hat{\upsilon}_2\downarrow 1$ is not a member of α^* and $\hat{\pi}_3$ obeys the constraints imposed on $\hat{\pi}_1$ above; in addition, if $\bigwedge\{w\hat{\omega}\hat{\pi}_n\,|\,kent0\hat{\omega}\hat{\pi}_n\}$ and $\bigwedge\{\acute{\omega}_0=\acute{\omega}_1\vee\sim\grave{\omega}_0:L\vee\sim\grave{\omega}_0=\grave{\omega}_1\,|\,kent1\hat{\omega}_0\hat{\pi}_n\wedge kent1\hat{\omega}_1\hat{\pi}_n\}$ are both *true* when n is 1 they are both *true* when n is 3. Similar conclusions pertain to any Δ:Dec having $crowded[\![\Delta]\!]\psi=true$ and $rent[\![\Delta]\!]\psi_0=true$, except in that when $\hat{\pi}_2=\langle \P\mathscr{D}[\![\Delta]\!]\hat{\pi}_0,\P\mathscr{D}[\![\grave{\ }[\![\Delta]\!]\psi_0]\!]\hat{\pi}_0\rangle$ instead of $set\hat{\pi}_2\hat{\pi}_0=true$ we have $sewn[\![\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}_2=true$ (provided $apt\psi_0\hat{\rho}_0=true$).

⊲Needless to say, all the above applies equally to the other valuations and is proved by a structural induction for which the foundations have already been laid. The results leading up to 2.6.9 provide sufficient indication of the method of proof for further discussion to be superfluous.⊳

2.7.5. Lemma.

Let $\hat{\pi}_0$, $\hat{\pi}_1$, α^*, I^* and ψ_0 be subject to the constraints laid down in 2.7.4. For some ψ and Δ:Dec suppose that $crowded[\![\Delta]\!]\psi=true$, that $\psi[\![I]\!]\downarrow 1=2$ for all I having $\beta_0[\![I]\!]\downarrow 1$:G or I:$\mathscr{H}[\![\Delta]\!]$, that the list of identifiers common to $\mathscr{H}[\![\Delta]\!]$ and $\mathscr{S}[\![\iota[\![\Delta]\!]\psi_0]\!]$, I^*_0, is included in I^*, and that $torn[\![\Delta]\!]\psi_0=true$. Assume also that any constituent of Δ of the form Δ_0 within Δ_1 is such that no member of $\mathscr{S}[\![\Delta_1]\!]$ is in $\mathscr{H}[\![\Delta_0]\!]$ and no member of $\mathscr{H}[\![\Delta_1]\!]$ is in $\mathscr{S}[\![\Delta_0]\!]\S\mathscr{H}[\![\Delta_0]\!]$. Either $\lambda\zeta.\mathscr{S}[\![\Delta]\!]\zeta\beta_0\upsilon_0\sigma_0$ and $\lambda\zeta.\mathscr{S}[\![\iota[\![\Delta]\!]\psi_0]\!]\zeta\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0$ are proper or they are both $\perp$ or $\top$; moreover the choice of alternative does not depend on θ_0 or on the value of $\rho_0[\![I]\!]\downarrow 1$ when $\psi[\![I]\!]\downarrow 1=2$. If the former situation obtains then, writing $\hat{\pi}_2=\langle \P\mathscr{S}[\![\Delta]\!]\hat{\pi}_0, \P\mathscr{S}[\![\iota[\![\Delta]\!]\psi_0]\!]\hat{\pi}_0\rangle$ and $\hat{\pi}_3=\langle \hat{\pi}_2, \langle \grave{\rho}_2, \grave{\upsilon}_2, \langle \lambda\alpha.\langle area\alpha\grave{\sigma}_2\wedge\sim\alpha:\alpha^*, hold\alpha\grave{\sigma}_2\rangle\rangle \S\grave{\sigma}_2\dagger 1\rangle\rangle$, $\hat{\pi}_3$ obeys the constraints imposed on $\hat{\pi}_1$ above; in addition, if $\bigwedge\{w\hat{\omega}\hat{\pi}_n \mid kent0\hat{\omega}\hat{\pi}_n\}$ and $\bigwedge\{\acute{\omega}_0=\acute{\omega}_1\vee\sim\grave{\omega}_0:L\vee\sim\grave{\omega}_0=\grave{\omega}_1 \mid kent1\hat{\omega}_0\hat{\pi}_n\wedge kent1\hat{\omega}_1\hat{\pi}_n\}$ are both true when n is 1, if $\beta_0[\![I]\!]\downarrow 1=\P\mathscr{S}[\![\Delta]\!]\beta_0\sigma_0\downarrow\nu$ whenever I:I^* and $I=\mathscr{H}[\![\Delta]\!]\downarrow\nu$ for some ν and if $apt\psi_0\hat{\rho}_0=true$, then $sewn[\![\Delta]\!]1\psi_0\hat{\pi}_0\hat{\pi}_2=true$ and the conditions take the value $true$ when n is 4. Here $\hat{\pi}_4=\langle \hat{\pi}_3, \langle \grave{\rho}_3, \grave{\upsilon}_3, \langle \lambda\alpha.\langle area\alpha\grave{\sigma}_3\vee\alpha:\alpha^*_0, hold\alpha\grave{\sigma}_3\rangle\rangle \S\grave{\sigma}_3\dagger 1\rangle\rangle$ and $\alpha^*_0=\langle \grave{\rho}_0[\![I^*_0\downarrow 1]\!]\downarrow 1, \ldots, \grave{\rho}_0[\![I^*_0\downarrow\#I^*_0]\!]\downarrow 1\rangle$.

⊲The proof about $\hat{\pi}_2$ and $\hat{\pi}_3$ resembles that of 2.7.4 too closely to warrant much attention. We shall, however, consider two of the cases in the structural induction about π_4 which will serve to indicate how to deal with the remainder. The first of these is of interest as an exemplar of the occasions when the alternative recursion operator of 1.3.4 gives rise to a marginally less complex proof than the one we have actually adopted.

⊲Suppose that Δ is of the form $I_1,\ldots,I_n$==E and that $\lambda\zeta.\mathscr{R}[\![E]\!]\zeta\beta_0\upsilon_0\sigma_0$ and $\lambda\zeta.\mathscr{R}[\![\iota[\![E]\!]\psi_0]\!]\zeta\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0$ are not improper (since

otherwise $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\rho_0\upsilon_0\sigma_0$ and $\lambda\zeta.\mathscr{T}[\![\iota[\![\Delta]\!]\psi_0]\!]\zeta\rho_0\upsilon_0\vartheta_0$ are improper). Let $\pi_5=\langle \mathscr{R}[\![E]\!]\pi_0, \mathscr{U}_\bullet[\![E]\!]\psi_0]\!]\pi_0\rangle$ and $\pi_6=\langle \pi_5,\langle \rho_5,\upsilon_5,\langle \lambda\alpha.\langle area\alpha\sigma_5\wedge\sim\alpha:\alpha^*, hold\alpha\vartheta_5\rangle\rangle \S\vartheta_5\dagger 1\rangle$, so that π_6 satisfies the constraints on π_1 of 2.7.4. In particular $\omega_0\langle \upsilon_5\downarrow 1,\upsilon_5\downarrow 1\rangle\pi_5=true$ so $\#\upsilon_5\downarrow 1|L^*=n$ if and only if $\#\upsilon_5\downarrow 1|L^*=n$. Unless $\#\upsilon_5\downarrow 1|L^*=n$ $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\cup\rho_0\upsilon_0\sigma_0$ and $\lambda\zeta.\mathscr{T}[\![\iota[\![\Delta]\!]\psi_0]\!]\zeta\rho_0\upsilon_0\vartheta_0$ are both τ, so presume that $\#\upsilon_5\downarrow 1|L^*=n$.

⊲Assume first that $\psi_0[\![I_1]\!]\downarrow 1\wedge\ldots\wedge\psi_0[\![I_n]\!]\downarrow 1=true$, so that $I_m:I^*$ when $1\leq m\leq n$ and, writing α^*_0 for $\langle \rho_5[\![I_1]\!]\downarrow 1,\ldots,\rho_5[\![I_n]\!]\downarrow 1\rangle$, every member of α^*_0 is in α^*. Set $\pi_2=\langle\langle invert\rho_5(arid[holds(\upsilon_5\downarrow 1)\sigma_5/\langle I_1,\ldots,I_n\rangle]),\upsilon_5\dagger 1,\sigma_5\rangle$, $\langle \rho_5,\upsilon_5\dagger 1, updates\alpha^*_0(holds(\upsilon_5\downarrow 1)\vartheta_5)\vartheta_5\rangle\rangle$, $\pi_3=\langle \pi_2,\langle \rho_2,\upsilon_2,\langle \lambda\alpha.\langle area\alpha\vartheta_2\wedge\sim\alpha:\alpha^*, hold\alpha\vartheta_2\rangle\rangle\S\vartheta_2\dagger 1\rangle\rangle$ and $\pi_4=\langle \pi_3,\langle \rho_3,\upsilon_3, updates\alpha^*_0(holds(\upsilon_5\downarrow 1)\vartheta_5)\vartheta_3\rangle\rangle$. Now $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\rho_0\upsilon_0\sigma_0=\lambda\zeta.\zeta\rho_2\upsilon_2\vartheta_2$ and $\lambda\zeta.\mathscr{T}[\![\iota[\![\Delta]\!]\psi_0]\!]\zeta\rho_0\upsilon_0\vartheta_0=\lambda\zeta.\zeta\rho_2\upsilon_2\vartheta_2$ so both are proper. For every ν and $\hat\omega$ we can show by induction that $kent\nu\hat\omega\pi_3\supset gyven\hat\omega\langle holds(\upsilon_5\downarrow 1)\sigma_5,\alpha^*_0\rangle\vee kent\nu\hat\omega\pi_3$ since the properties of π_5 assure us that α^* and $\upsilon_5\downarrow 1$ have no members in common. Hence π_3 satisfies the conditions imposed on π_1 in 2.7.4.

From 2.7.1 it is plain that $holds(\upsilon_5\downarrow 1)\sigma_5=\mathscr{U}[\![\Delta]\!]\rho_0\sigma_0\downarrow 1$, so if $\rho_0[\![\mathscr{H}[\![\Delta]\!]\downarrow\nu]\!]\downarrow 1=\mathscr{U}[\![\Delta]\!]\rho_0\sigma_0\downarrow\nu$ for ν having $1\leq\nu\leq\#\mathscr{H}[\![\Delta]\!]$ we must have $\hat\rho_2=\hat\rho_0$. Let $\bigwedge\{\omega\hat\omega\pi_r|kent0\hat\omega\pi_r\}$ and $\bigwedge\{\acute\omega_0=\acute\omega_1\vee\sim\grave\omega_0:L\wedge\sim\grave\omega_0=\grave\omega_1|kent1\hat\omega_0\pi_r\wedge kent1\acute\omega_1\pi_r\}$ be *true* when r is 1, so that by 2.7.4 they are *true* when r is 6. Because $\hat\rho_4=\hat\rho_6$ we can show by the usual induction technique that $kent\nu\hat\omega\pi_4\supset kent\nu\hat\omega\pi_6$ for all $\nu<2$ and all $\hat\omega$, which suffices to show that the conditions take the value *true* when r is 4. Thus the result is established in this case.⊳

When $\psi_0[\![I_1]\!]\downarrow 1\wedge\ldots\wedge\psi_0[\![I_n]\!]\downarrow 1=false$ $\iota[\![\Delta]\!]\psi_0$ is $I_1,\ldots,I_n==_\bullet[\![E]\!]\psi_0$ instead of $I_1,\ldots,I_n=_\bullet[\![E]\!]\psi_0$. We can show that $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\rho_0\upsilon_0\sigma_0$ and

$\lambda\zeta.\mathscr{T}[\![\iota[\![\Delta]\!]\psi_0]\!]\zeta\hat{\rho}_0\upsilon_0\delta_0$ are both $\perp$ and $\top$ unless they are proper much as before. A simple variant of the argument concerning $\mathscr{D}[\![I==E]\!]$ in 2.6.5 provides the proof of the remainder. Consequently the present proposition holds when Δ is $I_1,\ldots,I_n==E$.⊁

The verification for I==E proceeds along similar lines while that for I=E is less interesting still. Accordingly we next discuss the situation when Δ is of the form Δ_0 within Δ_1 with Δ_0 and Δ_1 constrained as in the statement of the lemma. Suppose that $\lambda\zeta.\mathscr{D}[\![\Delta_0]\!]\zeta\hat{\rho}_0\upsilon_0\sigma_0$ and $\lambda\zeta.\mathscr{D}[\![\mathscr{A}[\![\Delta_0]\!]\psi_0]\!]\zeta\hat{\rho}_0\upsilon_0\delta_0$ are both proper and let $\hat{\pi}_5=\langle\neg\mathscr{D}[\![\Delta_0]\!]\hat{\pi}_0,\neg\mathscr{D}[\![\mathscr{A}[\![\Delta_0]\!]\psi_0]\!]\hat{\pi}_0\rangle$ and $\hat{\pi}_6=\langle\hat{\pi}_5,\langle\hat{\rho}_5,\upsilon_5,\langle\lambda\alpha.\langle area\alpha\delta_5\wedge\sim\alpha:\alpha^*,hold\alpha\delta_5\rangle\rangle\S\delta_5\dagger1\rangle\rangle$. When $apt\psi\hat{\rho}_5=true\ sewn[\![\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}_5=true$, so as $crowded[\![\Delta_0]\!]\psi=true$ in general $\hat{\pi}_5$ and $\hat{\pi}_6$ satisfy the constraints imposed on $\hat{\pi}_0$ and $\hat{\pi}_1$ in 2.7.4. Accordingly either $\lambda\zeta.\mathscr{T}[\![\Delta_1]\!](\zeta\circ trim[\![\Delta_1]\!]\hat{\rho}_0)\hat{\rho}_0\upsilon_0\sigma_0$ and $\lambda\zeta.\mathscr{T}[\![\iota[\![\Delta_1]\!]\psi_0]\!](\zeta\circ trim[\![\Delta_1]\!]\hat{\rho}_0)\hat{\rho}_5\upsilon_5\delta_5$ take the same improper value or they are both proper. In the latter case, writing $\hat{\pi}_7=\langle\neg\mathscr{T}[\![\Delta_1]\!]\hat{\pi}_5,\neg\mathscr{T}[\![\iota[\![\Delta_1]\!]\psi_0]\!]\hat{\pi}_5\rangle$ and $\hat{\pi}_2=\langle\langle trim[\![\Delta_1]\!]\hat{\rho}_0\hat{\rho}_7,\upsilon_7,\sigma_7\rangle,\langle trim[\![\Delta_1]\!]\hat{\rho}_0\hat{\rho}_7,\upsilon_7,\delta_7\rangle\rangle$, $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\hat{\rho}_0\upsilon_0\sigma_0=\lambda\zeta.\zeta\hat{\rho}_2\upsilon_2\sigma_2$ and $\lambda\zeta.\mathscr{T}[\![\iota[\![\Delta]\!]\psi_0]\!]\zeta\hat{\rho}_0\upsilon_0\delta_0=\lambda\zeta.\zeta\hat{\rho}_2\upsilon_2\delta_2$. The proof of 2.6.6 ensures that when $apt\psi_0\hat{\rho}_0=true\ sewn[\![\Delta]\!]1\psi_0\hat{\pi}_0\hat{\pi}_2=true$; indeed this is why it is couched in terms of a transformation for which $\mathscr{T}[\![\Delta_1]\!]$ need not be $\mathscr{T}[\![\iota[\![\Delta_1]\!]\psi_0]\!]$ instead of one suited to the special situation of 2.5.9. Now assume also that $\hat{\rho}_0[\![\mathscr{H}[\![\Delta]\!]\downarrow\nu]\!]\downarrow1=\neg\mathscr{T}[\![\Delta]\!]\hat{\rho}_0\sigma_0\downarrow\nu$ whenever $1\leq\nu\#\mathscr{H}[\![\Delta]\!]$ and that $\bigwedge\{\omega\hat{\omega}\hat{\pi}_n\mid kent0\hat{\omega}\hat{\pi}_n\}$ and $\bigwedge\{\hat{\omega}_0=\hat{\omega}_1\vee\sim\hat{\omega}_0:\mathsf{L}\vee\sim\hat{\omega}_0=\hat{\omega}_1\mid kent1\hat{\omega}_0\hat{\pi}_n\wedge kent1\hat{\omega}_1\hat{\pi}_n\}$ are $true$ when n=1. From 2.7.4 we know that $\hat{\rho}_5[\![\mathscr{H}[\![\Delta]\!]\downarrow\nu]\!]\downarrow1=\neg\mathscr{T}[\![\Delta]\!]\hat{\rho}_0\sigma_0\downarrow\nu$ and that these conditions are $true$ when n is 6; consequently $\hat{\rho}_5[\![\mathscr{H}[\![\Delta_1]\!]\downarrow\nu]\!]\downarrow1=\neg\mathscr{T}[\![\Delta_1]\!]\hat{\rho}_5\sigma_5\downarrow\nu$ whenever $1\leq\nu\leq\#\mathscr{H}[\![\Delta_1]\!]$ and we can apply the induction hypothesis to $\hat{\pi}_5$. Deriving

$\hat{\pi}_8$ and $\hat{\pi}_9$ from $\hat{\pi}_7$ just as $\hat{\pi}_3$ and $\hat{\pi}_4$ are derived from $\hat{\pi}_2$ we thus see that these conditions are *true* when n is 9. Finally, as $kent\nu\hat{\omega}\hat{\pi}_4 \supset kent\nu\hat{\omega}\hat{\pi}_9$ for all ν and $\hat{\omega}$, $\bigwedge\{w\hat{\omega}\hat{\pi}_n \mid kent0\hat{\omega}\hat{\pi}_n\}$ and $\bigwedge\{\acute{\omega}_0=\acute{\omega}_1 \vee \sim\grave{\omega}_0 : L \vee \sim\grave{\omega}_0=\grave{\omega}_1 \mid kent1\hat{\omega}_0\hat{\pi}_n \wedge kent1\hat{\omega}_1\hat{\pi}_n\}$ take the value *true* when n is 4.

The proof required when Δ is Δ_1 and...and Δ_n is similar to this and will therefore be omitted.⊁

2.7.6. Lemma.

Let $\hat{\pi}_0$ and ψ_0 satisfy $fit\hat{\pi}_0\hat{\pi}_0 \wedge apt\psi_0\hat{\rho}_0 = true$, and suppose that Δ:Dec is such that $opts(\mathscr{H}[\![\Delta]\!])\psi_0 = true^*$, $rent[\![\Delta]\!]\psi_0 = true$ and $crowded[\![\Delta]\!]\psi = true$ for some ψ having $\psi[\![I]\!]\downarrow 1 = 2$ whenever $\grave{\rho}_0[\![I]\!]\downarrow 1 : L$ or $I : \mathscr{H}[\![\Delta]\!]$. Assume also that any constituent of Δ of the form Δ_0 within Δ_1 is such that as member of $\mathscr{I}[\![\Delta_1]\!]$ is in $\mathscr{H}[\![\Delta_0]\!]$ and no member of $\mathscr{H}[\![\Delta_1]\!]$ is in $\mathscr{I}[\![\Delta_0]\!] \S \mathscr{H}[\![\Delta_0]\!]$. Take a_1 to be the relation a of 2.2.2 and a_2 to be the relation a of 2.4.5, we assume that for all o_0, o_1, o_2 and o_3 in the relevant domains if $a_2\langle o_1, o_0\rangle \wedge a_1\langle o_3, o_1\rangle \wedge a_1\langle o_3, o_2\rangle = true$ then $a_2\langle o_2, o_0\rangle = true$. For any ζ such that $\bigwedge\{c\zeta\hat{\pi} \mid sewn[\![\Delta]\!]o\psi_0\hat{\pi}_0\hat{\pi}\} = true$ we have $c\langle \mathscr{D}[\![\text{rec } \Delta]\!]\zeta, \mathscr{D}[\![\mathscr{A}[\![\text{rec } \Delta]\!]\psi_0]\!]\zeta\rangle\hat{\pi}_0 = true$.

⊀Let $\hat{\pi}_1$ have $p\hat{\pi}_1 = fit\hat{\pi}_1\hat{\pi}_0 = true$, and define $\hat{\sigma}_4$ to be $\langle \mathscr{D}[\![\text{rec } \Delta]\!]\zeta\hat{\rho}_1\upsilon_1\sigma_1, \mathscr{D}[\![\mathscr{A}[\![\text{rec } \Delta]\!]\psi_0]\!]\zeta\grave{\rho}_1\upsilon_1\partial_1\rangle$ for some ζ subject to $\bigwedge\{c\,\zeta\hat{\pi} \mid sewn[\![\Delta]\!]o\psi_0\hat{\pi}_0\hat{\pi}\} = true$. Set $\psi_1 = \psi_0[false^*/\mathscr{I}[\![\Delta]\!]][true^*/\mathscr{H}[\![\Delta]\!]]$, $\langle \acute{\alpha}^*, \grave{\alpha}^*\rangle = \langle news(\#\mathscr{I}[\![\Delta]\!])\sigma_1, news(\#\mathscr{I}[\![\mathscr{A}[\![\Delta]\!]\psi_1]\!])\partial_1\rangle$, $\hat{\rho}_2 = \langle fix(\lambda\rho.\hat{\rho}_1[\acute{\alpha}^*/\mathscr{I}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho\sigma_2/\mathscr{H}[\![\Delta]\!]]), \grave{\rho}_1[\grave{\alpha}_1^*/\mathscr{I}[\![\mathscr{A}[\![\Delta]\!]\psi_1]\!]]\rangle$, $\upsilon_2 = \upsilon_1$ and $\sigma_2 = \langle updates\acute{\alpha}^*dummy^*\sigma_1, updates\grave{\alpha}^*dummy^*\partial_1\rangle$, so that $\hat{\sigma}_4 = \langle \mathscr{T}[\![\Delta]\!]\zeta\hat{\rho}_2\upsilon_2\sigma_2, \mathscr{T}[\![\mathscr{A}[\![\Delta]\!]\psi_1]\!]\zeta\grave{\rho}_2\upsilon_2\partial_2\rangle$. As $\mathscr{H}[\![\mathscr{A}[\![\Delta]\!]\psi_1]\!] = \langle\rangle$ here we can take the I* and α* of 2.7.5 to be $\mathscr{H}[\![\Delta]\!]$ and $\langle \grave{\rho}_2[\![\mathscr{H}[\![\Delta]\!]\downarrow 1]\!]\downarrow 1, \ldots, \grave{\rho}_2[\![\mathscr{H}[\![\Delta]\!]\downarrow \#\mathscr{H}[\![\Delta]\!]]\!]\downarrow 1\rangle$ respectively.

When $\hat{\pi}_3 = \langle \hat{\pi}_2, \langle \grave{\rho}_2, \upsilon_2, \langle \lambda\alpha.\langle area\alpha\partial_2 \wedge \sim\alpha : \alpha^*, hold\alpha\partial_2\rangle\rangle \S \partial_2 \dagger 1\rangle\rangle$

we can show that for all $\hat{\omega}$

$kent1\hat{\omega}\hat{\pi}_3 \supset hoten\hat{\omega}\hat{\rho}_2 \vee \hat{\omega} = \langle dummy, dummy \rangle \vee kent1\hat{\omega}\hat{\pi}_1$, and consequently $\hat{\pi}_2$ and $\hat{\pi}_3$ satisfy the constraints on the $\hat{\pi}_0$ and $\hat{\pi}_1$ of 2.7.4. It follows from 2.7.5 that either $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\hat{\rho}_2\hat{\upsilon}_2\hat{\sigma}_2$ and $\lambda\zeta.\mathscr{T}[\![\ell[\![\Delta]\!]\psi_1]\!]\zeta\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2$ take the same improper value or they are both proper. In the first case $a_2\hat{\rho}_4 = true$ trivially, so we discount it and consider only the second case. Writing $\hat{\pi}_4$ for $\langle \P\mathscr{T}[\![\Delta]\!]\hat{\pi}_2, \P\mathscr{T}[\![\ell[\![\Delta]\!]\psi_1]\!]\grave{\pi}_2 \rangle$ we know from 2.7.2 that $\hat{\pi}_4$ is proper and that $\hat{\rho}_4 = \langle \zeta\hat{\rho}_4\hat{\upsilon}_4\hat{\sigma}_4, \zeta\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4 \rangle$.

To remove the members of G×L from the environment we now introduce

$\hat{\pi}_5 = \langle \langle fix(\lambda\rho.\hat{\rho}_1[\hat{\alpha}^*/\mathscr{G}[\![\Delta]\!]][\P\mathscr{T}[\![\Delta]\!]\rho\hat{\sigma}_5/\mathscr{K}[\![\Delta]\!]]), \hat{\upsilon}_2, \hat{\sigma}_2 \rangle, \grave{\pi}_2 \rangle$ and $\hat{\rho}_6 = \langle \mathscr{T}[\![\Delta]\!]\zeta\hat{\rho}_5\hat{\upsilon}_5\hat{\sigma}_5, \grave{\rho}_4 \rangle$. Because $\hat{\rho}_5[\![I]\!]\downarrow 1 = \hat{\rho}_3[\![I]\!]\downarrow 1$ unless $I : \mathscr{K}[\![\Delta]\!]$ we can infer from 2.7.5 that $\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta\hat{\rho}_5\hat{\upsilon}_5\hat{\sigma}_5$ is proper and that $\hat{\rho}_6 = \langle \zeta\hat{\rho}_6\hat{\upsilon}_6\hat{\sigma}_6, \zeta\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6 \rangle$ where $\hat{\pi}_6 = \langle \P\mathscr{T}[\![\Delta]\!]\hat{\pi}_5, \grave{\pi}_4 \rangle$; we shall show that $a_2\hat{\rho}_6 = true$. Defining $\hat{\pi}_7 = \langle \hat{\pi}_5, \grave{\pi}_3 \rangle$ we can demonstrate by an obvious induction that for all ν and $\hat{\omega}$

$kent\nu\hat{\omega}\hat{\pi}_7 \supset hoten\hat{\omega}\hat{\rho}_5 \vee \hat{\omega} = \langle dummy, dummy \rangle \vee kent\nu\hat{\omega}\hat{\pi}_1$; since $p\hat{\pi}_1 = true$ we thus know that $\bigwedge\{\omega\hat{\omega}\hat{\pi}_7 \mid kent0\hat{\omega}\hat{\pi}_7\}$, $\bigwedge\{\hat{\omega}_0 : L \wedge \hat{\omega}_0 = \hat{\omega}_1 \rightarrow area\hat{\omega}_1\hat{\sigma}_6 \wedge \grave{\omega}_0 = \grave{\omega}_1, true \mid kent1\hat{\omega}_0\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_7\}$ and $\bigwedge\{\grave{\omega}_0 : L \wedge \grave{\omega}_0 = \grave{\omega}_1 \rightarrow area\grave{\omega}_0\grave{\sigma}_6 \wedge \hat{\omega}_0 = \hat{\omega}_1, true \mid kent1\hat{\omega}_0\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_7\}$ are all $true$. Moreover $apt\psi_0\hat{\rho}_5 = true$, so we can apply 2.7.5 to $\hat{\pi}_5$ and $\hat{\pi}_7$ to establish that $sewn[\![\Delta]\!]1\psi_1\hat{\pi}_5\hat{\pi}_6 = true$, $\bigwedge\{\omega\hat{\omega}\pi_6 \mid kent0\hat{\omega}\hat{\pi}_6\} = true$ and $p_0\hat{\pi}_6 = true$. Consequently $sewn[\![\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}_6 = true$ and $p\hat{\pi}_6 = true$, and in view of the nature of ζ we have $a_2\hat{\rho}_6 = true$.

Unfortunately although $\grave{\rho}_6$ is $\grave{\rho}_4$ $\hat{\sigma}_6$ need not be $\hat{\sigma}_4$, and to convince ourselves that $a_2\hat{\rho}_4 = true$ we must use standard semantics, in which the corresponding entities do coincide. If the program in which Δ is embedded has a sensible initial continuation, 2.3.9 entitles us to assume that for some χ, $\hat{\rho}_8$ and $\hat{\sigma}_8$ we have

$u(\acute{\rho}_8,\acute{\rho}_1)\wedge s(\acute{\sigma}_8,\acute{\sigma}_1)\wedge\bigwedge\{c(\chi\acute{\rho},(\zeta,\grave{\rho},\upsilon_1))\mid knit[\![\text{rec }\Delta]\!]0(\acute{\rho}_8,\acute{\rho}_1)\hat{\rho}\wedge u\hat{\rho}\}=true$ (adopting the predicates of 2.2.7); from 2.3.8 we even know that $a(\acute{\sigma}_8,\acute{\sigma}_4)=true$, where $\acute{\sigma}_8=\mathscr{D}[\![\text{rec }\Delta]\!]\acute{\rho}_8\chi\acute{\sigma}_8$. Since $\P\mathscr{A}[\![\text{rec }\Delta]\!]\acute{\pi}_1$ is the proper vector $\acute{\pi}_4$, $\P\mathscr{A}[\![\text{rec }\Delta]\!]\acute{\rho}_8\acute{\sigma}_8$ is proper by 2.7.2 and thus is $\P\mathscr{T}[\![\Delta]\!]\acute{\rho}_9\acute{\sigma}_9$, where owing to 1.5.8 we can take $\acute{\rho}_9$ to be $fix(\lambda\rho.\acute{\rho}_8[\acute{\alpha}^*/\mathscr{S}[\![\Delta]\!]][\P\mathscr{A}[\![\Delta]\!]\rho\acute{\sigma}_9/\mathscr{A}[\![\Delta]\!]])$ and $\acute{\sigma}_9$ to be $updates\acute{\alpha}^*dummy^*\acute{\sigma}_8$. As $s(\acute{\sigma}_8,\acute{\sigma}_1)=true$ it is plain that $s(\acute{\sigma}_9,\acute{\sigma}_5)=true$, and therefore by the remarks following 2.7.2 $u(\acute{\rho}_9,\acute{\rho}_5)=true$. Whenever $knit[\![\Delta]\!]1(\acute{\rho}_9,\acute{\rho}_5)\hat{\rho}=true$ $knit[\![\text{rec }\Delta]\!]0(\acute{\rho}_8,\acute{\rho}_1)\hat{\rho}=true$ also; hence applying 2.7.2 to $\mathscr{T}[\![\Delta]\!]$ and using the nature of χ reveals that $a_1(\acute{\sigma}_8,\acute{\sigma}_6)=a_1((\chi^*\P\mathscr{T}[\![\Delta]\!]\acute{\rho}_9)\acute{\sigma}_9,\zeta\acute{\rho}_6\upsilon_6\acute{\sigma}_6)=true$.

In consequence $a_2\hat{\sigma}_6\wedge a_1(\acute{\sigma}_8,\acute{\sigma}_4)\wedge a_1(\acute{\sigma}_8,\acute{\sigma}_6)=true$ and from our hypothesis about a_2 and a_1 we can deduce that $a_2(\acute{\sigma}_4,\grave{\sigma}_6)=true$. This means that $a_2\hat{\sigma}_4=true$ and thus that $c(\mathscr{D}[\![\text{rec }\Delta]\!]\zeta,\mathscr{D}[\![\mathscr{A}[\![\text{rec }\Delta]\!]\psi_0]\!]\zeta)\hat{\pi}_0=true$.▷

The restriction on a_1 and a_2 is quite plausible, for in essence it states that the equivalence between members of A must not concern itself with the locations and functions used to compute results. In the absence of such a limitation there is no reason to suppose that $\acute{\sigma}_4$ and $\grave{\sigma}_6$ will be equivalent when $\acute{\sigma}_6$ and $\grave{\sigma}_6$ are; we might, for instance, take A to be P, a_2 to be $\lambda\hat{\pi}.\hat{\pi}=(\bot,\bot)\vee\hat{\pi}=(\top,\top)\rightarrow true,p\hat{\pi}$ and ζ to be $(\lambda\rho\upsilon\sigma.(\rho,\upsilon,\sigma),\lambda\rho\upsilon\sigma.(\rho,\upsilon,\sigma))$, when pairing $\acute{\sigma}_4$ with $\grave{\sigma}_6$ may give rise to illegitimate members of V×G. Intuitively it is more reasonable to take a_2 to be $\lambda\hat{\pi}.\hat{\pi}=(\bot,\bot)\vee\hat{\pi}=(\top,\top)\rightarrow true,\bigwedge\{\omega_0\hat{\beta}\hat{\pi}\vee\sim(\acute{\beta}:\mathbb{B}\vee\grave{\beta}:\mathbb{B})\vee\sim gyven(\acute{\sigma}\downarrow 3,\grave{\sigma}\downarrow 3)\mid\hat{\beta}:\mathbb{V}\times\mathbb{V}\}$, so that if a_1 is $\lambda(\acute{\sigma},\grave{\pi}).(\acute{\sigma},\grave{\pi})=(\bot,\bot)\vee(\acute{\sigma},\grave{\pi})=(\top,\top)\rightarrow true,s\acute{\sigma}$ the stipulation about a_1 and a_2 is valid. The latter version of a_2 also satisfies the additional assumption which we require in our final result.

2.7.7. Proposition.

Suppose that a_1 and a_2, as defined in 2.7.6, are such that for all relevant o_0, o_1, o_2 and o_3 if $a_2\langle o_1,o_2\rangle \wedge a_1\langle o_3,o_1\rangle \wedge a_1\langle o_3,o_2\rangle = true$ then $a_2\langle o_2,o_0\rangle = true$ and if $a_2\langle o_2,o_1\rangle \wedge a_2\langle o_0,o_1\rangle = true$ then $a_2\langle o_2,o_0\rangle = true$. Let rec Δ be a recursive Mal declaration such that no constituents of Δ contain further recursive declarations by incidence and any constituents of Δ of the form Δ_0 within Δ_1 have $\mathscr{I}[\![\Delta_1]\!]$ disjoint from $\mathscr{H}[\![\Delta_0]\!]$ and $\mathscr{H}[\![\Delta_1]\!]$ disjoint from $\mathscr{I}[\![\Delta_0]\!] \S \mathscr{H}[\![\Delta_0]\!]$. For any ψ_0 and certain $\hat{\pi}_0$ having $apt\psi_0\hat{\rho}_0 \wedge fit\hat{\pi}_0\hat{\pi}_0 = true$ and

$crushed[\![\text{rec } \Delta]\!](\lambda I.\#\rho_0[\![I]\!] > 0 \rightarrow (\hat{\rho}_0[\![I]\!]\downarrow 1 : G \rightarrow 2,1),1) \wedge rent[\![\text{rec } \Delta]\!]\psi_0 = true$

rec Δ and $\mathscr{d}[\![\text{rec } \Delta]\!]\psi_0$ are equivalent with respect to $\hat{\pi}_0$.

⊲Let $\psi_1 = \psi_0[false^*/\mathscr{I}[\![\Delta]\!]][true^*/\mathscr{H}[\![\Delta]\!]]$,

$\psi_2 = \psi_0[false^*/\mathscr{I}[\![\Delta]\!]][opts(\mathscr{H}[\![\Delta]\!])\psi_0/\mathscr{H}[\![\Delta]\!]]$ and

$\psi_3 = \psi_0[false^*/\mathscr{I}[\![t[\![\Delta]\!]\psi_2]\!]][true^*/\mathscr{H}[\![t[\![\Delta]\!]\psi_2]\!]]$. Introduce a fresh version of opt, opt_0, having

$opt_0 = \lambda I\psi.\ \psi = \psi_0 \wedge I : \mathscr{H}[\![\Delta]\!] \rightarrow true,$

$(\lambda\psi'.opt[\![I]\!]\psi')$

$(\lambda I.(\lambda v.v > 0 \rightarrow \langle \psi[\![I]\!]\downarrow 1,\ldots,\psi[\![I]\!]\downarrow v\rangle \S \psi_2[\![I]\!], \psi[\![I]\!])(\#\psi[\![I]\!] - \#\psi_2[\![I]\!]));$

$sewn_0$ is derived from this just as $sewn$ is from opt in 2.4.5. Take $t[\![\Delta]\!]\psi_2$ to be the transform of Δ according to 1.4.6, but take $t[\![t[\![\Delta]\!]\psi_2]\!]\psi_3$ and $t[\![\Delta]\!]\psi_1$ to be the transforms induced by opt_0 rather than those induced by opt; a simple structural induction serves to show that $t[\![t[\![\Delta]\!]\psi_2]\!]\psi_3 = t[\![\Delta]\!]\psi_1$.

Suppose that the programs in which Δ and its transform are embedded are provided with continuations which tally with respect to c. By 2.6.9 there will be some ζ_0 such that $\bigwedge\{c\zeta_0\hat{\pi} \mid sewn[\![\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}\} = true$. There will also be some ζ_1 and $\hat{\pi}_1$ having $\bigwedge\{c\langle\zeta_0,\zeta_1\rangle\hat{\pi} \mid sewn_0[\![t[\![\Delta]\!]\psi_2]\!]0\psi_0\langle\hat{\pi}_0,\hat{\pi}_1\rangle\hat{\pi}\} = true$ for which we can even assume that $\bigwedge\{c\langle\zeta_0,\zeta_1\rangle\hat{\pi} \mid sewn_0[\![\Delta]\!]0\psi_0\langle\hat{\pi}_0,\hat{\pi}_1\rangle\hat{\pi}\} = true$

as ψ_1 'factors through' ψ_2 (in the sense that if $apt\psi_2\hat{\rho}_2 \wedge apt\psi_3\langle\hat{\rho}_2,\hat{\rho}_3\rangle = true$ then $apt\psi_1\langle\hat{\rho}_2,\hat{\rho}_3\rangle = true$ for all $\hat{\rho}_2$ and $\hat{\rho}_3$). Applying 2.7.5 to the transformations induced by opt_0 on rec $\iota[\![\Delta]\!]\psi_2$ and on rec Δ,

$c\langle\mathscr{D}[\![\text{rec } \iota[\![\Delta]\!]\psi_2]\!]\zeta_0, \mathscr{D}[\![\text{rec } \iota[\![\iota[\![\Delta]\!]\psi_2]\!]\psi_3]\!]\zeta_1\rangle\langle\hat{\pi}_0,\hat{\pi}_1\rangle = true$ and

$c\langle\mathscr{D}[\![\text{rec } \Delta]\!]\zeta_0, \mathscr{D}[\![\text{rec } \iota[\![\Delta]\!]\psi_1]\!]\zeta_1\rangle\langle\hat{\pi}_0,\hat{\pi}_1\rangle = true$.

Take any $\hat{\pi}_2$ and $\hat{\pi}_3$ having $p\hat{\pi}_2 \wedge fit\hat{\pi}_2\hat{\pi}_0 = true$ and $p\langle\hat{\pi}_2,\hat{\pi}_3\rangle \wedge fit\langle\hat{\pi}_2,\hat{\pi}_3\rangle\langle\hat{\pi}_0,\hat{\pi}_1\rangle = true$, so that $p\langle\hat{\pi}_2,\hat{\pi}_3\rangle \wedge fit\langle\hat{\pi}_2,\hat{\pi}_3\rangle\langle\hat{\pi}_0,\hat{\pi}_1\rangle = true$. We have demonstrated that $a_2\langle\mathscr{D}[\![\text{rec } \iota[\![\Delta]\!]\psi_2]\!]\zeta_0\hat{\rho}_2\hat{\upsilon}_2\hat{\sigma}_2, \mathscr{A}[\![\text{rec } \iota[\![\Delta]\!]\psi_1]\!]\zeta_1\hat{\rho}_3\hat{\upsilon}_3\hat{\sigma}_3\rangle = true$ and that $a_2\langle\mathscr{D}[\![\text{rec } \Delta]\!]\zeta_0\hat{\rho}_2\hat{\upsilon}_2\hat{\sigma}_2, \mathscr{D}[\![\text{rec } \iota[\![\Delta]\!]\psi_1]\!]\zeta_1\hat{\rho}_3\hat{\upsilon}_3\hat{\sigma}_3\rangle = true$, and hence from our assumption about a_2

$a_2\langle\mathscr{D}[\![\text{rec } \Delta]\!]\zeta_0\hat{\rho}_2\hat{\upsilon}_2\hat{\sigma}_2, \mathscr{D}[\![\mathscr{A}[\![\text{rec } \Delta]\!]\psi_0]\!]\zeta_0\hat{\rho}_2\hat{\upsilon}_2\hat{\sigma}_2\rangle = true$, as was to be proved.▷

The condition about within declarations is essential both for the validity of the proof of 2.6.6 and for the truth of the theorem above. In 1.3.6 it was pointed out that rec (*f*=fn*z*.*z* within *f*=fn*z*.(\$*f*)*z*) inside (\$*f*)0 does not terminate; yet 1.4.6 can make it the transform of rec (*f*=fn*z*.*z* within *f*==fn*z*.*fz*) inside *f*0, which returns a location containing 0 as its result. By contrast, the insistence that there be no recursive declarations by incidence in Δ is purely a technical device intended to obviate the need for predicates on programs akin to those of 2.6.1. Were these to be introduced rather than requiring that $crowded[\![\Delta]\!]\psi = true$ in 2.7.5 we would be content with $crushed[\![\Delta]\!]\psi = true$.

Strictly speaking we should not make assertions like the one above to the effect that $p\langle\hat{\pi}_2,\hat{\pi}_3\rangle = true$ if $p\hat{\pi}_2 \wedge p\langle\hat{\pi}_2,\hat{\pi}_3\rangle = true$. Rather we should work in terms of a predicate roughly resembling $p\hat{\pi}_2 \wedge p\langle\hat{\pi}_2,\hat{\pi}_3\rangle \wedge p\langle\hat{\pi}_2,\hat{\pi}_3\rangle$, which could be built up as in 2.2.8 or 2.4.5.

CHAPTER THREE

STACK SEMANTICS

3.1. Idealized versions of realistic implementations.

3.1.1. Remitted procedure pointers.

The implementation of recursion implicit in 2.1.4 is absurdly inefficient, for it demands that during the evaluation of a recursively declared identifier I there be three stores: the one forming part of ρ⟦I⟧↓1, the one forming part of ρ⟦rec⟧↓1 and the one which can be modified by the evaluating mechanism. It would be possible to decrease the amount of memory needed by copying only those locations α having $plot\alpha(\rho⟦I⟧\downarrow1\downarrow2)()(\rho⟦I⟧\downarrow1\downarrow3)$ or $plot\alpha(\rho⟦rec⟧\downarrow1\downarrow1)(\rho⟦rec⟧\downarrow1\downarrow2)(\rho⟦rec⟧\downarrow1\downarrow3)$ equal to *true*, but tracing them all would be very time-consuming. In this chapter recursive declarations will be described in terms of a formalism which is as convincing as that of 2.1.1 and which nevertheless gives rise to sensible implementation techniques. Because these involve wielding pointers into the stack instead of storing members of U and Y as portions of function closures and label entry points, the definition of Mal in appendix 3 (which uses the resulting equations) will be called its 'stack semantics'.

It is not enough to revise the treatment of recursion, since Mal has inherited from Pal a domain structure which allows functions and label variables to be stored. By means of an assignment any member of V can be passed out of the program block in which it is set up, so some of the free variables of a function may not be in scope when it is eventually applied. A satisfactory interpreter for the language must either keep little elements of U in closures or preserve more than just the current level of the environment. Even worse inefficiencies arise with label entry points, which require not only surrogate environments but private

stacks also: pointers into the current incarnations of these entities are inadequate because the values present at the time of declaration may have been overwritten long ago. Even programs like that of 3.1.5 presume the existence of an elaborate mechanism for handling the stack.

Notwithstanding this, there are many Mal programs which produce correct answers if their abstractions and labels are translated into pieces of code lacking supplementary environments and stacks. When executed, these pieces of code will obtain the state vectors they require by manipulating pointers into the existing state; in view of the remarks above they will not yield gibberish only if they are confined to the blocks wherein lie their declarations. In 3.1.4 we shall formulate syntactic constraints on assignment statements sufficient to make sure that this does not happen.

Suppose that within a particular block no assignments of locally created members of L*, J or F are made to identifiers declared outside it. When the flow of control leaves it, the locations appended to the store area after entry to the block will not be accessible using *plot* from those adjoined before entry. Moreover the environment brought into play will be that pertaining prior to entry, whilst the stack will differ from the earlier one only by the addition of an extra element. Hence if the exits from the block are restricted as in 2.6.2 there will be no need to retain the locations set up inside it. To discard them by reducing the area to its original size we introduce

$restore=\lambda\sigma_0\sigma_1.(\lambda\alpha.(area\alpha\sigma_0 \wedge area\alpha\sigma_1, hold\alpha\sigma_1))\,\S\sigma_1\dagger 1.$

We also require a primitive which by analogy with revert decreases the height of the stack:

$pop=\lambda\upsilon_0\upsilon_1.\upsilon_1\dagger(\#\upsilon_1-\#\upsilon_0).$

On entering a block we tuck away the current environment

level, stack height and store area in the environment so that they can be referred to when we adjust the state after leaving the block. For shortness we shall actually hoard the entire state $\langle \rho,\upsilon,\sigma\rangle$ rather than $\langle\langle \lambda I.\#\rho[\![I]\!],\#\rho[\![\mathsf{res}]\!],\#\rho[\![\mathsf{rec}]\!]\rangle,\#\upsilon,\lambda\alpha.\mathit{area}\alpha\sigma\rangle$ Preserving these extra facts does not run counter to the principle that only one state vector be accessed, because they are never needed by the semantic equations and can be eliminated by a structural induction using simple inclusive predicates. As we have already intimated, in stack semantics G is not a summand of D and the entity $\rho[\![\mathsf{rec}]\!]$ is not required to serve the purpose for which it was intended in 2.1.4; accordingly we shall use it as the vehicle by which we transfer the house-keeping information from the beginning to the end of the block. For any ν $\rho[\![\mathsf{rec}]\!]\downarrow 1\downarrow 1$ will provide the pointers set up on entry into a block which surrounds the current one, so we can return the state to a size appropriate to it by means of

$$\mathit{level}=\lambda\nu\pi.(\lambda\pi'.\nu<1\rightarrow\langle \mathit{arid},\langle\rangle,\mathit{empty}\rangle,\nu>\#\rho[\![\mathsf{rec}]\!]\rightarrow\pi,$$
$$\langle \mathit{revert}(\pi'\downarrow 1)\rho,\mathit{pop}(\pi'\downarrow 2)\upsilon,\pi'\downarrow 3\rangle)$$
$$(\rho[\![\mathsf{rec}]\!]\downarrow(\#\rho[\![\mathsf{rec}]\!]-\nu+1)\downarrow 1).$$

On quitting a block we return as closely as possible to the state pertaining before entering it by activating

$$\mathit{remit}=\lambda\zeta\rho\upsilon\sigma.(\lambda\pi'.(\lambda\sigma''.\zeta(\mathit{revert}(\pi'\downarrow 1)\rho)(\langle\upsilon\downarrow 1\rangle\,\S\,\mathit{pop}(\pi'\downarrow 2)\upsilon)\sigma'')$$
$$(\mathit{restore}(\upsilon\downarrow 1:\mathsf{L}\rightarrow\mathit{update}(\upsilon\downarrow 1)\mathit{dummy}(\pi'\downarrow 3),\pi'\downarrow 3)\sigma))$$
$$(\rho[\![\mathsf{rec}]\!]\downarrow 1\downarrow 1).$$

In practice the role of *pop* in *remit* is nugatory when the primitive is used at the end of an expression, since the height of the stack at this point inevitably exceeds that on entry by one. However it is important to enlarge the area to include $\upsilon\downarrow 1$ when $\upsilon\downarrow 1$ is a location, because we shall permit an expression to return a location as a result provided its content could have been

declared before entering the relevant block; under these circumstances we shall have to ensure that $plot\alpha(revert\rho_0\rho_1)(pop\upsilon_0\upsilon_1)\sigma_1$ will be *true* for every α having $plot\alpha(revert\rho_0\rho_1)(\langle hold(\upsilon_1\downarrow 1)\sigma_1\rangle \S pop\upsilon_0\upsilon_1)\sigma_1$ equal to *true*.

When storage is automatically being deleted on leaving an expression there is little virtue in incurring the overheads of garbage collection. The equations of appendix 3 therefore employ *new*, although 3.3.9 applies equally to ones employing *novel*; we shall not bother to make *mv* and *sv* take account of this. Moreover we can set up a continuous version of *novel* for which *novel* stack semantics does not require discontinuous continuations.

Because the flow of control must be capable of jumping out of expressions (but not into them) the value of a label must consist of more than a translation of the program from the point at which the label is set onwards. Before execution is allowed to resume there the environment and stack pointers must be returned to their original values, and for the sake of economy the store is attenuated also. Accordingly we now let J be Z°, for which $\mathscr{P}[\![I:E]\!]$ and $\mathscr{Q}[\![I::E]\!]$ are

$\lambda\zeta\rho\upsilon\sigma.\langle \lambda\rho'\upsilon'\sigma'.\mathscr{S}[\![E]\!]\zeta(revert\rho\rho')(pop\upsilon\upsilon')(restore\sigma\sigma')\rangle \S\mathscr{P}[\![E]\!]\zeta\rho\upsilon\sigma$ and

$\lambda\zeta\rho\upsilon\sigma.\langle \lambda\rho'\upsilon'\sigma'.\mathscr{S}[\![E]\!]\zeta(revert\rho\rho')(pop\upsilon\upsilon')(restore\sigma\sigma')\rangle \S\mathscr{Q}[\![E]\!]\zeta\rho\upsilon\sigma$

respectively. Similarly, to deal with val we put a variant of *remit* into the res component of the environment. In contrast to the situation at the end of a block the presence of *pop* in *remit* can now be crucial, as res E may terminate the evaluation of a nest of expressions, thereby cutting back the stack.

This approach is not quite adequate for function closures, however: although a lower level in the environment may be called for when a function is applied it cannot be obtained purely by subjecting the current state to *revert*, as higher levels of the environment may be needed on leaving the function. Thus when F

is 0° we might regard $\mathscr{F}[\![\mathsf{fn}()E]\!]$, for instance, as being

$\lambda\rho.\lambda\zeta'\rho'\upsilon'\sigma'.(\lambda\rho''.\upsilon'\downarrow 1\,|\,L^*=() \rightarrow \mathscr{L}[\![E]\!](\mathit{remit}\zeta')\rho''(\upsilon'\dagger 1)\sigma',\tau)$

$(\mathit{divert}\rho'(\mathit{rend}[\![\mathsf{fn}()E]\!](\mathit{revert}\rho\rho'))[\langle \rho',\upsilon'\dagger 1,\sigma'\rangle //\mathsf{rec}])$,

so that during the application of fn()E the environment would rise above the level prevalent in the surrounding code. Unfortunately this choice of operator would make an abstraction set up in E and passed out beyond it demand an environment level exceeding that available; 3.1.2 will illustrate the folly of doing this.

Accordingly we are obliged to take $\mathscr{F}[\![\mathsf{fn}()E]\!]$ to be

$\lambda\rho.\lambda\zeta'\rho'\upsilon'\sigma'.(\lambda\rho''.\upsilon'\downarrow 1\,|\,L^*=() \rightarrow \mathscr{L}[\![E]\!](\mathit{remit}\zeta')\rho''(\upsilon'\dagger 1)\sigma',\tau)$

$(\mathit{divert}\rho'(\mathit{rend}[\![\mathsf{fn}()E]\!]\rho)[\langle \rho',\upsilon'\dagger 1,\sigma'\rangle //\mathsf{rec}])$,

which cannot bring such consequences in its train. At first sight this operator does not seem to conform to the principle that we can always keep pointers instead of spare copies of the environment; to make it (and its analogues for the other kinds of abstraction) do so we modify the structure of U laid down in 1.3.2. In the environment appropriate to stack semantics we shall hold members of N as well as members of D in order to isolate which declaration of I is referred to by $\rho[\![I]\!]\downarrow 1$. Because the free variables of an abstraction are always set up in the blocks surrounding the abstraction, instead of keeping their denotations as part of the corresponding element of F we need only keep these pointers; we have desisted from doing so in the operator above merely to be more concise.

To note the incarnation of I to which $\rho[\![I]\!]\downarrow 1$ refers the environment has simply to preserve the height of $\rho[\![I]\!]$ alongside the denotation. For the purposes of 3.2.8, however, we wish to specify precisely the block in which the relevant declaration takes place. This can be done by including in the environment layer for I another member of N, which represents the depth of

nesting of the block, so that we actually set
U=[Ide↔[D×N×N]*]×[J×N×N]*×[P×N×N]*. We append δ:D to ρ:U at I:Ide using the convention that
ρ[δ//I]=ρ[⟨δ,#ρ⟦rec⟧+1,#ρ⟦I⟧+1⟩/I]; likewise when ζ:J
ρ[ζ//res]=ρ[⟨ζ,#ρ⟦rec⟧+1,#ρ⟦res⟧+1⟩/res] and when π:P
ρ[π//rec]=ρ[⟨π,#ρ⟦rec⟧+1,#ρ⟦rec⟧+1⟩/rec]. The depth of nesting components, written as #ρ⟦rec⟧+1, are not essential to the equations of appendix 3 and can be eliminated in an obvious manner, but some of them will be required by the predicates of 3.2.5. We cannot erase all mention of #ρ⟦I⟧+1, on the other hand, as within declarations allow an identifier to have more than one meaning in a block although the height of ρ⟦rec⟧ is unchanged. For any δ*:D* we set
ρ[δ*//I*]=(I*=⟨⟩→ρ,(ρ[δ*↑1//I*↑1])[δ*↓1//I*↓1]), and we make the tacit assumption that #*dummy**=#I* whenever we write ρ[*dummy**//I*] in our equations.

We supplement the primitives of 1.3.2 with *ravel*, which rearranges the denoted values in the order prescribed by their markers. The values found in the environment where the markers first come to light are those to which significance is attached; all the others are viewed as meaningless accretions which would not be present in an implementation. For any ρ and I we examine the set of all ν:N such that ρ⟦I⟧↓ν↑1=ρ⟦I⟧↓1↑1; the maximal element of this set will be nearest the bottom of the environment and will therefore yield that component of the form ρ⟦I⟧↓ν↓1 which is of interest. This integer is yielded by
lead=λνω*.*fix*(λφν'ν".ν">#ω*→ν', φ((ω*↓ν"↑1=ω*↓ν↑1)→ν",ν')(ν"+1))ν1;
ravel=$\lambda\rho_0\rho_1$.(λφ.⟨λI.φ(ρ_0⟦I⟧)(ρ_1⟦I⟧),φ(ρ_0⟦res⟧)(ρ_1⟦res⟧),⟨⟩⟩)
($\lambda\omega^*_0\omega^*_1$.*fix*(λφν.ν>#$\omega^*_0$→⟨⟩,⟨$\omega^*_1$↓*lead*ν$\omega^*_0$⟩ §φ(ν+1))1)
aligns every integer with the denotation to which it corresponds.

Consequently $ravel\rho\rho[\![I]\!]\downarrow 1\downarrow 1$ is the entity required by the semantic equations. In practice $lead\nu(\rho[\![I]\!])$ is $\#\rho[\![I]\!]-\rho[\![I]\!]\downarrow\nu\downarrow 3+1$, but we shall never never need to presume that this is the case and our primitives will be unaffected by the nature of $\rho[\![I]\!]\downarrow\nu\dagger 2$. There are of course many possible ways of varying the construction for U adopted here; one of the more important will be analysed in 3.6.3, where modes will be discussed.

We can now set up an implementation of recursion by tying a knot through the environment in exactly the same way as we did through the store in 1.3.4. After the recursive declaration rec I==E has been executed there may be many functions into which have been compiled integral markers indicating where the outcome of the declaration has been put. When I is invoked during the application of one of these functions *ravel* will ensure that this outcome will be picked up as the value of I. If δ:D we write

$\rho[\delta///I]=(\lambda\omega^*.\langle(\lambda I'.I'=I\rightarrow\omega^*,\rho[\![I']\!]),\rho[\![res]\!],\rho[\![res]\!]\rangle)$

$(fix(\lambda\phi\nu.\nu>\#\rho[\![I]\!]\rightarrow\langle\rangle,$

$\langle((\nu=lead1(\rho[\![I]\!]))\rightarrow\langle\delta\rangle\S\rho[\![I]\!]\downarrow\nu\dagger 1,\rho[\![I]\!]\downarrow\nu)\rangle\S\phi(\nu+1))1).$

In terms of this $\mathscr{D}[\![$rec I==E$]\!]$ becomes $\lambda\zeta\rho\upsilon\sigma.\mathscr{R}[\![E]\!](\lambda\rho'\upsilon'\sigma'.\zeta\rho'[\upsilon'\downarrow 1///I](\upsilon'\dagger 1)\sigma')\rho[dummy//I]\upsilon\sigma$; 3.3.8 will show that this operator is indeed equivalent to that for rec I=E. The comparable equation for multiple declarations, given in appendix 3, makes use of the convention that if δ^*:D* and I*:Ide* then $\rho[\delta^*///I^*]=(I^*=\langle\rangle\rightarrow\rho,(\rho[\delta^*\dagger 1///I^*\dagger 1])[\delta^*\downarrow 1///I^*\downarrow 1])$. It is the absence of *fix* from this equation and from that for labels that enables us to construct the environment from [D×N×N]* instead of [D°×N×N]*.

Notice that if we start the execution of a program in a suitable library environment the primitives provided here always

append additional layers in an orderly fashion. More exactly, if ρ the environment at any stage then $tidy\rho\rho=true$ where

$$tidy=\lambda\rho_0\rho_1.\wedge\{(\lambda\phi.\wedge\{\phi(\rho_0[\![I]\!])(\rho_1[\![I]\!))\,|\,I:Ide\}\wedge\phi(\rho_0[\![res]\!])(\rho_1[\![res]\!])$$
$$\wedge((level\nu\langle\rho_0,\langle\rangle empty\rangle\downarrow 1)[\![rec]\!]=\nu-1))$$
$$(\lambda\omega^*_0\omega^*_1.\sim(1\le\nu\le\#\omega^*_0)\to true,$$
$$(1\le\omega^*_0\downarrow\nu\downarrow 2\le\#\rho_0[\![rec]\!]+1)\wedge(lead\nu(\omega^*_0\S\omega^*_1)>\#\omega^*_0))$$
$$|\nu:N\}.$$

3.1.2. Example.

Let E_0 be x=dummy inside f=fn()fn()x inside E_1 and let E_1 be x=nil inside $(f(x))x$. Under the first abstraction operator put forward in 3.1.1 E_0 does not return the answer *dummy*.

Take any proper ρ_0, υ_0 and σ_0 such that $news^4\sigma_0$ is proper. Define $\alpha_1=new\sigma_0$, $\sigma_1=update\alpha_1 dummy\sigma_0$, $\rho_1=\rho_0[\langle\rho_0,\upsilon_0,\sigma_0\rangle//rec][\alpha_1//x]$, $\alpha_2=new\sigma_1$, $\sigma_2=update\alpha_2(\mathscr{F}[\![fn()fn()x]\!]\rho_1)\sigma_1$, $\rho_2=\rho_1[\langle\rho_1,\upsilon_0,\sigma_1\rangle//rec][\alpha_2//f]$, $\alpha_3=new\sigma_2$, $\sigma_3=update\alpha_3\langle\rangle\sigma_2$ and $\rho_3=\rho_2[\langle\rho_2,\upsilon_0,\sigma_2\rangle//rec][\alpha_3//x]$. For any ζ_0 write $\zeta_{n+1}=remit\zeta_n$ when $0\le n\le 2$, so that

$$\mathscr{P}[\![E_0]\!]\zeta_0\rho_0\upsilon_0\sigma_0=\mathscr{L}[\![f=fn()fn()x\ inside\ x=nil\ inside\ (f(x))x]\!]\zeta_1\rho_1\upsilon_0\sigma_1$$
$$=\mathscr{L}[\![x=nil\ inside\ (f(x))x]\!]\zeta_2\rho_2\upsilon_0\sigma_2$$
$$=\mathscr{L}[\![(f(x))x]\!]\zeta_3\rho_3\upsilon_0\sigma_3.$$

Suppose that evaluation takes place from right to left, and define $\alpha_4=new\sigma_3$, $\sigma_4=update\alpha_4(\mathscr{F}[\![fn()x]\!]\rho_4)\sigma_3$, $\rho_4=divert\rho_3(rend[\![fn()fn()x]\!](revert\rho_1\rho_3))[\langle\rho_3,\langle\alpha_3\rangle\S\upsilon_0,\sigma_3\rangle//rec]$ and $\zeta_4=\lambda\rho\upsilon.(\upsilon\downarrow 1)\zeta_3\rho(\upsilon\dagger 1)$ to give

$$\mathscr{L}[\![(f(x))x]\!]\zeta_3\rho_3\upsilon_0\sigma_3=\mathscr{R}[\![fx]\!]\zeta_4\rho_3(\langle\alpha_3\rangle\S\upsilon_0)\sigma_3$$
$$=\mathscr{R}[\![f]\!](\lambda\rho\upsilon.(\upsilon\downarrow 1)(s\nu\zeta_4)\rho(\upsilon\dagger 1))\rho_3(\langle\alpha_3,\alpha_3\rangle\S\upsilon_0)\sigma_3$$
$$=\mathscr{F}[\![fn()fn()x]\!]\rho_1(s\nu\zeta_4)\rho_3(\langle\alpha_3,\alpha_3\rangle\S\upsilon_0)\sigma_3$$
$$=\mathscr{L}[\![fn()x]\!](remit(s\nu\zeta_4))\rho_4(\langle\alpha_3\rangle\S\upsilon_0)\sigma_3$$
$$=(s\nu\zeta_4)\rho_3(\langle\alpha_3,\alpha_3\rangle\S\upsilon_0)\sigma_3$$
$$=\mathscr{F}[\![fn()x]\!]\rho_4\zeta_3\rho_3(\langle\alpha_3\rangle\S\upsilon_0)\sigma_4.$$

As $divert\rho_3(rend[\![\,\mathsf{fn}()x\,]\!](revert\rho_4\rho_3))=\rho_3$,

$$\mathscr{F}[\![\,\mathsf{fn}()x\,]\!]\rho_4\zeta_3\rho_3(\langle\alpha_3\rangle\,\S\upsilon_0)\sigma_4=\mathscr{L}[\![\,x\,]\!](remit\zeta_3)\rho_3[\langle\rho_3,\upsilon_0,\sigma_4\rangle\,//\mathsf{rec}]\upsilon_0\sigma_4$$
$$=\zeta_3\rho_3(\langle\alpha_3\rangle\,\S\upsilon_0)\sigma_4$$
$$=\zeta_0\rho_0(\langle\alpha_3\rangle\,\S\upsilon_0)(update\alpha_3\langle\rangle(restore\sigma_0\sigma_4)).$$

When evaluated using store semantics, on the other hand, $\mathscr{E}[\![\,E_0\,]\!]\zeta_0\rho_0\upsilon_0\sigma_0$ would be $\zeta_0\rho_0(\langle\alpha_1\rangle\,\S\upsilon_0)\sigma_4$ where $hold\alpha_1\sigma_4=dummy$.⊳

3.1.3. Alternative approaches.

There is nothing sacrosanct about the version of stack semantics suggested in 3.1.1, and we have chosen it largely because it can be built up with very little new notation. Here we mention two minor variations on its principles which are physically more realistic but do not provide any fresh insights into computing. Our intention is simply to indicate that semantic equations can be used to describe an implementation in as much detail as a given application may require and that such descriptions can be validated relative to standard semantics by the means we shall discuss in 3.2.4.

A radical alteration to the equations of appendix 3 would be wrought by fusing the stack and the store. This could be achieved by arranging for every value which hitherto would have been placed on the stack to be stored in a new location (so that L would become a summand of V). Coercions would be invoked by the primitive functions to extract these erstwhile members of E when necessary, while storage would be allocated and discarded by regarding the store area in use as a contiguous array of locations. Then if $count$:L→N and $point$:N→L satisfied $point=\lambda\nu.\bigsqcup\{\alpha\,|\,count\alpha=\nu\}$ and if $sum=\lambda\sigma.\bigvee\{count\alpha\,|\,area\alpha\sigma\}$, in place of *new* and *update* we might use $\lambda\sigma.point(sum\sigma+1)$ and

$\lambda\zeta\rho\sigma.\zeta\rho(\lambda\alpha.\alpha=hold(point(sum\sigma-1))\sigma\rightarrow\langle true,hold(point(sum\sigma))\sigma\rangle,$

$\alpha=point(sum\sigma-1)\rightarrow\langle true,dummy\rangle,\langle count\alpha\leq sum\sigma-1,hold\alpha\sigma\rangle)$

respectively (here we take S to be $L\rightarrow[T\times V]$ for simplicity). On entry to a block the first empty location would be reserved as a space in which to put the answer returned. Such considerations as these would make the semantic equations reflect the truth about interpreters in an intolerably messy manner, but they would enable us to express the validity of a display [4] which calculates denotations by means of an offset as well as the block level; taking S to be V^* would achieve the same end.

To recover the full power of Mal we could introduce an additional region of storage from which locations would not be erased on leaving blocks. Thus suppose S were $L\rightarrow[T\times T\times V]$, where the first lattice of truth values indicated which kind of storage was intended; instead of *restore* we would require $\lambda\sigma_0\sigma_1.\lambda\alpha.\langle\sigma_1\alpha\downarrow1,(\sigma_1\alpha\downarrow1\rightarrow\sigma_1\alpha\downarrow2,\sigma_0\alpha\downarrow2),\sigma_1\alpha\downarrow3\rangle$. Before passing functions and labels beyond their scopes we would have to give them representations as stored values incorporating more information than would be necessary as denoted values. A suitable substitute for *update* would be

$\lambda\zeta\rho\upsilon\sigma.(\lambda\beta.\zeta\rho(\langle dummy\rangle\,\S\upsilon\uparrow2)(\lambda\alpha.\alpha=\upsilon\downarrow2\rightarrow\langle\sigma\downarrow1,\sigma\downarrow2,\beta\rangle,\sigma\alpha))$

$(\upsilon\downarrow1:Z^{\circ}\wedge\sigma(\upsilon\downarrow2)\downarrow1\rightarrow\langle\upsilon\downarrow1,\rho,\upsilon\uparrow2\rangle,\upsilon\downarrow2:0^{\circ}\wedge\sigma(\upsilon\downarrow2)\downarrow1\rightarrow\langle\upsilon\downarrow1,\rho\rangle,\upsilon\downarrow1).$

We could adapt the proof of 3.3.9 to establish the equivalence of store semantics and equations based on these notions, but we prefer to defer all further discussion of the heap until 3.5.3.

3.1.4. Syntactic constraints enforcing validity.

To indicate which Mal programs can be executed using the mechanism of 3.1.1 we use context-dependent predicates which test the identifiers at the exits of an expression to verify that they

are not local to the block. We also insist that in an assignment E_0:=E_1 no exits of E_0 denoting locations have wider scope than the outcome of E_1; only this way can we ensure that a location will not be accessed outside its extent. Such checks would not be necessary were we to confine our remarks to a little language like Algol 60 [13], which shrinks V into B and does not permit blocks to return procedures as results, but here they are crucial. In fact we even require more sophisticated tests than those of 1.5.3, where we were content to distinguish between local variables and three varieties of global variable.

The predicates we adopt take as arguments a program, an integer (yielding the level of the block to which we revert after running the program) and an environment; more precisely, corresponding to the four major valuations we introduce *e*:Exp→N→U→T, *g*:Exp→N→U→T, *d*:Dec→N→U→T and *t*:Dec→N→U→T. The members of U thus invoked associate with each identifier the block level of its most recent declaration, thereby restricting what can stand as an expression exit. We also differentiate between those denotations which are locations and those which are not by providing ρ:U with some α:L in the first case and some β:V in the second. To increase the depth of nesting of the blocks we adjoin to ρ any state π_0:P which can be chosen at will. The dependence of the predicates on operations performed during the execution of the program is chimerical, because for any declaration (whether by incidence or by reference) we can determine how many blocks surround it purely by examining the text of the program. Consequently a compiler could test programs to see whether they satisfied these constraints. It could not, however, predict the sizes of the stacks needed because we do not require members of L* to have denoted constituents; were we to do so we would effectively be

eliminating arrays with dynamically varying bounds in favour of those with fixed bounds.

Before discussing the predicates we shall set down a recursive definition of them thus:

$e[\![E]\!] = \lambda\nu\rho.\, g[\![E]\!](\nu\wedge(\#\rho[\![\mathrm{rec}]\!]+1))\rho[\pi_0//\mathrm{rec}][\alpha^*//\mathscr{g}[\![E]\!]][\beta^*//\mathscr{h}[\![E]\!]];$

$g[\![I]\!] = \lambda\nu\rho.\, \#\rho[\![I]\!] > 0 \to \nu \geq \rho[\![I]\!]\downarrow 1\downarrow 2, \mathit{false};$

$g[\![B]\!] = \lambda\nu\rho.\, \mathit{true};$

$g[\![\mathrm{fn}()E]\!] = \lambda\nu\rho.\, ((\#\rho[\![\mathrm{res}]\!] > 0 \to \nu \geq \rho[\![\mathrm{res}]\!]\downarrow 1\downarrow 2, \mathit{false}) \vee \sim \mathit{free}[\![E]\!][\![\mathrm{res}]\!])$

$\wedge\bigwedge\{g[\![I]\!]\nu\rho \vee \sim \mathit{free}[\![E]\!][\![I]\!] \mid I:\mathrm{Ide}\}$

$\wedge e[\![E]\!]\nu\rho;$

$g[\![\mathrm{fn}I.E]\!] = \lambda\nu\rho.\, \mathit{false};$

$g[\![\mathrm{fn}I_1,\ldots,I_n.E]\!] = \lambda\nu\rho.\, \mathit{false};$

$g[\![\mathrm{fn}I..E]\!] = \lambda\nu\rho.\, ((\#\rho[\![\mathrm{res}]\!] > 0 \to \nu \geq \rho[\![\mathrm{res}]\!]\downarrow 1\downarrow 2, \mathit{false}) \vee \sim \mathit{free}[\![E]\!][\![\mathrm{res}]\!])$

$\wedge\bigwedge\{g[\![I]\!]\nu\rho \vee \sim \mathit{free}[\![E]\!][\![I]\!] \mid I:\mathrm{Ide}\}$

$\wedge e[\![E]\!](\nu\wedge(\#\rho[\![\mathrm{rec}]\!]+1))\rho[\pi_0//\mathrm{rec}][\beta//I];$

$g[\![\mathrm{fn}I_1,\ldots,I_n..E]\!] = \lambda\nu\rho.\, ((\#\rho[\![\mathrm{res}]\!] > 0 \to \nu \geq \rho[\![\mathrm{res}]\!]\downarrow 1\downarrow 2, \mathit{false}) \vee \sim \mathit{free}[\![E]\!][\![\mathrm{res}]\!])$

$\wedge\bigwedge\{g[\![I]\!]\nu\rho \vee \sim \mathit{free}[\![E]\!][\![I]\!] \mid I:\mathrm{Ide}\}$

$\wedge e[\![E]\!](\nu\wedge(\#\rho[\![\mathrm{rec}]\!]+1))\rho[\pi_0//\mathrm{rec}][\beta^*//(I_1,\ldots,I_n)];$

$g[\![\mathrm{O}E]\!] = \lambda\nu\rho.\, e[\![E]\!](\#\rho[\![\mathrm{rec}]\!]+1)\rho;$

$g[\![E_0 \Omega E_1]\!] = \lambda\nu\rho.\, e[\![E_0]\!](\#\rho[\![\mathrm{rec}]\!]+1)\rho \wedge e[\![E_1]\!](\#\rho[\![\mathrm{rec}]\!]+1)\rho;$

$g[\![E_0 := E_1]\!] = \lambda\nu\rho.\, (\lambda\nu_0.\,(\lambda\nu_1.\, e[\![E_0]\!]\nu_0\rho \wedge e[\![E_1]\!]\nu_1\rho)$

$\wedge\{I:\mathit{exit}[\![E_0]\!] \to (\mathit{ravel}\rho\rho[\![I]\!]\downarrow 1\downarrow 1 : L \to \rho[\![I]\!]\downarrow 1\downarrow 2, \nu_0), \nu_0$

$\mid I:\mathrm{Ide}\})(\#\rho[\![\mathrm{rec}]\!]+1))$

$\wedge(E_0$ has no exits of the form get E, val E,

Δ inside E or $E_2 E_3$);

$g[\![E_1,\ldots,E_n := E_0]\!] = \lambda\nu\rho.\, (\lambda\nu_0.\,(\lambda\nu_1.\, e[\![E_1]\!]\nu_0\rho \wedge \ldots \wedge e[\![E_n]\!]\nu_0\rho \wedge e[\![E_0]\!]\nu_1\rho)$

$\wedge\{\sim I:\mathit{exit}[\![E_1]\!] \S \ldots \S \mathit{exit}[\![E_n]\!] \to \nu_0,$

$\mathit{ravel}\rho\rho[\![I]\!]\downarrow 1\downarrow 1 : L \to \rho[\![I]\!]\downarrow 1\downarrow 2, \nu_0 \mid I:\mathrm{Ide}\})$

$(\#\rho[\![\mathrm{rec}]\!]+1)$

$\wedge(E_m$ has no exits of the form get E,

val E, Δ inside E or $E_2 E_3$ when $1 \leq m \leq n$);

$g[\![\text{get } L]\!] = e[\![E]\!];$

$g[\![\text{put } E]\!] = e[\![E]\!];$

$g[\![E_0 \text{ aug } E_1]\!] = \lambda\rho\nu.e[\![E_0]\!]\nu\rho \wedge e[\![E_1]\!]\nu\rho;$

$g[\![E_1,\ldots,E_n]\!] = \lambda\rho\nu.e[\![E_1]\!]\nu\rho \wedge \ldots \wedge e[\![E_n]\!]\nu\rho;$

$g[\![\$E]\!] = e[\![E]\!];$

$g[\![E\$]\!] = e[\![E]\!];$

$g[\![\&E]\!] = e[\![E]\!];$

$g[\![E\&]\!] = e[\![E]\!];$

$g[\![E_0E_1]\!] = \lambda\rho\nu.g[\![E_0]\!]\nu\rho \wedge g[\![E_1]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$g[\![\text{val } E]\!] = \lambda\nu\rho.e[\![E]\!](\nu\wedge(\#\rho[\![\text{rec}]\!]+1))\rho[\pi_0//\text{rec}][\perp//\text{res}];$

$g[\![\text{res } E]\!] = \lambda\nu\rho.e[\![E]\!]0\rho;$

$g[\![\text{goto } E]\!] = \lambda\nu\rho.e[\![E]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$g[\![\Delta \text{ inside } L]\!] = \lambda\nu\rho.e[\![E]\!](\nu\wedge(\#\rho[\![\text{rec}]\!]+1))\rho[\pi_0//\text{rec}][\alpha^*//\mathcal{S}[\![\Delta]\!]][\beta^*//\mathcal{R}[\![\Delta]\!]]$
$\wedge d[\![\Delta]\!]0\rho;$

$g[\![E_0;\ E_1]\!] = \lambda\nu\rho.g[\![E_0]\!](\#\rho[\![\text{rec}]\!]+1)\rho \wedge g[\![E_1]\!]\nu\rho;$

$g[\![\text{if } E_0 \text{ then } E_1 \text{ else } E_2]\!] = \lambda\nu\rho.e[\![E_0]\!](\#\rho[\![\text{rec}]\!]+1)\rho \wedge g[\![E_1]\!]\nu\rho \wedge g[\![E_2]\!]\nu\rho;$

$g[\![\text{while } E_0 \text{ do } E_1]\!] = \lambda\nu\rho.e[\![E_0]\!](\#\rho[\![\text{rec}]\!]+1)\rho \wedge g[\![E_1]\!](\#\rho[\![\text{rec}]\!]+1);$

$g[\![I:E]\!] = g[\![E]\!];$

$g[\![I::E]\!] = g[\![L]\!];$

$g[\![(E)]\!] = g[\![E]\!];$

$d[\![I=E]\!] = \lambda\nu\rho.\mathit{false};$

$t[\![I=E]\!] = \lambda\nu\rho.e[\![E]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$d[\![I_1,\ldots,I_n=E]\!] = \lambda\nu\rho.\mathit{false};$

$t[\![I_1,\ldots,I_n=E]\!] = \lambda\nu\rho.e[\![E]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$d[\![I==E]\!] = \lambda\nu\rho.e[\![E]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$t[\![I==E]\!] = \lambda\nu\rho.e[\![E]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$d[\![I_1,\ldots,I_n==E]\!] = \lambda\nu\rho.e[\![E]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$t[\![I_1,\ldots,I_n==E]\!] = \lambda\nu\rho.e[\![E]\!](\#\rho[\![\text{rec}]\!]+1)\rho;$

$d[\![\Delta_0 \text{ within } \Delta_1]\!] = \lambda\nu\rho.d[\![\Delta_0]\!]0\rho \wedge d[\![\Delta_1]\!]0\rho[\alpha^*//\mathcal{S}[\![\Delta_1]\!]][\beta^*//\mathcal{R}[\![\Delta_1]\!]];$

$t[\![\Delta_0$ within $\Delta_1]\!] = \lambda\nu\rho.\, d[\![\Delta_0]\!]0\rho \wedge t[\![\Delta_1]\!]0\rho[\alpha^*//\mathscr{S}[\![\Delta_1]\!]][\beta^*//\mathscr{X}[\![\Delta_1]\!]]$
$\wedge(\mathscr{X}[\![\Delta_0]\!] \S \mathscr{S}[\![\Delta_1]\!]$ and $\mathscr{S}[\![\Delta_0]\!] \S \mathscr{X}[\![\Delta_0]\!] \S \mathscr{X}[\![\Delta_1]\!]$
have no repeated elements);

$d[\![\Delta_1$ and...and $\Delta_n]\!] = \lambda\nu\rho.\, d[\![\Delta_1]\!]\nu\rho \wedge \ldots \wedge d[\![\Delta_n]\!]\nu\rho$;

$t[\![\Delta_1$ and...and $\Delta_n]\!] = \lambda\nu\rho.\, t[\![\Delta_1]\!]\nu\rho \wedge \ldots \wedge t[\![\Delta_n]\!]\nu\rho$;

$d[\![$rec $\Delta]\!] = \lambda\nu\rho.\, t[\![\Delta]\!]\nu\rho[\alpha^*//\mathscr{S}[\![\Delta]\!]][\beta^*//\mathscr{X}[\![\Delta]\!]]$;

$t[\![$rec $\Delta]\!] = t[\![\Delta]\!]$;

$d[\![(\Delta)]\!] = d[\![\Delta]\!]$;

$t[\![(\Delta)]\!] = t[\![\Delta]\!]$.

Owing to the difficulty in deciding whether fn$u$.u:=v, say, satisfies the condition imposed on assignment we forbid the presence of abstractions by reference. It would be possible to allow those abstractions which did not contain assignments to their formal parameters, but the necessary changes to the predicates are extensive and do not yield a wider class of computations: if I is not covertly assigned to in E 2.5.3 establishes that E is equivalent to $e[\![E]\!](\lambda I'.I'=I \rightarrow \mathit{false}, \mathit{true})$, and thus a version of 2.5.4 confirms that fnI.E can be replaced by fnI..E, which we allow in any case. Similar considerations govern declarations by reference which are not recursive, for unless I=E is equivalent to I=$E it may later give rise to an illegitimate assignment. As recursive declarations are still permitted x=y, say, can be replaced by rec x=y.

Abstractions by incidence must be severely constrained because they are used both in function applications and as expression exits. A failure to confine their free variables to those which can be passed out of the surrounding context can lead to a disaster like 3.1.5.

It is convenient to prohibit get E, val E and Λ inside E from appearing as exits on the left hand sides of assignment statements. We could devise a function which would list all the 'secondary' exits from the exits of an expression; were this to

be substituted for *exit* in the predicates we could remove the prohibition, but the effort would not justify the outcome. More unfortunate is the interdiction on the appearance of E_2E_3 among the exits of the left hand side of $E_0:=E_1$, as this eliminates such programs as x=nil aug 0 inside $x1:=0$. However even it could be removed by determining all the exits of abstractions in E_2 as well as of E_2 itself; in particular our predicates could be extended to cover the most important case, when E_2 is an identifier.

Another category of programs which is needlessly excluded is typified by (l: dummy; l); dummy, in which a value is passed out of scope but no ill-effects arise because it is immediately discarded. Again we could easily modify the predicates (and the proof of 3.2.8) to deal with this category but the ends do not justify the means.

3.1.5. Example.

Let E_0 be x=0 inside ((1+(x:=val (fn$z$.res 2); 1); E_1) and let E_1 be if x=2 then x else $x(x)$. When evaluated using stack semantics E_0 does not return the answer 2.

◁Take any proper ρ_0 and σ_0 and define $\alpha_0=new\sigma_0$, $\rho_1=\rho_0[\langle\rho_0,\langle\rangle,\sigma_0\rangle//rec][\alpha_0//x]$, $\sigma_1=update\alpha_0 0\sigma_0$, $\alpha_1=new\sigma_1$, $\pi_2=\langle\rho_1,\langle\alpha_0,1\rangle,\sigma_1\rangle$, $\rho_3=\rho_1[\pi_2//rec][\lambda\rho.mv(remit\zeta_3)\rho[\pi_2//rec]//res]$, $\phi_0=\mathscr{F}[\![$ fn$z$.res 2$]\!]\rho_3$, $\sigma_3=updates\langle\alpha_0,\alpha_1\rangle\langle\phi_0,\phi_0\rangle\sigma_1$, $\alpha_2=new\sigma_3$, $\rho_4=divert\rho_1(rend[\![$ res 2$]\!]\rho_3)[\langle\rho_1,\langle\rangle,\sigma_3\rangle//rec][\phi_0//z]$ and $\sigma_4=update\alpha_2 2\sigma_3$. For any ζ_0 set $\zeta_1=\lambda\rho\upsilon.\mathscr{F}[\![E_1]\!](mv(remit\zeta_0))\rho(\upsilon\dagger 1)$, $\zeta_2=\lambda\rho\upsilon.sv\zeta_1(\langle 1+\upsilon\downarrow 2\rangle\,\S\,\upsilon\dagger 3)$ and $\zeta_3=sv(\lambda\rho\upsilon.\zeta_2\rho(\langle dummy\rangle\,\S\,\upsilon\dagger 2)\circ update(\upsilon\downarrow 2)(\upsilon\downarrow 1))$. Ignoring the alterations to $\rho[\![$rec$]\!]$ caused by using $\mathscr{E}$ instead of $\mathscr{F}$,

$$
\begin{aligned}
\mathcal{S}[\![E_0]\!]\zeta_0\rho_0\langle\rangle\sigma_0 &= \mathcal{S}[\![1+(x:=\mathrm{val}\ (\mathrm{fn}z..\mathrm{res}\ 2);\ 1)]\!]\zeta_1\rho_1\langle\rangle\sigma_1\\
&= \mathcal{S}[\![x:=\mathrm{val}\ (\mathrm{fn}z..\mathrm{res}\ 2)]\!]\zeta_2\rho_1\langle 1\rangle\sigma_1\\
&= \mathcal{S}[\![\mathrm{val}\ \mathrm{fn}z..\mathrm{res}\ 2]\!]\zeta_3\rho_1\langle\alpha_0,1\rangle\sigma_1\\
&= \mathcal{S}[\![\mathrm{fn}z..\mathrm{res}\ 2]\!](mv(remit\zeta_3))\rho_3\langle\alpha_0,1\rangle\sigma_1\\
&= \zeta_2\rho_2\langle 1\rangle\sigma_3\\
&= \zeta_1\rho_1\langle 2\rangle\sigma_3\\
&= \mathcal{S}[\![x(x)]\!](mv(remit\zeta_0))\rho_1\langle\rangle\sigma_3\\
&= \phi_0(mv(remit\zeta_0))\rho_1\langle\alpha_0\rangle\sigma_3\\
&= \mathcal{S}[\![\mathrm{res}\ 2]\!](mv(remit(mv(remit\zeta_0))))\rho_4\langle\rangle\sigma_3\\
&= mv(remit\zeta_3)\rho_4[\pi_2//\mathrm{rec}]\langle 2\rangle\sigma_3\\
&= \zeta_3\rho_1\langle\alpha_3\rangle\sigma_4\\
&= \tau.
\end{aligned}
$$

Other forms of semantic equation yield the intuitive meaning of the program, in which a location containing 2 is returned as the result.➤

3.2. Preparations for an inductive proof.

3.2.1. Locations accessible from outer blocks.

Here we shall trace out the relation between evaluating a program using *new* stack semantics and evaluating its transform under the rules of 1.4.6 using *new* store semantics. Because the semantic equations of appendix 3 regard functions and label entry points as consisting purely of code they do not provide all the information we shall require to construct this relation. In particular they do not explicitly indicate what area of store is to be retained when a jump is made nor which markers are to be adjoined to the environment when a function is applied. We therefore deviate slightly from our intention not to preserve portions of the state vector distinct from it by taking J to be $Z^{\circ}\times N$ and F to be $0^{\circ}\times U$. Thus both $\mathscr{P}[\![\,\mathrm{I:E}\,]\!]\zeta\rho\upsilon\sigma\downarrow 1$ and $\mathscr{Q}[\![\,\mathrm{I::E}\,]\!]\zeta\rho\upsilon\sigma\downarrow 1$ are now $\langle\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![E]\!]\zeta(\mathit{revert}\rho\rho')(\mathit{pop}\upsilon\upsilon')(\mathit{restore}\sigma\sigma'),\#\rho[\![\mathrm{rec}]\!]+1\rangle$, while $\mathscr{F}[\![\,\mathrm{fn()E}\,]\!]$ is

$$\lambda\rho.\langle\lambda\zeta'\rho'\upsilon'\sigma'.(\lambda\rho''.\upsilon'\downarrow 1\,|\,L^{*}=\langle\rangle\rightarrow\mathscr{G}[\![\,\mathrm{E}\,]\!](\mathit{remit}\zeta')\rho''(\upsilon'\dagger 1)\sigma',\mathrm{T})$$
$$(\mathit{divert}\rho'(\mathit{rend}[\![\,\mathrm{fn()E}\,]\!]\rho)[\langle\rho',\upsilon'\dagger 1,\sigma'\rangle//\mathrm{rec}]),$$
$$\mathit{rend}[\![\,\mathrm{fn()E}\,]\!]\rho\rangle\ ;$$

analogous modifications are made to $\mathscr{G}[\![\,\mathrm{val\ E}\,]\!]$ and to $\mathscr{F}[\![\,\Phi\,]\!]$ when Φ is any other form of abstraction. These components play no part in the semantic equations, so that $\mathscr{G}[\![\,\mathrm{goto\ E}\,]\!]$, for instance, is now $\mathscr{R}[\![\,\mathrm{E}\,]\!]\circ(\lambda\zeta\rho\upsilon.(\upsilon\downarrow 1\downarrow 1)\rho(\upsilon\dagger 1))$; consequently an easy proof (vaguely reminiscent of that of 2.3.9) suffices to show that these equations are essentially equivalent to those of appendix 3, which do not contain so many superfluities.

Even within the predicates the roles of N in J and of U in F will be limited to supplying members of $[\mathrm{Ide}\rightarrow N]\times N\times N$, N and $L\rightarrow T$. To illustrate this we now give the basic correspondence between the state vectors $\acute{\pi}$ and $\grave{\pi}$ arising from a program and its transform. Owing to our simplified treatment of recursion W

reduces to L+B+L*+J+F+J in stack semantics, and we must arrange that the state $\hat{\pi}$, which is evaluated using store semantics, contains no members of G or P. To extract the witnessed values from the state $\hat{\pi}$ resulting from stack semantics we apply *ravel*, so that

$$hoten=\lambda\hat{\omega}\hat{\rho}.\vee\{\vee\{1\le\nu\le\#\hat{\rho}[\![I]\!]\wedge 1\le\nu\le\#\hat{\rho}[\![I]\!]\to\hat{\omega}=\langle ravel\hat{\rho}\hat{\rho}[\![I]\!]\downarrow\nu\downarrow 1,\hat{\rho}[\![I]\!]\downarrow\nu\rangle, false \mid I:Ide\}$$
$$\vee(1\le\nu\le\#\hat{\rho}[\![res]\!]\wedge 1\le\nu\le\#\hat{\rho}[\![res]\!]\to\hat{\omega}=\langle ravel\hat{\rho}\hat{\rho}[\![res]\!]\downarrow\nu\downarrow 1,\hat{\rho}[\![res]\!]\downarrow\nu\rangle, false)\mid\nu:N\};$$

$$gyven=\lambda\hat{\omega}\hat{\upsilon}.\vee\{1\le\nu\le\#\vec{\omega}\wedge 1\le\nu\le\#\vec{\omega}\to\hat{\omega}=\langle\hat{\upsilon}\downarrow\nu,\hat{\upsilon}\downarrow\nu\rangle, false\mid\nu:N\}.$$

Following 2.1.6 we set

$$access=\lambda\hat{\omega}\hat{\pi}.\langle(\vec{\omega}:L\to(area\vec{\omega}\vec{\sigma}\to hold\vec{\omega}\vec{\sigma},\tau),\vec{\omega}),(\vec{\omega}:L\to(area\vec{\omega}\vec{\sigma}\to hold\vec{\omega}\vec{\sigma},\tau),\vec{\omega})\rangle.$$

The tracing algorithm thus becomes

$$seen=\lambda\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}.\nu_1<1\to\hat{\omega}_0=\hat{\omega}_1,$$
$$\vec{\omega}_1:L\vee\vec{\omega}_1:L\to seen\nu_0(\nu_1-1)\hat{\omega}_0(access\hat{\omega}_1\hat{\pi})\hat{\pi},$$
$$\hat{\omega}_1:B\times B\to false,$$
$$\hat{\omega}_1:L^*\times L^*\to\vee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}\wedge gyven\hat{\omega}_2\hat{\omega}_1\mid\hat{\omega}_2:W\times W\},$$
$$\hat{\omega}_1:J\times J\to\vee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$$
$$\wedge(hoten\hat{\omega}_2\langle level(\vec{\omega}_1\downarrow 2)\hat{\pi}\downarrow 1,\vec{\omega}_1\downarrow 2\rangle\wedge\nu_0<3$$
$$\vee gyven\hat{\omega}_2\langle level(\vec{\omega}_1\downarrow 2)\hat{\pi}\downarrow 2,\vec{\omega}_1\downarrow 3\rangle\wedge\nu_0<2)$$
$$\mid\hat{\omega}_2:W\times W\},$$
$$\hat{\omega}_1:F\times F\to\vee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}\wedge\nu_0<3$$
$$\wedge hoten\hat{\omega}_2\langle divert\hat{\rho}(\vec{\omega}_1\downarrow 2),\vec{\omega}_1\downarrow 2\rangle\mid\hat{\omega}_2:W\times W\},$$
$$\hat{\omega}_1:J\times J\to\vee\{seen\nu_0(\nu_1-1)\hat{\omega}_0\hat{\omega}_2\hat{\pi}$$
$$\wedge(hoten\hat{\omega}_2\langle level(\vec{\omega}_1\downarrow 2)\hat{\pi}\downarrow 1,\vec{\omega}_1\downarrow 2\rangle\wedge\nu_0<3$$
$$\vee gyven\hat{\omega}_2\langle level(\vec{\omega}_1\downarrow 2)\hat{\pi}\downarrow 2,\vec{\omega}_1\downarrow 3\rangle\wedge\nu_0<2)$$
$$\mid\hat{\omega}_2:W\times W\},$$
$$false.$$

On this occasion we let

$$kent=\lambda\nu\hat{\omega}\hat{\pi}.\bigvee\{\bigvee\{seen\nu\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}$$
$$\wedge(hoten\hat{\omega}_1\hat{\rho}\vee gyven\hat{\omega}_1\delta\vee gyven\hat{\omega}_1(\sigma\downarrow 2,\partial\downarrow 2))$$
$$|\nu_1:N\}|\hat{\omega}_1:W\times W\}.$$

The results of 2.1.7 and 2.1.8 remain relevant, so that, for instance, if ν_0, ν_1, $\hat{\omega}_0$, $\hat{\omega}_1$ and $\hat{\pi}$ satisfy $kent\nu_0\hat{\omega}_1\hat{\pi}=true$ and $seen\nu_0\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}=true$ then $kent\nu_0\hat{\omega}_0\hat{\pi}=true$.

We are also interested in those parts of the state vectors which can be witnessed at levels corresponding to the heights of the environment and the stack on entering outer blocks. Accordingly we write

$$known=\lambda\hat{\pi}_0\hat{\omega}\hat{\pi}_1.\bigvee\{\bigvee\{seen3\nu_1\hat{\omega}\hat{\omega}_1\hat{\pi}_1$$
$$\wedge(hoten\hat{\omega}_1(revert\delta_0\delta_1,revert\delta_0\partial_1)$$
$$\vee gyven\hat{\omega}_1(pop\upsilon_0\upsilon_1,pop\upsilon_0\upsilon_1)\vee gyven\hat{\omega}_1(\sigma\downarrow 2,\partial\downarrow 2))$$
$$|\nu_1:N\}|\hat{\omega}_1:W\times W\},$$

for which $known\hat{\pi}\hat{\omega}\hat{\pi}=kent3\hat{\omega}\hat{\pi}$ for all $\hat{\omega}$ and $\hat{\pi}$ with $neat(ravel\beta\delta,\partial)=true$ and $\#\upsilon=\#\upsilon$. Neither *kent* nor *known* traces the values witnessed in the output stream, since they cannot affect the future course of the computation and may demand environment levels higher than the prevalent one.

Owing to the rather stringent restrictions imposed in 3.1.4 it is to be expected that the values witnessed at points in the state vector attainable from outer blocks will be such that they could themselves have been set up while these blocks were being executed. Thus if they are in J the associated environments and stacks must not be higher than those pertaining to the blocks. Such properties can be expressed in terms of the projections of 2.4.1 using

$found=\lambda\pi_0\hat{\omega}\pi_1.\acute{\omega}:L\vee\grave{\omega}:L\rightarrow(\acute{\omega}:L\rightarrow area\acute{\omega}\sigma_0,false)\wedge(\grave{\omega}:L\rightarrow area\grave{\omega}\sigma_1,true),$

$\hat{\omega}:L^*\times L^*\rightarrow\#\acute{\omega}=\#\grave{\omega}\wedge\bigwedge\{known\pi_0\hat{\alpha}\pi_1\vee\sim gyven\hat{\alpha}\hat{\omega}\,|\,\hat{\alpha}:L\times L\},$

$\hat{\omega}:J\times J\rightarrow\acute{\omega}{\downarrow}2\leq\#\acute{\rho}_0[\![\,rec\,]\!]$

$\wedge\bigwedge\{\acute{\varepsilon}:L\vee\grave{\varepsilon}:V\vee\sim gyven\langle level(\acute{\omega}{\downarrow}2)\pi_1{\downarrow}2,\grave{\omega}{\downarrow}3\rangle\,|\,\hat{\varepsilon}:E\times E\}$

$\wedge q_0\grave{\omega}{\downarrow}2=revert(level(\acute{\omega}{\downarrow}2)\pi_1{\downarrow}1)(\textcircled{t}q_0\grave{\rho}_1)$

$\wedge q_0\grave{\omega}{\downarrow}3=pop(level(\acute{\omega}{\downarrow}2)\pi_1{\downarrow}2)(\textcircled{p}q_0\grave{\upsilon}_1),$

$\hat{\omega}:F\times F\rightarrow neat\langle\acute{\omega}{\downarrow}2,\grave{\omega}{\downarrow}2\rangle\wedge tidy(\acute{\omega}{\downarrow}2)\acute{\rho}_0$

$\wedge(ravel(divert\acute{\rho}_1(\acute{\omega}{\downarrow}2))(\textcircled{u}q_0(divert\grave{\rho}_1(\grave{\omega}{\downarrow}2)))$

$=\textcircled{u}q_0(divert\grave{\rho}_1(\grave{\omega}{\downarrow}2))),$

$\hat{\omega}:J\times J\rightarrow\acute{\omega}{\downarrow}2\leq\#\acute{\rho}_0[\![\,rec\,]\!]$

$\wedge\bigwedge\{\acute{\varepsilon}:L\vee\grave{\varepsilon}:V\vee\sim gyven\langle level(\acute{\omega}{\downarrow}2)\pi_1{\downarrow}2,\grave{\omega}{\downarrow}3\rangle\,|\,\hat{\varepsilon}:E\times E\}$

$\wedge q_0\grave{\omega}{\downarrow}2=revert(level(\acute{\omega}{\downarrow}2)\pi_1{\downarrow}1)(\textcircled{t}q_0\grave{\rho}_1)$

$\wedge q_0\grave{\omega}{\downarrow}3=pop(level(\acute{\omega}{\downarrow}2)\pi_1{\downarrow}2)(\textcircled{p}q_0\grave{\upsilon}_1),$

$true.$

together with suitable versions of p_0 and q_0, which will be provided in 3.2.4.

3.2.2. Proposition.

Suppose that π_0, π_1 and π_2 satisfy $revert\acute{\rho}_2\acute{\rho}_0=revert\acute{\rho}_2\acute{\rho}_1$, $revert\acute{\rho}_2\grave{\rho}_0=revert\acute{\rho}_2\grave{\rho}_1$, $pop\acute{\upsilon}_2\acute{\upsilon}_0=pop\acute{\upsilon}_2\acute{\upsilon}_1$ and $pop\acute{\upsilon}_2\grave{\upsilon}_0=pop\acute{\upsilon}_2\grave{\upsilon}_1$. If $known\pi_2(access\hat{\omega}\pi_1)\pi_0=true$ whenever $known\pi_2\hat{\omega}\pi_0=true$ then $known\pi_2\hat{\omega}\pi_0=true$ whenever $known\pi_2\hat{\omega}\pi_1=true$.

⊲Assume that for some ν and all $\hat{\omega}_0$ and $\hat{\omega}_1$ if $seen3\nu\hat{\omega}_0\hat{\omega}_1\pi_1=true$ and $known\pi_2\hat{\omega}_1\pi_0=true$ then $known\pi_2\hat{\omega}_0\pi_0=true$. Let $\hat{\omega}_0$ and $\hat{\omega}_1$ be such that $seen3(\nu+1)\hat{\omega}_0\hat{\omega}_1\pi_1=true$ and $known\pi_2\hat{\omega}_1\pi_0=true$. If $\acute{\omega}_1:L$ or $\grave{\omega}_1:L$ $seen3\nu\hat{\omega}_0\hat{\omega}_2\pi_1=true$ and $known\pi_2\hat{\omega}_2\pi_0=true$, where $\hat{\omega}_2=access\hat{\omega}_1\pi_1$. If $\hat{\omega}_1:L^*\times L^*$ $seen3\nu\hat{\omega}_0\hat{\omega}_2\pi_1=true$ for some $\hat{\omega}_2$ such that $gyven\hat{\omega}_2\hat{\omega}_1=true$. Thus inevitably $seen3\nu\hat{\omega}_0\hat{\omega}_2\pi_1=true$ for some $\hat{\omega}_2$ having $known\pi_2\hat{\omega}_2\pi_0=true$, and by the induction hypothesis $known\pi_2\hat{\omega}_0\pi_0=true$.

Consequently for all ν, $\hat{\omega}_0$ and $\hat{\omega}_1$ when $seen3\nu\hat{\omega}_0\hat{\omega}_1\pi_1=true$

and $known\pi_2\hat{\omega}_1\hat{\pi}_0 = true$ $known\pi_2\hat{\omega}_1\hat{\pi}_0 = true$. As $hoten\hat{\omega}_1\langle revert\hat{\rho}_2\hat{\rho}_1, revert\hat{\rho}_2\hat{\rho}_1\rangle \supset known\pi_2\hat{\omega}_1\hat{\pi}_0$ and $hoten\hat{\omega}_1\langle pop\hat{\upsilon}_2\hat{\upsilon}_1, pop\hat{\upsilon}_2\hat{\upsilon}_1\rangle \supset known\pi_2\hat{\omega}_1\hat{\pi}_0$ we can conclude that $known\pi_2\hat{\omega}_0\hat{\pi}_1 \supset known\pi_2\hat{\omega}_0\hat{\pi}_0$ for all $\hat{\omega}_0$.⊁

Suppose further that when $known\pi_2\hat{\omega}\hat{\pi}_0 = true$ $access\hat{\omega}\hat{\pi}_0 = access\hat{\omega}\hat{\pi}_1$. By the argument above $known\pi_2\hat{\omega}\hat{\pi}_1 \supset known\pi_2\hat{\omega}\hat{\pi}_0$ for all $\hat{\omega}$, so when $known\pi_2\hat{\omega}\hat{\pi}_1 = true$ $access\hat{\omega}\hat{\pi}_1 = access\hat{\omega}\hat{\pi}_0$. Accordingly we can use the same argument again and assert finally that for all $\hat{\omega}$ $known\pi_2\hat{\omega}\hat{\pi}_0 = true$ if and only if $known\pi_2\hat{\omega}\hat{\pi}_1 = true$.

3.2.3. Proposition.

Let $\{\hat{\pi}_m\}$ be any sequence such that $\hat{\pi}_{m+1} \sqsupseteq \hat{\pi}_m$ for all $m \geq 0$, and suppose that for all ν and $\hat{\omega}$ $kent\nu\hat{\omega}\hat{\pi}_0$ can only be *true* if $\hat{\omega}$ is proper. Then for all ν and $\hat{\omega}$ having $kent\nu\hat{\omega}(\bigsqcup\hat{\pi}_m) = true$ there is a sequence $\{\hat{\omega}_m\}$ such that $\hat{\omega}_{m+1} \sqsupseteq \hat{\omega}_m$ for all $m \geq 0$, $kent\nu\hat{\omega}_m\hat{\pi}_m = true$ for all $m \geq 0$ and $\hat{\omega} = \bigsqcup\hat{\omega}_m$.

⊀The proof of this is a dreary induction on *seen* like that of 2.4.3 which we can ignore without detriment to the quality of life. The exact counterpart of 2.4.3 can also be established by the same means when we have set up suitable reflexive predicates.⊁

3.2.4. Their final forms.

Pleasing though it may be to have formalized an implementation of Mal with less complex domains than are needed in 2.1.1, we cannot rest content until we have confirmed that the stack valuations compute the correct answer for any program obeying the constraints of 3.1.4. This we shall do in the next section by showing that evaluating an expression under *new* stack semantics is equivalent to evaluating its transform under *new* store semantics; together with 2.6.9 this will serve to validate the equations of appendix 3. The intention of the proof is quite different from that

underlying some results on implementations [7]: loosely speaking, whereas others may be concerned to verify that two ways of removing the top element of the stack do the same thing we wish to establish that this 'thing' is the right one. As pointed out in 3.1.3, though the details of a compiler can be formulated in our framework slight variations in the organization of the environment do not by themselves seem to warrant such treatment.

Later we shall introduce our ultimate versions of the predicates of 2.4.5; the propositions about them which we shall then verify will culminate in 3.2.8, where the adoption of *remit* will be vindicated. First, however, we must set up reflexive projections suitable for use with the tracing algorithms of 3.2.1. To do this we simply omit the mappings on some components of the domain of witnessed values from the projections of 2.4.2. As W is now L+B+L*+J+F+J while J and F are Z°×N and O°×U respectively,

$q_0 = \lambda\omega.\ \omega{:}L\to\omega, \omega{:}B\to\omega, \omega{:}L^*\to\omega,$

$\omega{:}J\to\omega,$

$\omega{:}F\to\omega,$

$\omega{:}J\to\omega,$

$\perp$

and when $n\geq 0$

$q_{n+1} = \lambda\omega.\ \omega{:}L\to\omega, \omega{:}B\to\omega, \omega{:}L^*\to\omega,$

$\omega{:}J\to(\mathbb{Z}q_n{}^{\circ}\times q_{n+1})\omega,$

$\omega{:}F\to(\mathbb{P}q_n{}^{\circ}\times\mathbb{N}q_{n+1})\omega,$

$\omega{:}J\to(\mathbb{Z}q_n{}^{\circ}\times q_{n+1})\omega,$

$\perp.$

The functors we use are defined precisely as in 2.4.1 with the sole exception of that for U, which continues to ignore $\mathbb{D}$ (the functor on the domain of denotations) but is now given by

$\mathbb{U}\omega = \lambda\rho.(\lambda I.\ (\omega\times\omega\times\omega)^*(\rho[\![\,I\,]\!]), (\omega\times\omega\times\omega)^*(\rho[\![\,\mathrm{res}\,]\!]), (\mathbb{P}\omega\times\omega\times\omega)^*(\rho[\![\,\mathrm{rec}\,]\!]))$.

We shall not bother to provide the analogue of $\mathbb{U}$ appropriate to

stack semantics, but it is obvious both that such a function exists and that the projections above concur with a suitable variant of 2.4.2.

Just as before we are interested in countable conjunctions of relations which truncate their arguments using reflexive projections. Thus for every n there are p_{n+1} and a_{n+1} which together with a form of *fit* give rise to

$c_{n+1}=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge\{a_{n+1}(\mathbb{z}q_n\zeta\rho\upsilon\sigma,\mathbb{z}q_n\zeta\rho\upsilon\sigma)\mid p_{n+1}\hat{\pi}\wedge fit\hat{\pi}\hat{\pi}_0\}$

and to $c=\lambda\hat{\zeta}\hat{\pi}_0.\bigwedge c_{n+1}\hat{\zeta}\hat{\pi}_0$. Here $\mathbb{z}q_n\zeta$ refers to the projections defined above whereas $\mathbb{z}q_n\zeta$ refers to those of 2.4.1; similar remarks hold for the moieties of every other kind of pair, so that w_n, for instance, is $\lambda\hat{\omega}\hat{\pi}.w_n((q_n\times q_n)\hat{\omega})((\mathbb{P}q_n\times\mathbb{P}q_n)\hat{\pi})$.

Our definition of p_0 must offset the simplicity of the domains by being very prolix. Because label values in stack semantics do not incorporate environments it is necessary to ensure that the current state can provide them when required; in particular the list of states preserved on entering nested blocks, $\rho[\![\,\mathrm{rec}\,]\!]$, must be ordered according to the heights of the stacks (and the areas of used storage) and must be subject to the function *tidy* of 3.1.1. We also subsume in p_0 the assertion of 3.2.1 that any value obtained using *known* without leaving a portion of the state present in an earlier block has to satisfy *found* if variables are not to be passed outside their scopes. In addition the predicate retains most of the features of its counterpart in 2.6.1 and thus turns into the amorphous agglomerate

$p_0 = \lambda\hat{\pi}.\, neat(\, ravel\hat{\rho}\hat{\rho}, \hat{\rho}\,) \wedge \#\hat{\upsilon} = \#\hat{\upsilon} \wedge \#\hat{\sigma}\downarrow 2 = \#\hat{\sigma}\downarrow 2 \wedge \#\hat{\sigma}\downarrow 3 = \#\hat{\sigma}\downarrow 3$

$\wedge \bigvee\{ \bigwedge\{ area\hat{\alpha}_m\hat{\sigma} \wedge area\hat{\alpha}_m\hat{\sigma} \mid 1 \le m \le n \} \rightarrow \bigvee\{ \hat{\alpha}_m = \hat{\alpha}_l \mid 1 \le m < l \le n \}, false \mid 2 \le n \}$

$\wedge \bigwedge\{ \hat{\omega}_0 : L \wedge \hat{\omega}_0 = \hat{\omega}_1 \rightarrow \hat{\omega}_0 = \hat{\omega}_1, true \mid kent3\hat{\omega}_0\hat{\pi} \wedge kent3\hat{\omega}_1\hat{\pi} \}$

$\wedge \bigwedge\{ \omega_0\hat{\beta}\hat{\pi} \vee \sim gyven\hat{\beta}\langle \hat{\sigma}\downarrow 3, \hat{\sigma}\downarrow 3 \rangle \mid \hat{\beta} : V \times V \}$

$\wedge \bigwedge\{ (\lambda\pi_0\pi_1 . \bigwedge\{ found\pi_0\hat{\omega}\hat{\pi} \vee \sim known\pi_0\hat{\omega}\hat{\pi} \mid \hat{\omega} : W \times W \}$

$\wedge (revert\rho_0 \circ revert\rho_1 = revert\rho_0) \wedge (pop\upsilon_0 \circ pop\upsilon_1 = pop\upsilon_0)$

$\wedge (restore\sigma_0 \circ restore\sigma_1 = restore\sigma_0) \wedge tidy\rho_0\rho_0$

$(level\upsilon\hat{\pi})(level(\upsilon+1)\hat{\pi}) \mid \upsilon : N \}$

$\wedge \bigwedge\{ \bigwedge\{ \sim(1 \le \nu_0 \le \#\hat{\rho}[\![I_0]\!] \wedge 1 \le \nu_1 \le \#\hat{\rho}[\![I_1]\!]) \rightarrow true,$

$\hat{\rho}_0[\![I_0]\!]\downarrow\nu_0 : V \vee \sim(\hat{\rho}[\![I_0]\!]\downarrow\nu_0 = \hat{\rho}[\![I_1]\!]\downarrow\nu_1) \rightarrow true,$

$(\sim \bigvee\{ known(level(\hat{\rho}[\![I_0]\!]\downarrow\nu_0\downarrow 2 - 1)\hat{\pi})\langle \varepsilon, \hat{\rho}[\![I_0]\!]\downarrow\nu_0 \rangle \hat{\pi} \mid \varepsilon : E \}$

$\wedge I_0 = I_1 \wedge \hat{\rho}[\![I_0]\!]\downarrow\nu_0 \dagger 1 = \hat{\rho}[\![I_1]\!]\downarrow\nu_1 \dagger 1)$

$\mid \nu_0 : N \wedge \nu_1 : N \} \mid I_0 : Ide \wedge I_1 : Ide \}$

$\wedge \mathfrak{Q} q_0\hat{\rho} = \mathfrak{Q} q_0(ravel\hat{\rho}\hat{\rho}).$

The closing clause of this predicate will be required in 3.3.4 and 3.3.5. That relating $\hat{\rho}[\![I_0]\!]\downarrow\nu_0$ and $\hat{\rho}[\![I_1]\!]\downarrow\nu_1$ is an amalgam of two others: 3.3.3 will implicitly use the fact that when $ravel\hat{\rho}\hat{\rho}[\![I_0]\!]\downarrow\nu_0\downarrow 1 : L$ we also have $known(level(\hat{\rho}[\![I_0]\!]\downarrow\nu_0\downarrow 2 - 1)\hat{\pi})\langle \varepsilon, \hat{\rho}[\![I_0]\!]\downarrow\nu_0 \rangle \hat{\pi} = false$ for all ε, whereas 3.3.7 will require I_0 and I_1 to coincide when $\hat{\rho}[\![I_0]\!]\downarrow\nu_0 : L$ and $\hat{\rho}[\![I_0]\!]\downarrow\nu_0 = \hat{\rho}[\![I_1]\!]\downarrow\nu_1$. The first of these conditions holds only because in 3.1.4 we prohibited abstractions and some declarations by reference; the second, however, is inevitably valid but would be irrelevant were we to adopt the alternative forms for $\mathscr{T}[\![I=E]\!]$ and $\mathscr{T}[\![I_1,\ldots,I_n=E]\!]$ mentioned in 1.3.4.

Since the tracing algorithms of 3.2.1 do not take account of the output we restrain it in p_0 by means of

$\omega_0 = \lambda\hat{\omega}\hat{\pi}.\, \hat{\omega} : B \times B \rightarrow b\hat{\omega}, (\hat{\omega} : L \wedge \hat{\omega} : E) \vee \hat{\omega} : J \times J \vee \hat{\omega} : F \times F \vee \hat{\omega} : J \times J.$

As usual we build up a sequence of predicates ω_n each of which induces some a_{n+1} on $A^\circ \times A^\circ$ together with

$p_{n+1}=\lambda\hat{\pi}.p_0\hat{\pi}\wedge\bigwedge\{w_n\hat{\omega}\hat{\pi}\,|\,kent1\hat{\omega}\hat{\pi}\}.$

Before defining them we clarify the description of c_{n+1} above by setting

$fit=\lambda\hat{\pi}_0\hat{\pi}_1.p_1\hat{\pi}_1$

$\wedge\bigwedge\{\hat{\varepsilon}:L\times L\vee\hat{\varepsilon}:V\times V\vee\sim gyven\hat{\varepsilon}\theta_1\,|\,\hat{\varepsilon}:E\times E\}$

$\wedge(\mathfrak{U}q_0\times\mathfrak{U}q_0)\hat{\rho}_0=(\mathfrak{U}q_0\times\mathfrak{U}q_0)\hat{\rho}_1\wedge(\mathfrak{V}q_0\times\mathfrak{V}q_0)\hat{\upsilon}_0=(\mathfrak{V}q_0\times\mathfrak{V}q_0)\hat{\upsilon}_1$

$\wedge restore\sigma_1\circ restore\sigma_0=restore\sigma_1$

$\wedge restore\delta_1\circ restore\delta_0=restore\delta_1$

$\wedge\bigwedge\{\bigwedge\{known(level\nu\hat{\pi}_0)\hat{\alpha}\hat{\pi}_1\vee\sim known(level\nu\hat{\pi}_0)\hat{\alpha}\hat{\pi}_0\vee\nu>\#\rho_0[\![rec]\!]$

$\,|\,\hat{\alpha}:L\times L\}\,|\,\nu:N\}.$

The only noteworthy feature of this function is its insistence that on executing an expression no fresh locations become associated with the environment levels of outer blocks to which control may be restored.

As intimated by 3.1.4 the exits of an expression permit the return of only those members of E which could have been present on entry to a specified block. The substitute for *set* therefore takes an additional parameter, the depth of nesting of the block, and insists that the top element of the stack conforms to it. More formally, we introduce

$pat=\lambda\hat{\pi}_0\nu\hat{\pi}_1.((\upsilon_0\downarrow1:L\wedge known(level\nu\hat{\pi}_1)\langle\upsilon_0\downarrow1,\delta_0\downarrow1\rangle\hat{\pi}_0)$

$\vee(\delta_0\downarrow1:L\rightarrow\sim area(\delta_0\downarrow1)\delta_1,true))$

$\wedge found(level\nu\hat{\pi}_0)(access\langle\upsilon_0\downarrow1,\delta_0\downarrow1\rangle\hat{\pi}_0)\hat{\pi}_0$

$\wedge fit(\langle\rho_0,\upsilon_0\dagger1,\sigma_0\rangle,\langle\rho_0,\delta_0\dagger1,\delta_0\rangle)\hat{\pi}_1.$

Notice that when $\nu=\#\rho_0[\![rec]\!]+1$ and $\hat{\varepsilon}=access\langle\upsilon_0\downarrow1,\delta_0\downarrow1\rangle\hat{\pi}_0$ we require only that $found\hat{\pi}_0\hat{\varepsilon}\hat{\pi}_0=true$ rather than that $found\hat{\pi}_1\hat{\varepsilon}\hat{\pi}_0=true$ in order to provide for such programs as nil aug false. Corresponding to c_{n+1} we have

$j_{n+1}=\lambda\hat{\zeta}\nu\hat{\pi}_0.\bigwedge\{a_{n+1}\langle\mathfrak{Z}q_n\zeta\rho\upsilon\sigma,\mathfrak{Z}q_n\zeta\rho\upsilon\sigma\rangle\,|\,p\hat{\pi}\wedge pat\hat{\pi}\nu\hat{\pi}_0\}$ and

$j=\lambda\hat{\zeta}\nu\hat{\pi}_0.\bigwedge j_{n+1}\hat{\zeta}\nu\hat{\pi}_0.$

Plainly $j_{n+1}=\lambda\hat{\zeta}\nu\hat{\pi}.j_{n+1}\hat{\zeta}\nu((\mathfrak{V}q_0\times\mathfrak{V}q_0)\hat{\pi})$ and in fact we even

know that $w_{n+1}=\lambda\hat{\omega}\hat{\pi}.w_{n+1}\hat{\omega}((\mathbf{P}q_0\times\mathbf{P}q_0)\hat{\pi})$, where

$w_{n+1}=\lambda\hat{\omega}\hat{\pi}.w_0\hat{\omega}\hat{\pi}$

$\wedge(\hat{\omega}:J\times J\to\bigwedge\{c_n(\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1)(\langle\acute{\rho}_0,\acute{\upsilon}_0,\acute{\sigma}\rangle,\grave{\omega}\dagger 1\S(\grave{\partial})$

$\wedge\acute{\omega}\downarrow 1=\lambda\rho\upsilon\sigma.(\acute{\omega}\downarrow 1)(revert\acute{\rho}_0\rho)(pop\acute{\upsilon}_0\upsilon)(restore\acute{\sigma}_0\sigma)$

$|\acute{\pi}_0=level(\acute{\omega}\downarrow 2)\hat{\pi}\},$

$\hat{\omega}:F\times F\to\bigwedge\{c_n(\acute{\omega}\downarrow 1)\hat{\zeta}\circ revert\acute{\rho}_0,(\grave{\omega}\downarrow 1)(\grave{\zeta}\circ revert\grave{\rho}_0))\hat{\pi}_1$

$|fit\hat{\pi}_1\hat{\pi}_1\wedge tidy(\acute{\omega}\downarrow 2)(level\vee\acute{\pi}_0\downarrow 1)\wedge j_n\hat{\zeta}\vee\hat{\pi}_0$

$\wedge\acute{\pi}_1=\langle divert\acute{\rho}_0(\acute{\omega}\downarrow 2),\langle\acute{\upsilon}_1\downarrow 1\rangle\S\acute{\upsilon}_0,\acute{\sigma}_0\rangle$

$\wedge\grave{\pi}_1=\langle divert\grave{\rho}_0(\grave{\omega}\downarrow 2),\langle\grave{\upsilon}_1\downarrow 1\rangle\S\grave{\upsilon}_0,\grave{\sigma}_0\rangle$

$\wedge\bigwedge\{apt\psi(\acute{\rho}_1,\grave{\omega}\downarrow 2)$

$\vee\sim apt\psi(divert\acute{\rho}(\acute{\omega}\downarrow 2),\grave{\omega}\downarrow 2)|\psi\}\},$

$\hat{\omega}:J\times J\to\bigwedge\{j_n(\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1)0(\langle\acute{\rho}_0,\acute{\upsilon}_0,\acute{\sigma}\rangle,\grave{\omega}\dagger 1\S(\grave{\partial})\rangle$

$\wedge(\acute{\omega}\downarrow 1)\acute{\rho}_1=remit(\acute{\omega}\downarrow 1)\acute{\rho}_1[\acute{\pi}_0//rec]$

$|\acute{\pi}_0=level(\acute{\omega}\downarrow 2)\hat{\pi}\wedge\mathbf{P}q_0\acute{\rho}_0=\mathbf{P}q_0(revert\acute{\rho}_0\acute{\rho}_1)\},$

$true).$

Having provided the necessary recurrence relations we can write $w=\lambda\hat{\omega}\hat{\pi}.\bigwedge w_n\hat{\omega}\hat{\pi}$, $p=\lambda\hat{\pi}.\bigwedge p_{n+1}\hat{\pi}$ and $a=\lambda\hat{\rho}.\bigwedge a_{n+1}\hat{\rho}$. Before relating these to the syntax of Mal we supplant the function *apt* of 1.4.6 with a version appropriate to stack semantics, which is

$apt=\lambda\psi\hat{\rho}.neat\hat{\rho}$

$\wedge\bigwedge\{\#\grave{\rho}[\![I]\!]=0\vee(\#\grave{\rho}[\![I]\!]>0\wedge\#\psi[\![I]\!]=0)\to true,$

$ravel\acute{\rho}\acute{\rho}[\![I]\!]\downarrow 1\downarrow 1:L\to\grave{\rho}[\![I]\!]\downarrow 1:L\wedge(\psi[\![I]\!]\downarrow 1=false),$

$\grave{\rho}[\![I]\!]\downarrow 1:V\vee(\grave{\rho}[\![I]\!]\downarrow 1:L\wedge(\psi[\![I]\!]\downarrow 1=true))|I:Ide\}.$

The predicates defined on program texts are very similar to those of 2.4.5. In the present case, however, we do not wish to assert that every program is equivalent to its transform if *opt* is chosen suitably, for evaluating the program of 3.1.5 using *new* stack semantics will have a disastrous effect, whereas evaluating it in *new* store semantics will not. Thus we demand equivalence only when the program is subject to the constraints of 3.1.4 and set

$E=\lambda E.\bigwedge\{c(\mathcal{E}[\![E]\!]\zeta',\mathcal{E}[\![\bullet[\![E]\!]\psi]\!]\zeta)\,\hat{\pi}\vee\sim e[\![E]\!]\nu\rho' \mid apt\psi\hat{\rho}\wedge rent[\![E]\!]\psi\wedge j\hat{\zeta}\nu\hat{\pi}\};$

$L=\lambda E.\bigwedge\{c(\mathcal{L}[\![E]\!]\zeta',\mathcal{L}[\![\bullet[\![E]\!]\psi]\!]\zeta)\,\hat{\pi}\vee\sim e[\![E]\!]\nu\rho' \mid apt\psi\hat{\rho}\wedge rent[\![E]\!]\psi\wedge j\hat{\zeta}\nu\hat{\pi}\};$

$R=\lambda E.\bigwedge\{c(\mathcal{R}[\![E]\!]\zeta',\mathcal{R}[\![\bullet[\![E]\!]\psi]\!]\zeta)\,\hat{\pi}\vee\sim e[\![E]\!]\nu\rho' \mid apt\psi\hat{\rho}\wedge rent[\![E]\!]\psi\wedge j\hat{\zeta}\nu\hat{\pi}\};$

$G=\lambda E.\bigwedge\{\ c(\mathcal{G}[\![E]\!]\zeta',\mathcal{G}[\![g[\![E]\!]\psi]\!]\zeta)\,\hat{\pi}\vee\sim g[\![E]\!]\nu\rho' \mid apt\psi\hat{\rho}\wedge torn[\![E]\!]\psi\wedge j\hat{\zeta}\nu\hat{\pi}\}$

$\wedge(\mathcal{J}[\![E]\!]\,\S\mathcal{K}[\![E]\!]=())$

$\vee\bigwedge\{\bigwedge\{w(\mathcal{P}[\![E]\!]\zeta'\rho'\upsilon'\sigma'\,\S\mathcal{Q}[\![E]\!]\zeta'\rho'\upsilon'\sigma'+\nu_1,$

$swap(\mathcal{J}[\![E]\!]\,\S\mathcal{K}[\![E]\!])(\mathcal{J}[\![g[\![E]\!]\psi]\!]\,\S\mathcal{K}[\![g[\![E]\!]\psi]\!])$

$(\mathcal{P}[\![g[\![E]\!]\psi]\!]\zeta\rho\upsilon\,\S\mathcal{Q}[\![g[\![E]\!]\psi]\!]\zeta\rho\upsilon)+\nu_1)\,\hat{\pi}$

$\mid 1\le\nu_1\le\#\mathcal{J}[\![E]\!]\,\S\mathcal{K}[\![E]\!]\}$

$\vee\sim g[\![E]\!]\nu\rho' \mid apt\psi\hat{\rho}\wedge torn[\![E]\!]\psi\wedge j\hat{\zeta}\nu\hat{\pi}\}.$

The predicates on declarations also bear a marked resemblance to those in 2.4.5, being

$D=\lambda\Delta.\bigwedge\{c(\mathcal{D}[\![\Delta]\!]\zeta',\mathcal{D}[\![d[\![\Delta]\!]\psi]\!]\zeta)\,\hat{\pi}_0\vee\sim d[\![\Delta]\!]0\rho_0 \mid\bigwedge\{c\hat{\zeta}\hat{\pi}_1 \mid spun[\![\Delta]\!]0\psi\hat{\pi}_0\hat{\pi}_1\}\};$

$T=\lambda\Delta.\bigwedge\{c(\mathcal{T}[\![\Delta]\!]\zeta'\,\mathcal{T}[\![t[\![\Delta]\!]\psi]\!]\zeta)\,\hat{\pi}_0\vee\sim d[\![\Delta]\!]0\rho_0 \mid\bigwedge\{c\hat{\zeta}\hat{\pi}_1 \mid spun[\![\Delta]\!]1\psi\hat{\pi}_0\hat{\pi}_1\}\}.$

Here we describe the state following the execution of a declaration using an elaborate set of properties the purpose of which will be clarified in 3.3.7. For the present we merely include them in a test function *spun*, which will be given below. This function is rendered more complex than *sewn* by the need to deal with transformations which map recursive declarations by incidence into recursive declarations by reference which are formulated in terms of environments appropriate to store semantics (rather than ones appropriate to stack semantics). More specifically, we let

$$
\begin{aligned}
spun = \lambda\Delta\nu\psi\hat{\pi}_0\hat{\pi}_1.(\lambda\hat{\pi}_0\hat{\pi}_1.\wedge\{I:\mathscr{I}[\![\Delta]\!] \cup \mathscr{H}[\![\Delta]\!] \rightarrow & revert\hat{\rho}_0\hat{\rho}_1[\![I]\!]\dagger\nu = \hat{\rho}_0[\![I]\!]\dagger\nu \\
& \wedge(I:\mathscr{I}[\![\Delta]\!] \rightarrow ravel\hat{\rho}_1\hat{\rho}_1[\![I]\!]\downarrow 1\downarrow 1 : L \\
& \qquad ravel\hat{\rho}_1\hat{\rho}_1[\![I]\!]\downarrow 1\downarrow 1 : V) \\
& \wedge\hat{\rho}_1[\![I]\!]\downarrow 1\downarrow 2 = \#\hat{\rho}_0[\![rec]\!]+1, \\
revert\hat{\rho}_0\hat{\rho}_1[\![I]\!] = \hat{\rho}_0[\![I]\!] & \wedge revert\hat{\rho}_0\hat{\rho}_1[\![I]\!] = \hat{\rho}_0[\![I]\!] \\
\wedge(\#\hat{\rho}_0[\![I]\!] > 0 \rightarrow & \hat{\rho}_1[\![I]\!]\downarrow 1 = \hat{\rho}_0[\![I]\!]\downarrow 1 \wedge \hat{\rho}_1[\![I]\!]\downarrow 1 = \hat{\rho}_0[\![I]\!]\downarrow 1, \\
& true)\,|\,I:Ide\} \\
\wedge\wedge\{\nu = 0 \rightarrow & revert\hat{\rho}_0\hat{\rho}_1[\![I]\!] = \hat{\rho}_0[\![I]\!], \\
I:\mathscr{I}[\![\Delta]\!] \rightarrow & revert\hat{\rho}_0\hat{\rho}_1[\![I]\!]\dagger 1 = \hat{\rho}_0[\![I]\!]\dagger 1 \\
& \wedge ravel\hat{\rho}_0\hat{\rho}_0[\![I]\!]\downarrow 1 : L \\
& \wedge\hat{\rho}_1[\![I]\!]\downarrow 1\downarrow 2 = \hat{\rho}_0[\![I]\!]\downarrow 1\downarrow 2, \\
I:\mathscr{H}[\![\Delta]\!] \rightarrow & \hat{\rho}_1[\![I]\!]\dagger lead1(\hat{\rho}_1[\![I]\!]) = \hat{\rho}_0[\![I]\!]\dagger lead1(\hat{\rho}_0[\![I]\!]) \\
& \wedge ravel\hat{\rho}_0\hat{\rho}_0[\![I]\!]\downarrow 1 : V \wedge \hat{\rho}_1[\![I]\!]\downarrow 1 : L \wedge \hat{\rho}_0[\![I]\!]\downarrow 1 : L \\
& \wedge\hat{\rho}_1[\![I]\!]\downarrow 1\downarrow 2 = \hat{\rho}_0[\![I]\!]\downarrow 1\downarrow 2, \\
& true\,|\,I:Ide\} \\
\wedge\hat{\rho}_1[\![res]\!] = \hat{\rho}_0[\![res]\!] & \wedge\hat{\rho}_1[\![rec]\!] = \hat{\rho}_0[\![rec]\!] \\
\wedge\rho_1[\![res]\!] = \hat{\rho}_0[\![res]\!] & \wedge\hat{\rho}_1[\![rec]\!] = \hat{\rho}_0[\![rec]\!] \\
\wedge restore\hat{\sigma}_0 \circ restore\hat{\sigma}_1 = & restore\hat{\sigma}_0 \\
\wedge restore\hat{\vartheta}_0 \circ restore\hat{\vartheta}_1 = & restore\hat{\vartheta}_0 \\
\wedge\wedge\{\wedge\{known(level\nu_1\hat{\pi}_0)\hat{\alpha}\hat{\pi}_0 \vee \sim known(level\nu_1\hat{\pi}_0)\hat{\alpha}\hat{\pi}_1 \\
& \vee\nu_1 > \#\hat{\rho}_0[\![rec]\!]\,|\,\hat{\alpha}:L\times L\}\,|\,\nu_1:N\} \\
\wedge apt\psi\hat{\rho}_0\wedge\hat{\sigma}_1 = & \hat{\sigma}_0\wedge fit\hat{\pi}_1\hat{\pi}_1\wedge fit\hat{\pi}_0\hat{\pi}_0 \\
\wedge(\lambda\psi'.apt\psi'\hat{\rho}_1\wedge & torn[\![\Delta]\!]\psi') \\
(\nu = 0 \rightarrow & \psi[false^*/\mathscr{I}[\![\Delta]\!]][opts(\mathscr{H}[\![\Delta]\!])\psi/\mathscr{H}[\![\Delta]\!]],\psi))
\end{aligned}
$$

$$((\mathfrak{P}q_0\times\mathfrak{P}q_0)\hat{\pi}_0)((\mathfrak{P}q_0\times\mathfrak{P}q_0)\hat{\pi}_1).$$

Before proceeding to the applications of these predicates we mention the result which asserts that they are indeed what we want. Like 2.4.5, this result could be formulated in terms of the general theory of 2.2.8, but the notation needed would be at least as complex as that used above.

3.2.5. Proposition.

Suppose that $a_1(\bot,\bot)=true$ and that for any $n\geq 0$ if w_n is inclusive and every $\hat{\omega}$ and $\hat{\pi}$ having $kent1\hat{\omega}\hat{\pi}\wedge p_1\hat{\pi}=true$ satisfy $w_n((q_n\times q_n)\hat{\omega})\hat{\pi}\supset w((q_n\times q_n)\hat{\omega})\hat{\pi}$, $w_{n+1}\hat{\omega}\hat{\pi}\supset w_n\hat{\omega}\hat{\pi}$ and $w_n\hat{\omega}\hat{\pi}\supset w_{n+1}((q_n\times q_n)\hat{\omega})\hat{\pi}$ as well then a_{n+1} is inclusive and for all $\hat{\delta}$ $a\hat{\delta}$ is proper, $a_{n+1}((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})=a((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})$, $a_{n+2}\hat{\delta}\supset a_{n+1}\hat{\delta}$ and $a_{n+1}\hat{\delta}\supset a_{n+2}((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})$. Then w, p, c, j and a, defined as in 3.2.4, are the unique inclusive predicates such that $a(\bot,\bot)=true$, for any $n\geq 0$ if $w\hat{\omega}\hat{\pi}\supset w_n\hat{\omega}\hat{\pi}$ and $w_n((q_n\times q_n)\hat{\omega})\hat{\pi}\supset w((q_n\times q_n)\hat{\omega})\hat{\pi}$ whenever $kent1\hat{\omega}\hat{\pi}\wedge p_1\hat{\pi}=true$ then $a\hat{\delta}\supset a_{n+1}\hat{\delta}$ and $a_{n+1}((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})\supset a((\mathfrak{A}q_n\times\mathfrak{A}q_n)\hat{\delta})$, and

(i) $w=\lambda\hat{\omega}\hat{\pi}.w_0\hat{\omega}\hat{\pi}$

$\wedge(\hat{\omega}:J\times J\to\bigwedge\{c(\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1)(\langle\acute{\rho}_0,\acute{\upsilon}_0,\acute{\sigma}\rangle,\grave{\omega}\dagger 1\S(\grave{\sigma}))$

$\wedge\acute{\omega}\downarrow 1=\lambda\rho\upsilon\sigma.(\acute{\omega}\downarrow 1)(revert\acute{\rho}_0\rho)(pop\acute{\upsilon}_0\upsilon)(restore\acute{\sigma}_0\sigma)$

$|\hat{\pi}_0=level(\acute{\omega}\downarrow 2)\hat{\pi}\},$

$\hat{\omega}:F\times F\to\bigwedge\{c(\acute{\omega}\downarrow 1)\hat{\zeta}\circ revert\acute{\rho}_0,(\grave{\omega}\downarrow 1)(\grave{\zeta}\circ revert\grave{\rho}_0))\hat{\pi}_1$

$|fit\hat{\pi}_1\hat{\pi}_1\wedge tidy(\acute{\omega}\downarrow 2)(level\upsilon\hat{\pi}_0)\wedge j\hat{\zeta}_0\upsilon_0\hat{\pi}_0$

$\wedge\hat{\pi}_1=\langle divert\acute{\rho}_0(\acute{\omega}\downarrow 2),\langle\acute{\upsilon}_1\downarrow 1\rangle\S\acute{\upsilon}_0,\acute{\sigma}_0\rangle$

$\wedge\hat{\pi}_1=\langle divert\grave{\rho}_0(\grave{\omega}\downarrow 2),\langle\grave{\upsilon}_1\downarrow 1\rangle\S\grave{\upsilon}_0,\grave{\sigma}_0\rangle$

$\wedge\bigwedge\{apt\psi(\acute{\rho}_1,\grave{\omega}\downarrow 2)$

$\vee\sim apt\psi(divert\acute{\rho}(\acute{\omega}\downarrow 2),\grave{\omega}\downarrow 2)\,|\psi\}\},$

$\hat{\omega}:J\times J\to\bigwedge\{j(\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1)\,0(\langle\acute{\rho}_0,\acute{\upsilon}_0,\acute{\sigma}\rangle,\grave{\omega}\dagger 1\S(\grave{\sigma}))$

$\wedge(\acute{\omega}\downarrow 1)\acute{\rho}_1=remit(\acute{\omega}\downarrow 1)\acute{\rho}_1[\hat{\pi}_0//rec]$

$|\hat{\pi}_0=level(\acute{\omega}\downarrow 2)\hat{\pi}\wedge\mathfrak{A}q_0\acute{\rho}_0=\mathfrak{A}q_0(revert\acute{\rho}_0\acute{\rho}_1)\},$

$true);$

(ii) $p=\lambda\hat{\pi}.p_0\hat{\pi}\wedge\bigwedge\{w\hat{\omega}\hat{\pi}\,|kent1\hat{\omega}\hat{\pi}\};$

(iii) $c=\lambda\hat{\zeta}\hat{\pi}.\bigwedge\{a(\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0)\,|p\hat{\pi}_0\wedge fit\hat{\pi}_0\hat{\pi}\};$

(iv) $j=\lambda\hat{\zeta}\upsilon\hat{\pi}.\bigwedge\{a(\acute{\zeta}\acute{\rho}_0\acute{\upsilon}_0\acute{\sigma}_0,\grave{\zeta}\grave{\rho}_0\grave{\upsilon}_0\grave{\sigma}_0)\,|p\hat{\pi}_0\wedge pat\hat{\pi}_0\upsilon\hat{\pi}\}$.

⊲The proof is very similar, both in outline and in detail, to that of 2.4.8. Thus first we show as in 2.4.6 that $p_{n+2}\hat{\pi}\supset p_{n+1}\hat{\pi}$

and $p_{n+1}\hat{\pi}\supset p_{n+2}((\mathbb{P}q_n\times\mathbb{P}q_n)\hat{\pi})$ for all $n\geq 0$ and $\hat{\pi}$, and then we follow 2.4.7 by verifying that p and the other predicates are inclusive. Similarly the demonstration that the predicates are unique is based on 2.2.6. None of the techniques involved are novel so we shall not discuss them at all.⊁

Henceforth we shall assume not merely that $a\langle\perp,\perp\rangle=true$ but that $a\langle\top,\top\rangle=true$ as well. Lemmata like 3.3.1 will also require the presumption that L is infinite in order to be sure that the transform of a program does not run out of store unless the program itself does. As our current version of $\omega\hat{\omega}\hat{\pi}$ depends on $\mathbb{E}q_0\acute{o}$ and $\mathbb{P}q_0\acute{o}$ whereas that in 2.6.1 does not, we shall analyse the equivalence of appendix 3 with store semantics only after giving three further results.

3.2.6. Lemma.

Suppose that $\hat{\pi}_0$ satisfies $p_0\hat{\pi}_0=true$, and define $\hat{\pi}_1$ to be $(\mathbb{P}q_0\times\mathbb{P}q_0)\hat{\pi}_0$. Then for all $\hat{\omega}$ $kent1\hat{\omega}\hat{\pi}_1=true$ if and only if $kent3\hat{\omega}\hat{\pi}_1=true$; in addition, if $\hat{\omega}$ satisfies $kent1\hat{\omega}\hat{\pi}_0=true$ then $found\hat{\pi}_0\hat{\omega}\hat{\pi}_0=true$.

⊀From the definition of *seen* in 3.2.1 it is plain that when $kent3\hat{\omega}\hat{\pi}_1=true$ we have $kent1\hat{\omega}\hat{\pi}_1=true$ also. Moreover we know that $p_0\hat{\pi}_1=true$, since for all ν and $\hat{\omega}_1$ $kent\nu\hat{\omega}_1\hat{\pi}_1=true$ if and only if $kent\nu\hat{\omega}_0\hat{\pi}_0=true$ for some $\hat{\omega}_0$ having $\hat{\omega}_1=(q_0\times q_0)\hat{\omega}_0$.

Assume that for some ν and all $\hat{\omega}_0$ and $\hat{\omega}_1$ if $seen1\nu\hat{\omega}_0\hat{\omega}_1\hat{\pi}_1$ and $kent3\hat{\omega}_1\hat{\pi}_1$ are *true* then $kent3\hat{\omega}_0\hat{\pi}_1$ is *true*, and take any $\hat{\omega}_0$ and $\hat{\omega}_1$ having $seen1(\nu+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_1\wedge kent3\hat{\omega}_1\hat{\pi}_1=true$. If $\acute{\omega}_1:L$ or $\grave{\omega}_1:L$ $seen1\nu\hat{\omega}_0\hat{\omega}_2\hat{\pi}_1\wedge kent3\hat{\omega}_2\hat{\pi}_1=true$, where $\hat{\omega}_2=access\hat{\omega}_1\hat{\pi}_1$. If $\hat{\omega}_1:L^*\times L^*$ $seen1\nu\hat{\omega}_0\hat{\omega}_2\hat{\pi}_1=true$ for some $\hat{\omega}_2$ such that $gyven\hat{\omega}_2\hat{\omega}_1=true$. If $\hat{\omega}_1:J\times J$ $seen1\nu\hat{\omega}_0\hat{\omega}_2\hat{\pi}_1=true$ for some $\hat{\omega}_2$ having $hoten\hat{\omega}_2\langle revert(\acute{\omega}_1\downarrow 2)\acute{o}_1,\grave{\omega}_1\downarrow 2\rangle=true$ or $gyven\hat{\omega}_2\langle pop(\acute{\omega}_1\downarrow 3)\acute{o}_1,\grave{\omega}_1\downarrow 3\rangle=true$; as $found\hat{\pi}_1\hat{\omega}_1\hat{\pi}_1=true$ and $(\mathbb{P}q_0\times\mathbb{P}q_0)\hat{\pi}_1=\hat{\pi}_1$, in fact we have

$\grave{\omega}_1\downarrow 2 = revert(\grave{\omega}_1\downarrow 2)\grave{\rho}_1$ and $\grave{\omega}_1\downarrow 3 = pop(\grave{\omega}_1\downarrow 3)\grave{\upsilon}_1$, giving $hoten\hat{\omega}_2\hat{\rho}_1 = true$ or $gyven\hat{\omega}_2\hat{\upsilon}_1 = true$. If $\hat{\omega}_1 : F\times F$ $seen1\nu\hat{\omega}_0\hat{\omega}_2\hat{\pi}_1 = true$ for some $\hat{\omega}_2$ having $hoten\hat{\omega}_2(divert\acute{\rho}_1(\vec{\omega}_1\downarrow 2), \grave{\omega}_1\downarrow 2) = true$; because $tidy(\vec{\omega}_1\downarrow 2)\acute{\rho}_1 = true$, $\grave{\rho}_1 = ravel\acute{\rho}_1\grave{\rho}_1$ and $divert\grave{\rho}_1(\grave{\omega}_1\downarrow 2) = ravel(divert\acute{\rho}_1(\vec{\omega}_1\downarrow 2))(divert\grave{\rho}_1(\grave{\omega}_1\downarrow 2))$ we actually know that $hoten\hat{\omega}_2\hat{\rho}_1 = true$. Finally, if $\hat{\omega}_1 : J\times J$ arises from val we repeat the argument used for label entry points. Hence under all circumstances there is some $\hat{\omega}_2$ such that $seen1\nu\hat{\omega}_0\hat{\omega}_2\hat{\pi}_1 \wedge kent3\omega_2\hat{\pi}_1 = true$ and from the induction hypothesis we can infer that $kent3\hat{\omega}_0\hat{\pi}_1 = true$.

As our assumption is valid when $\nu<1$ for all ν, $\hat{\omega}_0$ and $\hat{\omega}_1$ having $seen1\nu\hat{\omega}_0\hat{\omega}_1\hat{\pi}_1 \wedge kent3\hat{\omega}_1\hat{\pi}_1 = true$ $kent3\hat{\omega}_0\hat{\pi}_1 = true$. Moreover $kent3\hat{\omega}_1\hat{\pi}_1 = true$ when $hoten\hat{\omega}_1\hat{\rho}_1 = true$ or $gyven\hat{\omega}_1\hat{\upsilon}_1 = true$, so $kent3\hat{\omega}_0\hat{\pi}_1 = true$ whenever $kent1\hat{\omega}_0\hat{\pi}_1 = true$.

Take some $\hat{\omega}_3$ having $kent1\hat{\omega}_3\hat{\pi}_0 = true$; then, writing $\hat{\omega}_4 = (q_0\times q_0)\hat{\omega}_3$, $kent1\hat{\omega}_4\hat{\pi}_0 = true$ and $kent3\hat{\omega}_4\hat{\pi}_0 = true$. Since $p_0\hat{\pi}_1 = true$, $found\hat{\pi}_1\hat{\omega}_4\hat{\pi}_1 = true$ and from the definitions of 3.2.1 it is plain that $found\hat{\pi}_0\hat{\omega}_3\hat{\pi}_0 = true$.⊳

3.2.7. Proposition.

Let $\hat{\pi}_0$, $\hat{\pi}_1$, $\hat{\pi}_2$ and $\hat{\pi}_3$ be pairs such that $p_1\hat{\pi}_0 = true$, $p\hat{\pi}_2 = true$ and $\hat{\pi}_{n+1} = (\mathbb{P}q_0\times\mathbb{P}q_0)\hat{\pi}_n$ when n is 0 or 2. Set $\nu_0 = \#\acute{\rho}_0[\![rec]\!]$, and suppose that $level\nu\hat{\pi}_1 = level\nu\hat{\pi}_3$ and that $known(level\nu\hat{\pi}_0)\hat{\alpha}\hat{\pi}_0 \supset known(level\nu\hat{\pi}_0)\hat{\alpha}\hat{\pi}_2$ for all $\nu\leq\nu_0$ and all $\hat{\alpha}$. Assume also that $restore\acute{\sigma}_2\circ restore\acute{\sigma}_0 = restore\acute{\sigma}_2$, that $restore\grave{\sigma}_2\circ restore\grave{\sigma}_0 = restore\grave{\sigma}_2$ and that for all ν_8, ν_9 and I having $\acute{\rho}_0[\![I]\!]\downarrow\nu_8+1 = \acute{\rho}_2[\![I]\!]\downarrow\nu_9+1$ $ravel\acute{\rho}_0\acute{\rho}_0[\![I]\!]\downarrow\nu_8\downarrow 1 : L$ only if $ravel\acute{\rho}_2\acute{\rho}_2[\![I]\!]\downarrow\nu_9\downarrow 1 : L$. Any pair $\hat{\omega}$ having $kent1\hat{\omega}\hat{\pi}_0 \wedge kent1\hat{\omega}\hat{\pi}_2 = true$ satisfies $w\hat{\omega}\hat{\pi}_1 = true$ also; in particular, if $kent1\hat{\omega}\hat{\pi}_2 = true$ whenever $kent1\hat{\omega}\hat{\pi}_0 = true$ $p\hat{\pi}_0 = true$.

⊲As $p_0\hat{\pi}_0 = true$, by 3.2.6 any pair $\hat{\omega}$ having $kent1\hat{\omega}\hat{\pi}_0 = true$

satisfies $found\Phi_0\hat{\omega}\Phi_0=true$. Take any such $\hat{\omega}$ and presume also that $kent1\hat{\omega}\Phi_2=true$; because $p\Phi_2=true$ we can stipulate that $found\Phi_2\hat{\omega}\Phi_2=true$ and that $w\hat{\omega}\Phi_2=true$ when showing that $w\hat{\omega}\Phi_0=true$. The desired result is obvious unless $\hat{\omega}:J\times J$, $\hat{\omega}:F\times F$ or $\hat{\omega}:J\times J$, and since the third of these exceptional cases involves the same considerations as the first it will not be discussed.

If $\hat{\omega}:J\times J$ the fact that $found\Phi_0\hat{\omega}\Phi_0=true$ assures us that $\acute{\omega}\downarrow 2\leq\nu_0$; hence writing $\Phi_4=level(\acute{\omega}\downarrow 2)\Phi_0$ we have $\mathbb{P}q_0\Phi=level(\acute{\omega}\downarrow 2)\Phi_1=level(\acute{\omega}\downarrow 2)\Phi_3=\mathbb{P}q_0(level(\acute{\omega}\downarrow 2)\Phi_2)$. From the definition of fit in 3.2.4 and the knowledge that $w\hat{\omega}\Phi_2=true$ it now follows that $c\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_2\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_2\rangle\rangle=true$ and that $\acute{\omega}\downarrow 1=\lambda\rho\upsilon\sigma.(\acute{\omega}\downarrow 1)(revert\acute{\rho}\rho)(pop\acute{\upsilon}\upsilon)(restore\acute{\sigma}\sigma)$. Remembering that $found\Phi_2\hat{\omega}\Phi_2=true$ we deduce that $\langle revert\acute{\rho}\grave{\rho}_1,pop\acute{\upsilon}\grave{\upsilon}_1\rangle=q_0\grave{\omega}\dagger 1=\langle revert\acute{\rho}\grave{\rho}_3,pop\acute{\upsilon}\grave{\upsilon}_3\rangle$; applying an analogue of 2.4.3 to our present reflexive projections we therefore see that by 3.2.2 for all ν and $\hat{\alpha}$ $known(level\nu\Phi)\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle=known(level\nu\Phi)\hat{\alpha}\Phi_0$ and $known(level\nu\Phi)\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_2\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_2\rangle\rangle=known(level\nu\Phi)\hat{\alpha}\Phi_2$. Together with our initial assumptions these equalities imply that $fit\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_2\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_2\rangle\rangle=true$; moreover by 3.2.6 for all $\hat{\omega}$ we have

$$kent1\hat{\omega}\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle\supset kent3((q_0\times q_0)\hat{\omega})\langle\mathbb{P}q_0\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_0\rangle,q_0\grave{\omega}\dagger 1\S\langle\grave{\sigma}_1\rangle\rangle$$
$$\supset kent3((q_0\times q_0)\hat{\omega})((\mathbb{P}q_0\times\mathbb{P}q_0)\Phi_0),$$

which entails $p_1\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle=true$ as $p_1\Phi_0=true$. Given any $\hat{\zeta}$, Φ_5 and Φ_6 such that $c\hat{\zeta}\Phi_6=true$ and $p_1\Phi_5\wedge fit\Phi_5\Phi_6=true$ we have $c\hat{\zeta}\Phi_5=true$, for every Φ_7 satisfying $fit\Phi_7\Phi_5=true$ has $fit\Phi_7\Phi_6=true$ while the fact that $p_1\Phi_5=true$ ensures the existence of such Φ_7. Hence $c\langle\acute{\omega}\downarrow 1,\grave{\omega}\downarrow 1\rangle\langle\langle\acute{\rho},\acute{\upsilon},\acute{\sigma}_0\rangle,\grave{\omega}\dagger 1\S\langle\grave{\sigma}_0\rangle\rangle=true$ and $w\hat{\omega}\Phi_0=true$.

If $\hat{\omega}:F\times F$ note that $tidy(\acute{\omega}\downarrow 2)\acute{\rho}_0=true$ and that $tidy(\acute{\omega}\downarrow 2)\acute{\rho}_2=true$ since $found\Phi_0\hat{\omega}\Phi_0=true$ and $found\Phi_2\hat{\omega}\Phi_2=true$. In

consequence for any I with $1 \le \#(\acute{\omega}\downarrow 2)[\![I]\!]$ there are ν_8 and ν_9 such that $\acute{\rho}_0[\![I]\!]\downarrow\nu_8\dagger 1 = (\acute{\omega}\downarrow 2)[\![I]\!]\downarrow 1\dagger 1 = \acute{\rho}_1[\![I]\!]\downarrow\nu_9\dagger 1$; for all such ν_8 and ν_9, $ravel\acute{\rho}_0\acute{\rho}_0[\![I]\!]\downarrow\nu_8\downarrow 1 : L$ only if $ravel\acute{\rho}_2\acute{\rho}_2[\![I]\!]\downarrow\nu_9\downarrow 1 : L$ so, writing $\acute{\rho}_8 = divert\acute{\rho}_0(\acute{\omega}\downarrow 2)$ and $\acute{\rho}_9 = divert\acute{\rho}_2(\acute{\omega}\downarrow 2)$, $ravel\acute{\rho}_8\acute{\rho}_8[\![I]\!]\downarrow 1\downarrow 1 : L$ only if $ravel\acute{\rho}_9\acute{\rho}_9[\![I]\!]\downarrow 1\downarrow 1 : L$. Hence every ψ subject to $apt\psi(\acute{\rho}_9(\acute{\omega}\downarrow 2), \grave{\omega}\downarrow 2) = true$ is also restricted by $apt\psi(\acute{\rho}_8(\acute{\omega}\downarrow 2), \grave{\omega}\downarrow 2) = true$; inspecting 3.2.5 will now confirm that as $w\hat{\omega}\Pi_2 = true$ necessarily $w\hat{\omega}\Pi_0 = true$.⊁

3.2.8. Proposition.

Suppose that ν_0, $\hat{\zeta}_0$, Π_0 and Π_1 satisfy $\hat{\rho}_0 = (revert\acute{\rho}_0\acute{\rho}_1, revert\grave{\rho}_0\grave{\rho}_1)$, $\theta_0 = \theta_1$, $\pi_0 = \rho_1[\![rec]\!]\downarrow 1\downarrow 1$, $\#\acute{\rho}_0[\![rec]\!] = \#\acute{\rho}_1[\![rec]\!] - 1$, $j\hat{\zeta}_0\nu_0\Pi_0 = true$, $p_1\Pi_1 = true$ and $\bigwedge\{\bigwedge\{known(level\nu\pi_0)\hat{\alpha}\Pi_0 \vee \sim known(level\nu\pi_0)\hat{\alpha}\Pi_1 \mid \nu : N\} \mid \hat{\alpha} : L\times L\} = true$. Writing $\nu_1 = \nu_0 \wedge \#\acute{\rho}_1[\![rec]\!]$ and $\hat{\zeta}_1 = (remit\acute{\zeta}_0, \grave{\zeta}_0 \circ revert\grave{\rho}_0)$ we have $j\hat{\zeta}_1\nu_1\Pi_1 = true$.

⊀Take any pair Π_3 having $p\Pi_3 \wedge pat\Pi_3\nu_1\Pi_1 = true$, and set $\hat{\varepsilon}_0 = (\acute{\theta}_3\downarrow 1, \grave{\theta}_3\downarrow 1)$, $\hat{\varepsilon}_1 = access\hat{\varepsilon}_0\Pi_3$ and

$\Pi_4 = ((revert\acute{\rho}_0\acute{\rho}_3, (\acute{\varepsilon}_0) \S popu\acute{\theta}_0\acute{\theta}_3, restore(\acute{\varepsilon}_0 : L \to update\acute{\varepsilon}_0 dummy\acute{\sigma}_0, \acute{\sigma}_0)\acute{\sigma}_3)$,

$(revert\grave{\rho}_0\grave{\rho}_3, \grave{\theta}_3, \grave{\sigma}_3))$,

so that $(\acute{\zeta}_1\acute{\rho}_3\acute{\theta}_3\acute{\sigma}_3, \grave{\zeta}_1\grave{\rho}_3\grave{\theta}_3\grave{\sigma}_3) = (\acute{\zeta}_0\acute{\rho}_4\acute{\theta}_4\acute{\sigma}_4, \grave{\zeta}_0\grave{\rho}_4\grave{\theta}_4\grave{\sigma}_4)$. To show that $p\Pi_4 \wedge pat\Pi_4\nu_0\Pi_0 = true$ we require also

$\Pi_5 = ((\acute{\rho}_3, \acute{\theta}_3\dagger 1, \acute{\sigma}_3), (\grave{\rho}_3, \grave{\theta}_3\dagger 1, \grave{\sigma}_3))$ and $\Pi_6 = ((\acute{\rho}_4, \acute{\theta}_4\dagger 1, \acute{\sigma}_4), (\grave{\rho}_4, \grave{\theta}_4\dagger 1, \grave{\sigma}_4))$.

Observe that given any π of the form $level\nu\pi_0$ for some ν we can apply 3.2.2 to Π_1, Π_0 and π, obtaining $known\pi\hat{\omega}\Pi_1 = true$ if and only if $known\pi\hat{\omega}\Pi_0 = true$ for all $\hat{\omega}$. Since $fit\Pi_5\Pi_1 = true$ we also know that for all $\hat{\alpha}$ $known\pi\hat{\alpha}\Pi_5 = true$ only if $known\pi\hat{\alpha}\Pi_1 = true$. Hence for all $\hat{\alpha}$ having $known\pi\hat{\alpha}\Pi_5 = true$ $known\pi\hat{\alpha}\Pi_0 = true$ and, as $known\pi\hat{\alpha}\Pi_3 = true$ and $p_0\Pi_3 \wedge p_0\Pi_0 = true$, $area\acute{\alpha}\acute{\sigma}_4 \wedge area\grave{\alpha}\grave{\sigma}_4 = true$. In addition when $\hat{\omega}$ satisfies $found\pi\hat{\omega}\Pi_1 = true$ the definition of *found* in 3.2.1 ensures that $found\pi\hat{\omega}\Pi_0 = true$.

To demonstrate that $fit\Pi_6\Pi_0 = true$ we have only to verify

that $\bigwedge\{known\pi\hat{\alpha}\pi_0 \vee \sim known\pi\hat{\alpha}\pi_6 \mid \hat{\alpha}:L\times L\}=true$ for all π of the form $level\nu\pi_0$ for some ν. For any $\hat{\omega}$ having $known\pi\hat{\omega}\pi_5=true$ we know that $access\hat{\omega}\pi_5=access\hat{\omega}\pi_6$ unless, perhaps, $\acute{\omega}:L$. In this exceptional case $\grave{\omega}:L$ since $kent3\hat{\omega}\pi_3=true$ and $p_1\pi_3=true$ so it follows from the paragraph above that $area\acute{\omega}\acute{\sigma}_4\wedge area\grave{\omega}\grave{\sigma}_4=true$ and $access\hat{\omega}\pi_5=access\hat{\omega}\pi_6$. Applying 3.2.2 to π_5, π_6 and π, for all $\hat{\omega}$ $known\pi\hat{\omega}\pi_5=true$ if and only if $known\pi\hat{\omega}\pi_6=true$. As $known\pi\hat{\alpha}\pi_5\supset known\pi\hat{\alpha}\pi_0$ for all $\hat{\alpha}$, $known\pi\hat{\alpha}\pi_6\supset known\pi\hat{\alpha}\pi_0$ for all $\hat{\alpha}$ and $fit\pi_6\pi_0=true$.

Take any π and ν with $\pi=level\nu\pi_0$; $pop\acute{\upsilon}\circ pop\acute{\upsilon}_0=pop\acute{\upsilon}$ so when n is 3 or 4 for all $\hat{\omega}$ $known\pi\hat{\omega}\pi_n=true$ if and only if $known\pi\hat{\omega}\pi_{n+2}=true$. The argument above thus establishes that for all $\hat{\omega}$ $known\pi\hat{\omega}\pi_3=true$ if and only if $known\pi\hat{\omega}\pi_4=true$; moreover when $\acute{\omega}:L$ and $known\pi\hat{\omega}\pi_4=true$ $area\acute{\omega}\acute{\sigma}_4\wedge area\grave{\omega}\grave{\sigma}_4=true$. As $known\pi\hat{\omega}\pi_3\supset found\pi\hat{\omega}\pi_3$ for all $\hat{\omega}$ we have $known\pi\hat{\omega}\pi_4\supset found\pi\hat{\omega}\pi_4$ for all $\hat{\omega}$.

If $\acute{\varepsilon}_0:V$ $access\hat{\varepsilon}_0\pi_4=\hat{\varepsilon}_1$ automatically, whilst otherwise $area\acute{\varepsilon}_0\acute{\sigma}_3\wedge area\grave{\varepsilon}_0\grave{\sigma}_3=true$ as $p_0\pi_3=true$ so $area\acute{\varepsilon}_0\acute{\sigma}_4\wedge area\grave{\varepsilon}_0\grave{\sigma}_4=true$ and $access\hat{\varepsilon}_0\pi_4=\hat{\varepsilon}_1$. Because $\nu_1\leq\#\acute{\rho}_0[\![\mathrm{rec}]\!]+1$ $level\nu_1\pi_1=level\nu_1\pi_0$ and we may take ν to be ν_1 itself. Unless $\grave{\varepsilon}_0:V$ or $area\grave{\varepsilon}_0\grave{\sigma}_1=false$ $known(level\nu_1\pi_1)\hat{\varepsilon}_0\pi_3=true$; since we have shown that for all $\hat{\omega}$ $known(level\nu_1\pi_0)\hat{\omega}\pi_3=true$ only if $known(level\nu_1\pi_0)\hat{\omega}\pi_4=true$, unless $\grave{\varepsilon}_0:V$ or $area\grave{\varepsilon}_0\grave{\sigma}_0=false$ $known(level\nu_1\pi_0)\hat{\varepsilon}_0\pi_4=true$. Likewise, as $found(level\nu_1\pi_3)\hat{\varepsilon}_1\pi_3=true$ and for all $\hat{\omega}$ $known(level\nu_1\pi_3)\hat{\omega}\pi_3=true$ if and only if $known(level\nu_1\pi_3)\hat{\omega}\pi_4=true$, we can assert that $found(level\nu_1\pi_3)\hat{\varepsilon}_1\pi_4=true$. Since $p_0\pi_0=true$ and $\nu_1\leq\nu_0$, $known(level\nu_0\pi_0)\hat{\varepsilon}_0\pi_4=true$ and $found(level\nu_0\pi_4)(access\hat{\varepsilon}_0\pi_4)\hat{\omega}\pi_4=true$, giving $pat\pi_4\nu 0\pi 0=true$.

Any π and ν subject to $\pi=level\nu\pi_4$ also satisfy either $\pi=level\nu\pi_0$ or $\pi=\pi_4$. We have already established that $\bigwedge\{found\pi\hat{\omega}\pi_4\vee\sim known\pi\hat{\omega}\pi_4\mid\hat{\omega}:L\times L\}=true$ when $\pi=level\nu\pi_0$, so it remains only to discuss $known\pi\hat{\omega}\pi_4$ when $\pi=\pi_4$. For all $\hat{\omega}$

$known\hat{\pi}_4\hat{\omega}\hat{\pi}_4 = \bigvee\{\bigvee\{seen3\nu\hat{\omega}\hat{\omega}_1\hat{\pi}_4 \wedge (hoten\hat{\omega}_1\hat{\rho}_4 \vee gyven\hat{\omega}_1\hat{\upsilon}_4) \mid \nu:N\} \mid \hat{\omega}_1:W\times W\}$

$= \bigvee\{\bigvee\{seen3\nu\hat{\omega}\hat{\omega}_1\hat{\pi}_4 \wedge (hoten\hat{\omega}_1\hat{\rho}_6 \vee gyven\hat{\omega}_1\hat{\upsilon}_6 \vee \hat{\omega}_1 = \hat{\varepsilon}_0) \mid \nu:N\} \mid \hat{\omega}_1:L\times L\}$

$= (known\hat{\pi}_0\hat{\omega}\hat{\pi}_4 \vee \hat{\omega} = \hat{\varepsilon}_0 \vee \bigvee\{seen3\nu\hat{\omega}\hat{\varepsilon}_1\hat{\pi}_3 \mid \nu:N\})$

$= (known\hat{\pi}_0\hat{\omega}\hat{\pi}_4 \vee \hat{\omega} = \hat{\varepsilon}_0 \vee \hat{\omega} = \hat{\varepsilon}_1)$;

the last link in this train of reasoning holds because either $\hat{\varepsilon}_1:L^*\times L^*$ and $\bigwedge\{known(level\nu_1\hat{\pi}_3)\hat{\alpha}\hat{\pi}_3 \vee \sim gyven\hat{\alpha}\hat{\varepsilon}_1 \mid \hat{\alpha}:L\times L\} = true$ or $\bigvee\{seen3\nu\hat{\omega}\hat{\varepsilon}_1\hat{\pi}_3 \mid \nu:N\} = (\hat{\omega} = \hat{\varepsilon}_1)$. Since $found\hat{\pi}_0\hat{\omega}\hat{\pi}_4 = true$ for all $\hat{\omega}$ having $known\hat{\pi}_0\hat{\omega}\hat{\pi}_4 = true$, $found\hat{\pi}_4\hat{\omega}\hat{\pi}_4 = true$ for all $\hat{\omega}$ having $known\hat{\pi}_4\hat{\omega}\hat{\pi}_4 = true$. In addition, if $\hat{\omega}_0$ and $\hat{\omega}_1$ have $kent3\hat{\omega}_0\hat{\pi}_4 \wedge kent3\hat{\omega}_1\hat{\pi}_4 = true$, $known\hat{\pi}_4\hat{\omega}_0\hat{\pi}_3 \wedge known\hat{\pi}_4\hat{\omega}_0\hat{\pi}_3 = true$ so when $\acute{\omega}_0:L$ and $\acute{\omega}_0 = \acute{\omega}_1$ necessarily $\grave{\omega}_0 = \grave{\omega}_1$. The other constituents of $p_0\hat{\pi}_4$ follow readily from the corresponding parts of $p_0\hat{\pi}_3$, so $p_0\hat{\pi}_4 = true$.

By induction on ν we can show that for all ν, $\hat{\omega}_0$ and $\hat{\omega}_1$ writing $\hat{\pi}_2 = (\mathbf{\Box}q_0\times\mathbf{\Box}q_0)\hat{\pi}_3$ gives $seen1\nu\hat{\omega}_0\hat{\omega}_1\hat{\pi}_4 \wedge kent1\hat{\omega}_1\hat{\pi}_3 \wedge known\hat{\pi}_4((q_0\times q_0)\hat{\omega}_1)\hat{\pi}_2 \supset kent1\hat{\omega}_0\hat{\pi}_3$; the condition $known\hat{\pi}_4((q_0\times q_0)\hat{\omega}_1)\hat{\pi}_2 = true$ is essential to ensure that $found\hat{\pi}_4\hat{\omega}_1\hat{\pi}_3 = true$. For all $\hat{\omega}_1$ $hoten\hat{\omega}_1\hat{\rho}_4 \vee gyven\hat{\omega}_1\hat{\upsilon}_4 \supset kent1\hat{\omega}_1\hat{\pi}_3 \wedge known\hat{\pi}_4\hat{\omega}_1\hat{\pi}_3$, so we have $kent1\hat{\omega}_0\hat{\pi}_4 \supset kent1\hat{\omega}_0\hat{\pi}_3$ for all $\hat{\omega}_0$. As $p_0\hat{\pi}_4 \wedge p\hat{\pi}_3 = true$ 3.2.7 implies that $p\hat{\pi}_4 = true$.

Hence $p\hat{\pi}_4 \wedge pat\hat{\pi}_4\nu_0\hat{\pi}_0 = true$ and $a\langle\acute{\zeta}_0\acute{\rho}_4\acute{\upsilon}_4\acute{\sigma}_4, \grave{\zeta}_0\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4\rangle = true$. The original pair $\hat{\pi}_3$ was chosen at random from among those having $p\hat{\pi}_3 \wedge pat\hat{\pi}_3\nu_1\hat{\pi}_1 = true$, so $a\langle\acute{\zeta}_1\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3, \grave{\zeta}_1\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3\rangle = true$ for all such $\hat{\pi}_3$, thus confirming that $j\hat{\zeta}_1\nu_1\hat{\pi}_1 = true$.✣

Having confirmed that the predicate j is appropriate to the situation pertaining on exit from a block, we can proceed to the proof that *new* stack semantics is equivalent with *new* store semantics except for the fact that its recursive declarations by incidence are akin to recursive declarations by reference.

3.3. Two comparable mechanisms.

3.3.1. Lemma.

If E:Exp satisfies $G[\![E]\!]=true$ then $E[\![E]\!]=true$.

⊲Suppose that ψ_0, $\hat{\zeta}_0$, ν_0 and $\hat{\pi}_0$ satisfy $apt\psi_0\hat{\rho}_0=true$, $rent[\![E]\!]\psi=true$, $j\hat{\zeta}_0\nu_0\hat{\pi}_0=true$ and $e[\![E]\!]\nu_0\hat{\rho}_0=true$. To demonstrate that $c\langle\mathscr{E}[\![E]\!]\hat{\zeta}_0,\mathscr{E}[\![e[\![E]\!]\psi_0]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$ select any pair $\hat{\pi}_1$ satisfying $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$, and define

$\psi_1=\psi_0[false*/\mathscr{S}[\![E]\!]][opts(\mathscr{X}[\![E]\!])\psi_0/\mathscr{X}[\![E]\!]]$, $\hat{\zeta}_1=\langle remit\acute{\zeta}_0,\grave{\zeta}_0\circ revert\grave{\rho}_0\rangle$ and $\langle\acute{\alpha}*,\grave{\alpha}*\rangle=\langle news(\#\mathscr{S}[\![E]\!])\acute{\sigma}_1,news(\#\mathscr{S}[\![g[\![E]\!]\psi_1]\!])\grave{\sigma}_1\rangle$. Introducing

$\acute{\rho}_3=\acute{\rho}_1[\langle\acute{\rho}_1,\acute{\upsilon}_1,\acute{\sigma}_1\rangle//\mathrm{rec}][\acute{\alpha}*//\mathscr{S}[\![E]\!]][dummy*//\mathscr{X}[\![E]\!]]$,

$\acute{\sigma}_3=updates\acute{\alpha}*dummy*\acute{\sigma}_1$ and $\nu_1=\nu_0\wedge\#\acute{\rho}_3[\![\mathrm{rec}]\!]$ we see that

$\langle\mathscr{E}[\![E]\!]\hat{\zeta}_0\hat{\rho}_1\hat{\upsilon}_1\hat{\sigma}_1,\mathscr{E}[\![e[\![E]\!]\psi_0]\!]\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=\langle\mathscr{E}[\![E]\!]\acute{\zeta}_1\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\mathscr{E}[\![g[\![E]\!]\psi_1]\!]\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle$

where

$\hat{\rho}_2=\langle\acute{\rho}_1[\langle\acute{\rho}_1,\acute{\upsilon}_1,\acute{\sigma}_1\rangle//\mathrm{rec}][\alpha*//\mathscr{S}[\![E]\!]][\mathscr{Q}[\![E]\!]\acute{\zeta}_1\acute{\rho}_3\acute{\upsilon}_1\acute{\sigma}_3//\mathscr{X}[\![E]\!]],$

$fix(\lambda\rho.\grave{\rho}_1[\grave{\alpha}*/\mathscr{S}[\![g[\![E]\!]\psi_1]\!]][\mathscr{Q}[\![g[\![E]\!]\psi_1]\!]\grave{\zeta}_1\rho\grave{\upsilon}_1/\mathscr{X}[\![g[\![E]\!]\psi_1]\!]])\rangle$,

$\hat{\upsilon}_2=\acute{\upsilon}_1$ and

$\hat{\sigma}_2=\langle updates\acute{\alpha}*(\mathscr{P}[\![E]\!]\acute{\zeta}_1\acute{\rho}_3\acute{\upsilon}_1\acute{\sigma}_3)\acute{\sigma}_1,updates\grave{\alpha}*(\mathscr{P}[\![g[\![E]\!]\psi_1]\!]\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_1)\grave{\sigma}_1\rangle$.

From 3.1.5 $g[\![E]\!]\nu_1\hat{\rho}_2=true$, and plainly $apt\psi_1\hat{\rho}_2=true$, so to show that $a\langle\mathscr{E}[\![E]\!]\acute{\zeta}_1\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\mathscr{E}[\![g[\![E]\!]\psi_1]\!]\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle=true$ it suffices to convince ourselves that $p\hat{\pi}_2$, $fit\hat{\pi}_2\hat{\pi}_2$ and $j\hat{\zeta}_1\nu_1\hat{\pi}_2$ are all $true$.

Any $\hat{\omega}$ satisfying $kent3\hat{\omega}\hat{\pi}_1=true$ cannot have $\acute{\omega}:\acute{\alpha}*$ or $\grave{\omega}:\grave{\alpha}*$, as then $area\acute{\omega}\acute{\sigma}_1=false$ or $area\grave{\omega}\grave{\sigma}_1=false$ in contradiction to the fact that $p_0\hat{\pi}_1=true$; consequently any $\hat{\omega}$ with $known(level\nu\hat{\pi}_1)\hat{\omega}\hat{\pi}_1=true$ for some ν is such that $access\hat{\omega}\hat{\pi}_1=access\hat{\omega}\hat{\pi}_2=access\hat{\omega}\hat{\pi}_3$. We can therefore apply 3.2.2 to arrive at the conclusion that for all ν and $\hat{\omega}$ $known(level\nu\hat{\pi}_1)\hat{\omega}\hat{\pi}_1=true$ if and only if $known(level\nu\hat{\pi}_1)\hat{\omega}\hat{\pi}_2=true$. Using 1.4.6 to write

$\hat{\sigma}_4=\langle\acute{\alpha}*\S\mathscr{Q}[\![E]\!]\acute{\zeta}_1\acute{\rho}_4\acute{\upsilon}_1\acute{\sigma}_4,$

$swap(\mathscr{S}[\![E]\!]\S\mathscr{X}[\![E]\!])(\mathscr{S}[\![g[\![E]\!]\psi_1]\!]\S\mathscr{X}[\![g[\![E]\!]\psi_1]\!])(\grave{\alpha}*\S\mathscr{Q}[\![g[\![E]\!]\psi_1]\!]\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_1)\rangle$,

$\theta_5 = \langle access\langle \upsilon_5 + 1, \delta_5 + 1\rangle \pi_2, \ldots, access\langle \upsilon_5 + \#\upsilon_5, \delta_5 + \#\delta_5\rangle \pi_2\rangle$ and $\theta_6 = \theta_4 \S \theta_5$, it is plain from the definition of *seen* in 3.2.1 that for all ω $known\pi_2\omega\pi_2 \supset known\pi_1\omega\pi_2 \vee given\omega\theta_6$. As $p_0\pi_1 = true$ $known(level\nu\pi_1)\omega\pi_1 \supset found(level\nu\pi_1)\omega\pi_1$ for all ν and ω, while as $G[\![E]\!] = true$ $given\omega\theta_6 \supset found\pi_2\omega\pi_2$ for all ω. Accordingly $\bigwedge\{found(level\nu\pi_2)\omega\pi_2 \vee \sim known(level\nu\pi_2)\omega\pi_2 \mid \omega : W \times W\} = true$ because $level\nu\pi_2 = (\nu \leq \#\rho_3[\![rec]\!] \rightarrow level\nu\pi_1, \pi_2)$ for all ν. From the description of $kent3\omega\pi_2$ as $kent3\omega\pi_1 \vee given\omega\theta_6$ and from the nature of $\rho_2[\![rec]\!]$ it follows that not only does $p_0\pi_1 = true$ but $p_0\pi_2 = true$ also.

Assume that for some ν and all ω_0 and ω_1 $seen1\nu\omega_0\omega_1\pi_2 \wedge (kent1\omega_1\pi_1 \vee given\omega_1\theta_6) \supset kent1\omega_0\pi_1 \vee given\omega_1\theta_6$ (which is certainly the case when $\nu = 0$) and take any ω_0 and ω_1 having $seen1(\nu+1)\omega_0\omega_1\pi_2 \wedge (kent1\omega_1\pi_1 \vee given\omega_1\theta_6) = true$. Should $kent1\omega_1\pi_1$ be *true* we know that $known\pi_0((q_0 \times q_0)\omega_1)((\wp q_0 \times \wp q_0)\pi_1)$ will be *true* from 3.2.6 and therefore that $found\pi_0\omega_1\pi_1$ will be *true*. Thus if $\omega_1 : L$ or $\omega_1 : L$ and if $\omega_2 = access\omega_1\pi_2$, either $\omega_2 = access\omega_1\pi_1$ or $given\omega_2\theta_6 = true$, so that $seen1\nu\omega_0\omega_2\pi_2 \wedge (kent1\omega_2\pi_2 \vee given\omega_2\theta_6) = true$. If $\omega_1 : J \times J$ either $found\pi_0\omega_1\pi_1 = true$ or $found\pi_2\omega_1\pi_2 = true$; consequently every ω_2 having $hoten\omega_2\langle revert(\omega_1 + 2)\rho_2, \omega_1 + 2\rangle \vee given\omega_2\langle pop(\omega_1 + 3)\upsilon_2, \omega_1 + 3\rangle = true$ has $kent1\omega_2\pi_1 \vee given\omega_2\theta_6 = true$, while for some such ω_2 $seen1\nu\omega_0\omega_2\pi_2 = true$. If $\omega_1 : F \times F$ then $kent1\omega_1\pi_1 = true$ so $tidy(\omega_1 + 2)\rho_1 = true$ and $hoten\omega_2\langle divert\rho_1(\omega_1 + 2), \omega_1 + 2\rangle \supset hoten\langle divert\rho_2(\omega_1 + 2), \omega_1 + 2\rangle$ for all ω_2; in particular there exists some ω_2 having $seen1\nu\omega_1\omega_2\pi_2 \wedge kent1\omega_2\pi_1 = true$. The reasoning being similar when $\omega_1 : L^* \times L^*$ or $\omega_1 : J \times J$ and vacuous when $\omega_1 : B \times B$ we may apply the induction hypothesis under all circumstances to obtain $kent1\omega_0\pi_1 \vee given\omega_0\theta_6 = true$.

As $hoten\omega_1\rho_2 \vee given\omega_1\theta_2 \supset kent1\omega_1\pi_1 \vee given\omega_1\theta_6$ for all ω_1 we can now assert that $kent1\omega_0\pi_2 \supset kent1\omega_0\pi_1 \vee given\omega_0\theta_6$ for all

$\hat{\omega}_0$ and that by the proof of 3.2.7 $w\hat{\omega}_0\hat{\pi}_2=true$ whenever $kent1\hat{\omega}_0\hat{\pi}_1=true$. Moreover since $p_0\hat{\pi}_2=true$ $w_0\hat{\omega}_0\hat{\pi}_2=true$ whenever $gyven\hat{\omega}_0\hat{0}_6=true$, so $w_0\hat{\omega}_0\hat{\pi}_2=true$ whenever $kent1\hat{\omega}_0\hat{\pi}_2=true$; hence we may assume that $p_1\hat{\pi}_2=true$ as well as that $known(level\nu\hat{\pi}_0)\hat{\alpha}\hat{\pi}_1\supset known(level\nu\hat{\pi}_0)\hat{\alpha}\hat{\pi}_0$ for all $\hat{\alpha}$ and ν. As $fit\hat{\pi}_1\hat{\pi}_0=true$ $j\hat{\zeta}_0\nu_0\hat{\pi}_1=true$ and we can apply 3.2.8 to $\hat{\pi}_1$ and $\hat{\pi}_2$ to obtain $j\hat{\zeta}_1\nu_1\hat{\pi}_1=true$.

From this equality and the stipulation that $G[\![E]\!]=true$ we now know that $w\hat{\omega}\hat{\pi}_2=true$ for all $\hat{\omega}$:J×J such that $gyven\hat{\omega}\hat{0}_6=true$. As we have already established that $p_0\hat{\pi}_2=true$ and that $w\hat{\omega}\hat{\pi}_2=true$ for all $\hat{\omega}$:W×W having $kent1\hat{\omega}\hat{\pi}_1=true$, $p\hat{\pi}_2=true$. Finally, for all $\hat{\pi}_1$ having $p\pi_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$ $a\langle\mathscr{E}[\![E]\!]\acute{\zeta}_0\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{E}[\![\iota[\![E]\!]\psi_0]\!]\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$, so $c\langle\mathscr{E}[\![E]\!]\acute{\zeta}_0,\mathscr{E}[\![\iota[\![E]\!]\psi_0]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$ (for all the suitable $\hat{\zeta}_0$, ν and $\hat{\pi}_0$) and $E[\![E]\!]=true$.⊁

3.3.2. Lemma.

For all I:Ide and B:Bas $G[\![I]\!]\wedge G[\![B]\!]=true$; when Φ:Abs has a body E:Exp such that $L[\![E]\!]=true$ $G[\![\Phi]\!]=true$.

⊰Fix attention on one particular collection comprising ψ, $\hat{\zeta}$, ν, $\hat{\pi}_0$ and $\hat{\pi}_1$ such that $apt\psi\hat{\rho}_0=true$, $j\hat{\zeta}\nu\hat{\pi}_0=true$ and $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$; set $\hat{\pi}=level\nu\hat{\pi}_0$.

Let I be such that $g[\![I]\!]\nu\acute{\rho}_0=true$; writing $\hat{\delta}$ for $\langle ravel\acute{\rho}_1\acute{\rho}_1[\![I]\!]\downarrow1\downarrow1,\grave{\rho}_1[\![I]\!]\downarrow1\rangle$ we know that $\grave{\delta}$:L or $\grave{\delta}$:V as $apt\psi\hat{\rho}_1=true$. Let $\grave{\alpha}=new\grave{\sigma}_1$, $\grave{\beta}=(\grave{\delta}:L\rightarrow(area\grave{\delta}\grave{\sigma}_1\rightarrow hold\grave{\delta}\grave{\sigma}_1,\tau),\grave{\delta})$, $\grave{\varepsilon}=(\psi[\![I]\!]\downarrow1=true\rightarrow\grave{\alpha},\grave{\delta})$ and

$\hat{\pi}_2=\langle\langle\acute{\rho}_1,\langle\acute{\delta}\rangle\S\acute{\upsilon}_1,\acute{\sigma}_1\rangle,\langle\grave{\rho}_1,\langle\grave{\varepsilon}\rangle\S\grave{\upsilon}_1,(\psi[\![I]\!]\downarrow1=true\rightarrow update\grave{\alpha}\grave{\beta}\grave{\sigma}_1,\grave{\sigma}_1)\rangle\rangle$, so that $\langle\mathscr{I}[\![I]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{I}[\![g[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=\langle\acute{\zeta}\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\grave{\zeta}\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle$. To show that $a\langle\mathscr{I}[\![I]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{I}[\![g[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$ it therefore suffices to prove that $p\hat{\pi}_2\wedge pat\hat{\pi}_2\nu\hat{\pi}_2=true$. As $g[\![I]\!]\nu\acute{\rho}_1=true$ and $p_0\hat{\pi}_1=true$, $\acute{\rho}_1[\![I]\!]\downarrow1\downarrow2\leq\nu$ and $lead1(\acute{\rho}_1[\![I]\!])>\#\acute{\rho}_1[\![I]\!]-\#\acute{\rho}[\![I]\!]$. In addition

$\mathfrak{A}q_0\grave{\rho}_1=\mathfrak{A}q_0(ravel\acute{\rho}_1\grave{\rho}_1)$ so $hoten\langle\acute{\delta},q_0\grave{\delta}\rangle\langle revert\acute{\rho}\acute{\rho}_1,\mathfrak{A}q_0(revert\acute{\rho}\grave{\rho}_1)\rangle=true$ and $known\hat{\pi}((q_0\times q_0)(access\hat{\delta}\hat{\pi}_1))((\mathfrak{P}q_0\times\mathfrak{P}q_0)\hat{\pi}_1)=true$; this entails $found\hat{\pi}(access\langle\acute{\delta},\grave{\varepsilon}\rangle\hat{\pi}_2)\hat{\pi}_2=true$ unless $\grave{\beta}:L^*$. If $\psi[\![I]\!]\downarrow 1=true$ then $area\grave{\varepsilon}\grave{\sigma}_0=false$, whilst if $\grave{\varepsilon}:L$ and $\psi[\![I]\!]\downarrow 1=false$ $\acute{\delta}:L$ (as $apt\psi\hat{\rho}_1=true$) and $known\hat{\pi}\hat{\delta}\hat{\pi}_1=true$. Since $fit\hat{\pi}_1\hat{\pi}_0=true$ it follows that $pat\hat{\pi}_2\nu\hat{\pi}_0=true$; furthermore induction establishes that $kent\nu_0\hat{\omega}\hat{\pi}_2\supset(\hat{\omega}=\langle\acute{\delta},\grave{\varepsilon}\rangle\vee kent\nu_0\hat{\omega}\hat{\pi}_1)$ for all ν_0 and $\hat{\omega}$ while 3.2.2 shows that $known(level\nu_1\acute{\pi}_1)\hat{\omega}\hat{\pi}_2\supset known(level\nu_1\acute{\pi}_1)\hat{\omega}\hat{\pi}_1$ for all ν_1 and $\hat{\omega}$, so $p\hat{\pi}_2=true$ and $a\langle\mathscr{g}[\![I]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{g}[\![g[\![I]\!]\psi]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$.

Notice that this result would still hold even if all we knew about $\hat{\zeta}$ was that $a\langle\acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma},\grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle=true$ for every $\hat{\pi}$ having $p\hat{\pi}\wedge pat\hat{\pi}\nu\hat{\pi}_0=true$ and $\acute{\upsilon}\downarrow 1:L\wedge known(level(\nu_2-1)\hat{\pi}_0)\langle\acute{\upsilon}\downarrow 1,\grave{\upsilon}\downarrow 1\rangle\hat{\pi}_0=false$ for some ν_2 such that $\nu_2<\acute{\rho}_0[\![I]\!]\downarrow 1\downarrow 2$. This follows from the fact that when $ravel\acute{\rho}_1\acute{\rho}_1[\![I]\!]\downarrow 1\downarrow 1:L$ we have $\grave{\rho}_1[\![I]\!]\downarrow 1:L$ and $known(level(\acute{\rho}_1[\![I]\!]\downarrow 1\downarrow 2-1)\hat{\pi}_1)\langle ravel\acute{\rho}_1\acute{\rho}_1[\![I]\!]\downarrow 1\downarrow 1,\grave{\rho}_1[\![I]\!]\downarrow 1\rangle\hat{\pi}_1=false$ as $p_0\hat{\pi}_1=true$.

The proof that $a\langle\mathscr{g}[\![B]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{g}[\![g[\![B]\!]\psi]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$ being palpable we turn to the one for $\langle\mathscr{g}[\![\Phi]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{g}[\![g[\![\Phi]\!]\psi]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle$. Suppose that $torn[\![\Phi]\!]\psi=true$ and that $g[\![\Phi]\!]\nu\acute{\rho}_0=true$; define $\langle\langle\acute{\xi},\acute{\rho}_9\rangle,\langle\grave{\xi},\grave{\rho}_9\rangle\rangle=\langle\mathscr{g}[\![\Phi]\!]\acute{\rho}_1,\mathscr{g}[\![g[\![\Phi]\!]\psi]\!]\grave{\rho}_1\rangle$ and $\hat{\pi}_3=\langle\langle\acute{\rho}_1,\langle\langle\acute{\xi},\acute{\rho}_9\rangle\S\acute{\upsilon}_1,\acute{\sigma}_1\rangle,\langle\grave{\rho}_1,\langle\langle\grave{\xi},\grave{\rho}_9\rangle\S\grave{\upsilon}_1,\grave{\sigma}_1\rangle\rangle$. Because $g[\![\Phi]\!]\nu\acute{\rho}_1=true$ we have $tidy\acute{\rho}_9(revert\acute{\rho}\acute{\rho}_1)=true$, and because $\mathfrak{A}q_0\grave{\rho}_1=\mathfrak{A}q_0(ravel\acute{\rho}_1\grave{\rho}_1)$ we know that $\mathfrak{A}q_0(divert\grave{\rho}_1\grave{\rho}_9)=\mathfrak{A}q_0(ravel(divert\acute{\rho}_1\acute{\rho}_9)(divert\grave{\rho}_1\grave{\rho}_9))$; consequently $found\hat{\pi}\langle\acute{\upsilon}_3\downarrow 1,\grave{\upsilon}_3\downarrow 1\rangle\hat{\pi}=true$. Together with the fact that $fit\hat{\pi}_1\hat{\pi}_0=true$ this ensures that $pat\hat{\pi}_3\nu\hat{\pi}_0=true$. Again induction shows that $kent\nu_0\hat{\omega}\hat{\pi}_3\supset(\hat{\omega}=\langle\acute{\upsilon}_3\downarrow 1,\grave{\upsilon}_3\downarrow 1\rangle\vee kent\nu_0\hat{\omega}\hat{\pi}_1)$ for all ν_0 and $\hat{\omega}$, and 3.2.2 shows that $known(level\nu_1\acute{\pi}_1)\hat{\omega}\hat{\pi}_3\supset known(level\nu_1\acute{\pi}_1)\hat{\omega}\hat{\pi}_1$ for all ν_1 and $\hat{\omega}$. Because $p_0\hat{\pi}_1=true$ we must have $p_0\hat{\pi}_3=true$; moreover $w\hat{\omega}\hat{\pi}_1=true$ and $found\hat{\pi}_3\hat{\omega}\hat{\pi}_3=true$ for all $\hat{\omega}$ having $kent1\hat{\omega}\hat{\pi}_1=true$. Accordingly from the definition in 3.2.5

$w\hat{\omega}\hat{\pi}_3=true$ for all $\hat{\omega}$ subject to $kent1\hat{\omega}\hat{\pi}_1=true$, and to prove that $p\hat{\pi}_3=true$ it remains only to verify that $w\langle\hat{\upsilon}_3{+}1,\hat{\upsilon}_3{+}1\rangle\hat{\pi}_3=true$. We shall do this on the assumption that Φ is of the form fnI..E, the proof being no more profound when Φ is $fnI_1,\ldots,I_n..E$.

⊲Let $\hat{\zeta}_0$, ν_2, $\hat{\pi}_4$ and $\hat{\pi}_5$ be entities having $\bigwedge\{apt\psi_0\langle\acute{\rho}_5,\grave{\rho}_9\rangle\vee\sim apt\psi_0\langle divert\acute{\rho}_3\acute{\rho}_9,\grave{\rho}_9\rangle\mid\psi_0\}=true$, $tidy\acute{\rho}_9(level\nu_2\hat{\pi}_4{+}1)=true$, $j\hat{\zeta}_0\nu_2\hat{\pi}_4=true$, $fit\hat{\pi}_5\hat{\pi}_5=true$ and $\hat{\pi}_5=\langle\langle divert\acute{\rho}_4\acute{\rho}_9,\langle\acute{\upsilon}_5{+}1\rangle\S\acute{\upsilon}_4,\acute{\sigma}_4\rangle,\langle divert\grave{\rho}_4\grave{\rho}_9,\langle\grave{\upsilon}_5{+}1\rangle\S\grave{\upsilon}_4,\grave{\sigma}_4\rangle\rangle$. We shall show that $c\langle\acute{\xi}\acute{\zeta}_0\circ revert\acute{\rho}_4,\grave{\xi}(\grave{\zeta}_0\circ revert\grave{\rho}_4)\rangle\hat{\pi}_5=true$ by selecting arbitrarily a pair $\hat{\pi}_6$ constrained by $p\hat{\pi}_6\wedge fit\hat{\pi}_6\hat{\pi}_5=true$. For this pair we introduce $\nu_3=\#\acute{\rho}_6[\![rec]\!]+1$, $\hat{\zeta}_1=\langle remit\acute{\zeta}_0,\grave{\zeta}_0\circ revert\grave{\rho}_4\rangle$, $\hat{\pi}_8=\langle\langle revert\acute{\rho}_4\acute{\rho}_6,\acute{\upsilon}_6\dagger1,\acute{\sigma}_6\rangle,\langle revert\grave{\rho}_4\grave{\rho}_6,\grave{\upsilon}_6\dagger1,\grave{\sigma}_6\rangle\rangle$ and $\hat{\pi}_7=\langle\langle divert\acute{\rho}_8\acute{\rho}_9[\hat{\pi}_8//rec][\acute{\upsilon}_6{+}1:L\rightarrow hold(\acute{\upsilon}_6{+}1)\acute{\sigma}_6,\acute{\upsilon}_6{+}1//I],\acute{\upsilon}_6\dagger1,\acute{\sigma}_6\rangle$, $\langle\grave{\rho}_6[\grave{\upsilon}_6{+}1:L\rightarrow hold(\grave{\upsilon}_6{+}1)\grave{\sigma}_6,\grave{\upsilon}_6{+}1/I],\grave{\upsilon}_6\dagger1,\grave{\sigma}_6\rangle\rangle$; the equations of appendix 3 make it plain that $\acute{\xi}\acute{\zeta}_0\acute{\rho}_6\acute{\upsilon}_6\acute{\sigma}_6=\mathscr{L}[\![E]\!]\acute{\zeta}_1\acute{\rho}_7\acute{\upsilon}_7\acute{\sigma}_7$ and that $\grave{\xi}\grave{\zeta}_0\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6=\mathscr{L}[\![\grave{\epsilon}[\![E]\!]\psi[false/I]]\!]\grave{\zeta}_1\grave{\rho}_7\grave{\upsilon}_7\grave{\sigma}_7$ whatever the value of $opt[\![I]\!]\psi$.

⊲Since $fit\hat{\pi}_6\hat{\pi}_5=true$ we know that $\mathfrak{U}q_0\acute{\rho}_6=\mathfrak{U}q_0(divert\acute{\rho}_8\acute{\rho}_9)$, and the fact that $tidy\acute{\rho}_9\acute{\rho}_4=true$ now assures us that for all $\hat{\omega}$ $hoten\hat{\omega}\langle divert\acute{\rho}_8\acute{\rho}_9,\grave{\rho}_6\rangle\supset hoten\hat{\omega}\hat{\rho}_6$; consequently for every $\hat{\omega}$ $known\hat{\pi}_7\hat{\omega}\hat{\pi}_7\supset known\hat{\pi}_6\hat{\omega}\hat{\pi}_6$. From 3.2.2 we have in addition that for all $\hat{\omega}$ and ν_4 $known(level\nu_4\hat{\pi}_7)\hat{\omega}\hat{\pi}_7\supset known(level\nu_4\hat{\pi}_7)\hat{\omega}\hat{\pi}_6$ when $\nu_4\leq\nu_3$. The definition of *found* in 3.2.1 is such that any $\hat{\omega}$ which satisfies $found\hat{\pi}_6\hat{\omega}\hat{\pi}_6=true$ also satisfies $found\hat{\pi}_8\hat{\omega}\hat{\pi}_7=true$ unless $\hat{\omega}:L^*\times L^*$; to see this when $\hat{\omega}:F\times F$, for instance, note that $tidy\acute{\rho}_9\acute{\rho}_8\wedge tidy(\acute{\omega}{+}2)\acute{\rho}_6\supset tidy(\acute{\omega}{+}2)\acute{\rho}_8$. As $p_0\hat{\pi}_6=true$, we know that $found(level\nu_4\hat{\pi}_6)\hat{\omega}\hat{\pi}_6=true$ whenever $known(level\nu_4\hat{\pi}_6)\hat{\omega}\hat{\pi}_6=true$ and thus that for all $\hat{\omega}$ and ν_4 $found(level\nu_4\hat{\pi}_7)\hat{\omega}\hat{\pi}_7=true$ if $known(level\nu_4\hat{\pi}_7)\hat{\omega}\hat{\pi}_7=true$. Routine checking of the remaining

clauses now confirms that $p_0\hat{\pi}_7=true$.

By induction on ν_1 we can establish that for all $\hat{\omega}_0$, $\hat{\omega}_1$ and ν_1 we have $seen1\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_7 \wedge kent1\hat{\omega}_1\hat{\pi}_6 \supset kent1\hat{\omega}_0\hat{\pi}_6$; this is due to the relation $kent1\hat{\omega}_1\hat{\pi}_6 \supset found\hat{\pi}_6\hat{\omega}_1\hat{\pi}_6$ set up in 3.2.6. Hence for every $\hat{\omega}_0$ $kent1\hat{\omega}_0\hat{\pi}_7 \supset kent1\hat{\omega}_0\hat{\pi}_6$, and $p_0\hat{\pi}_7$ being $true$, 3.2.7 allows us to deduce that $p\hat{\pi}_7=true$.

From 3.2.2 it is clear that for every $\hat{\omega}$ and ν_4 $known(level\nu_4\hat{\pi}_4)\hat{\omega}\hat{\pi}_{n+1} \supset known(level\nu_4\hat{\pi}_4)\hat{\omega}\hat{\pi}_n$ when n is 4, 6 or 7. Since $fit\hat{\pi}_6\hat{\pi}_5=true$ we even know that for all $\hat{\alpha}$ and $\nu_4 \leq \nu_3-1$ $known(level\nu_4\hat{\pi}_4)\hat{\alpha}\hat{\pi}_6 \supset known(level\nu_4\hat{\pi}_4)\hat{\alpha}\hat{\pi}_5$. Consequently for each $\hat{\alpha}$ we have $known(level\nu_4\hat{\pi}_4)\hat{\alpha}\hat{\pi}_8 \supset known(level\nu_4\hat{\pi}_4)\hat{\alpha}\hat{\pi}_4$ if $\nu_4 \leq \nu_3-1$ and $known(level\nu_4\hat{\pi}_8)\hat{\alpha}\hat{\pi}_7 \supset known(level\nu_4\hat{\pi}_8)\hat{\alpha}\hat{\pi}_8$ universally. Moreover, as $p_0\hat{\pi}_7=true$, $found(level\nu_4\hat{\pi}_8)\hat{\omega}\hat{\pi}_8=true$ whenever $known(level\nu_4\hat{\pi}_8)\hat{\omega}\hat{\pi}_8=true$ so $p_0\hat{\pi}_8=true$ and by 3.2.7 $p_1\hat{\pi}_8=true$; hence from the facts that $j\hat{\zeta}_0\nu_2\hat{\pi}_4=true$ and $fit\hat{\pi}_8\hat{\pi}_4=true$ we can infer that $j\hat{\zeta}_0\nu_2\hat{\pi}_8=true$.

Applying 3.2.8 to $\hat{\pi}_8$ and $\hat{\pi}_7$ (in place of $\hat{\pi}_0$ and $\hat{\pi}_1$) we see that $j\hat{\zeta}_1(\nu_2\wedge\nu_3)\hat{\pi}_7=true$. We originally assumed that $apt\psi\hat{\rho}_3=true$, so $apt\psi(divert\hat{\rho}_6\hat{\rho}_8,\grave{\rho}_8)=true$; thus once we have established that $e[\![E]\!](\nu_2\wedge\nu_3)\delta_7=true$ the knowledge that $torn[\![fnI..E]\!]\psi=true$ and $L[\![E]\!]=true$ will ensure that $c(\mathscr{L}[\![E]\!]\hat{\zeta}_1,\mathscr{L}[\![\cdot[\![E]\!]\psi[false/I]]\!]\grave{\zeta}_1)\hat{\pi}_7=true$ and that $a(\mathscr{L}[\![E]\!]\hat{\zeta}_1\hat{\rho}_7\grave{\upsilon}_7\hat{\sigma}_7,\mathscr{L}[\![\cdot[\![E]\!]\psi[false/I]]\!]\grave{\zeta}_1\grave{\rho}_7\grave{\upsilon}_7\grave{\sigma}_7)=true$.⊁

The stipulations of 3.1.1 are such that all E_0, ν_0, ν_1 and ρ having $g[\![E_0]\!]\nu_0\rho=true$ (as well as $\bigwedge\{free[\![E_0]\!][\![I]\!]\rightarrow(\nu_1\geq\rho[\![I]\!]+1+2\vee\sim\nu_0\geq\rho[\![I]\!]+1+2),true\mid I:Ide\}=true$ and $(free[\![E_0]\!][\![res]\!]\rightarrow(\nu_1\geq\rho[\![res]\!]+1+2\vee\sim\nu_0\geq\rho[\![res]\!]+1+2),true)=true$) satisfies $g[\![E_0]\!]\nu_1\rho=true$; the proof of this, and the comparable result for Δ_0, proceeds by structural induction. In our case we know that $g[\![fnI..E]\!]\nu\hat{\rho}_1=true$, $tear[\![fnI..E]\!]\delta_1=\hat{\rho}_9$, $tidy\hat{\rho}_9(level\nu_2\hat{\pi}_4+1)=true$ and $p_0\hat{\pi}_4=true$ so $g[\![fnI..E]\!]\nu_2\hat{\rho}_1=true$. Hence $e[\![E]\!](\nu_2\wedge\nu_3)\delta_1=true$,

$a\langle \acute{\xi}\acute{\zeta}_0(revert\acute{\rho}_4\acute{\rho}_6)\acute{\upsilon}_6\acute{\sigma}_6, \grave{\xi}(\grave{\zeta}_0\circ revert\grave{\rho}_4)\grave{\rho}_6\grave{\upsilon}_6\grave{\sigma}_6\rangle = true$ and $c\langle \acute{\xi}\acute{\zeta}_0\circ revert\acute{\rho}_4, \grave{\xi}(\grave{\zeta}_0\circ revert\grave{\rho}_4)\rangle \hat{\pi}_5 = true$. This being so for all the appropriate $\hat{\pi}_4$ and $\hat{\pi}_5$, $w\langle \acute{\upsilon}_3\downarrow 1, \grave{\upsilon}_3\downarrow 1\rangle \hat{\pi}_3 = true$ and $p\hat{\pi}_3 = true$.⊁

Hence $a\langle \mathscr{G}[\![\Phi]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, \mathscr{G}[\![g[\![\Phi]\!]\psi]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle = true$ for all ψ, $\hat{\zeta}$, ν, $\hat{\pi}_0$ and $\hat{\pi}_1$ of the form specified above, and we can conclude that $G[\![\Phi]\!] = true$.⊁

3.3.3. Lemma.

If $E[\![E_0]\!] \wedge E[\![E_1]\!] = true$ then $G[\![E_0 := E_1]\!] = true$.

⊲Let ψ, $\hat{\zeta}_0$, ν_0 and $\hat{\pi}_0$ have $apt\psi\hat{\rho}_0 = true$, $torn[\![E_0 := E_1]\!]\psi = true$, $j\hat{\zeta}_0\nu_0\hat{\pi}_0 = true$ and $g[\![E_0 := E_1]\!]\nu_0\acute{\rho}_0 = true$. Define

$\hat{\zeta}_1 = \langle \lambda\rho\upsilon.\acute{\zeta}_0\rho(\langle dummy\rangle \S\upsilon\dagger 2)\circ update(\upsilon\downarrow 2)(\upsilon\downarrow 1),$

$\lambda\rho\upsilon.\grave{\zeta}_0\rho(\langle dummy\rangle \S\upsilon\dagger 2)\circ update(\upsilon\downarrow 2)(\upsilon\downarrow 1)\rangle$,

and suppose that E_0 is evaluated before E_1, so that $\mathscr{G}[\![E_0 := E_1]\!]\acute{\zeta}_0 = \mathscr{L}[\![E_0]\!](\mathscr{R}[\![E_1]\!]\acute{\zeta}_1)$ and likewise $\mathscr{G}[\![g[\![E_0 := E_1]\!]\psi]\!]\grave{\zeta}_0 = \mathscr{L}[\![e[\![E_0]\!]\psi]\!](\mathscr{R}[\![e[\![E_1]\!]\psi]\!]\grave{\zeta}_1)$. Set $\nu_1 = \#\acute{\rho}_0[\![rec]\!] + 1$ and $\nu_2 = \wedge\{I: exit[\![E_0]\!] \rightarrow (ravel\acute{\rho}_0\acute{\rho}_0[\![I]\!]\downarrow 1\downarrow 1 : L \rightarrow \acute{\rho}_0[\![I]\!]\downarrow 1\downarrow 2, \nu_1), \nu_1 \mid I : Ide\}$, for which 3.1.4 dictates that $e[\![E_1]\!]\nu_2\acute{\rho}_0 = true$. Owing to the constraints imposed on E_0, however, we can make even wider claims: if $\hat{\zeta}$ is such that every $\hat{\pi}$ satisfying $p\hat{\pi} \wedge pat\hat{\pi}\nu_1\hat{\pi}_0 = true$, $known\ level(\nu_2 - 1)\hat{\pi}_0)\langle \acute{\upsilon}\downarrow 1, \grave{\upsilon}\downarrow 1\rangle \hat{\pi}_0 = false$ and $\acute{\upsilon}\downarrow 1 : L$ is subject to $a\langle \acute{\zeta}\acute{\rho}\acute{\upsilon}\acute{\sigma}, \grave{\zeta}\grave{\rho}\grave{\upsilon}\grave{\sigma}\rangle = true$ then actually $c\langle \mathscr{L}[\![E_0]\!]\acute{\zeta}, \mathscr{L}[\![e[\![E_0]\!]\psi]\!]\grave{\zeta}\rangle \hat{\pi}_0 = true$. This can be established by induction on the complexity of E_0, taking as the induction hypothesis that for some E and $\hat{\pi}_4$ and all $\hat{\zeta}$ such that $a\langle \acute{\zeta}\acute{\rho}_5\acute{\upsilon}_5\acute{\sigma}_5, \grave{\zeta}\grave{\rho}_5\grave{\upsilon}_5\grave{\sigma}_5\rangle = true$ whenever $p\hat{\pi}_5 \wedge pat\hat{\pi}_5\nu_1\hat{\pi}_4 = true$ and $\acute{\upsilon}_5\downarrow 1 : V$ or $known(level(\nu_2 - 1)\hat{\pi}_4)\langle \acute{\upsilon}_5\downarrow 1, \grave{\upsilon}_5\downarrow 1\rangle \hat{\pi}_4 = false$ we have $c\langle \mathscr{G}[\![E]\!]\acute{\zeta}, \mathscr{G}[\![g[\![E]\!]\psi]\!]\grave{\zeta}\rangle \hat{\pi}_4 = true$.

Suppose that $\hat{\pi}_1$ is any pair having $p\hat{\pi}_1 \wedge pat\hat{\pi}_1\nu_1\hat{\pi}_0 = true$, $known(level\nu_2\hat{\pi}_0)\langle \acute{\upsilon}_1\downarrow 1, \grave{\upsilon}_1\downarrow 1\rangle \hat{\pi}_0 = false$ and $\acute{\upsilon}_1\downarrow 1 : L$. We shall prove that $a\langle \mathscr{R}[\![E_1]\!]\acute{\zeta}_1\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, \mathscr{R}[\![e[\![E_1]\!]\psi]\!]\grave{\zeta}_1\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle = true$ by demonstrating that

$c\langle \mathscr{R}[\![E_1]\!]\acute{\zeta}_1, \mathscr{R}[\![\bullet[\![E_1]\!]\psi]\!]\grave{\zeta}_1\rangle\hat{\pi}_1 = true$. Since $e[\![E_1]\!]\nu_2\acute{\rho}_1 = true$ and $E[\![E_1]\!] = true$ doing so reduces to verifying that $j\langle sv\acute{\zeta}_1, sv\grave{\zeta}_1\rangle\nu_2\hat{\pi}_1 = true$ or that for some typical pair $\hat{\pi}_2$ having $p\hat{\pi}_2 \wedge pat\hat{\pi}_2\nu_2\hat{\pi}_1 = true$ we have $a\langle sv\acute{\zeta}_1\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2, sv\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle = true$. Take one particular $\hat{\pi}_2$ and define $\hat{\varepsilon}_1 = access\langle \acute{\upsilon}_2\downarrow 1, \grave{\upsilon}_2\downarrow 1\rangle\hat{\pi}_2$, $\hat{\varepsilon}_2 = \langle \acute{\upsilon}_2\downarrow 2, \grave{\upsilon}_2\downarrow 2\rangle$ and

$\hat{\pi}_3 = \langle \langle \acute{\rho}_2, \langle dummy\rangle \S\acute{\upsilon}_2\dagger 2, update\acute{\varepsilon}_2\acute{\varepsilon}_1\acute{\sigma}_2\rangle,$

$\langle \grave{\rho}_2, \langle dummy\rangle \S\grave{\upsilon}_2\dagger 2, update\grave{\varepsilon}_2\grave{\varepsilon}_1\grave{\sigma}_2\rangle\rangle$;

once we have shown that $p\hat{\pi}_3 \wedge pat\hat{\pi}_3\nu_0\hat{\pi}_0 = true$ the certainty that $j\hat{\zeta}_0\nu_0\hat{\pi}_0 = true$ will ensure that $a\langle sv\acute{\zeta}_1\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2, sv\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle = true$.

Following the technique of 2.5.5 we can readily confirm that $kent1\hat{\omega}\hat{\pi}_3 \supset (kent1\hat{\omega}\hat{\pi}_2 \vee (\hat{\omega} = \langle dummy, dummy\rangle))$ for all $\hat{\omega}$. Here we wish to establish rather more: that for every $\hat{\omega}$ and ν $known(level\nu\acute{\pi}_0)\hat{\omega}\hat{\pi}_3 \supset known(level\nu\acute{\pi}_1)\hat{\omega}\hat{\pi}_2 \vee (\hat{\omega} = \hat{\varepsilon}_1 \wedge \nu \geq \nu_2)$. Take any ν together with $\acute{\pi}$, which is $level\nu\acute{\pi}_1$, and assume that for some ν_3 and all $\hat{\omega}_0$, $\hat{\omega}_1$ and $\nu_4 \leq \nu_3$ $seen3\nu_4\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3 \wedge known\acute{\pi}\hat{\omega}_1\hat{\pi}_2 \supset known\acute{\pi}\hat{\omega}_0\hat{\pi}_2 \vee (\hat{\omega}_0 = \hat{\varepsilon}_1 \wedge \nu \geq \nu_2)$. Suppose that for some $\hat{\omega}_0$ and $\hat{\omega}_1$ $seen3(\nu_3+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}_3 = true$ and $known\acute{\pi}\hat{\omega}_1\hat{\pi}_2 = true$. If $\grave{\omega}_1$:L but $\grave{\omega}_1$ is not $\grave{\varepsilon}_2$ then $\acute{\omega}_1$ cannot be $\acute{\varepsilon}_2$ as $p_0\hat{\pi}_2 = true$, and so $seen3\nu_3\hat{\omega}_0(access\hat{\omega}_1\hat{\pi}_3)\hat{\pi}_3 = true$ and $known\acute{\pi}(access\hat{\omega}_1\hat{\pi}_3)\hat{\pi}_2 = true$. If $\hat{\omega}_1$:L*×L* then $seen3\nu_3\hat{\omega}_0\hat{\alpha}\hat{\pi}_3 = true$ and $known\acute{\pi}\hat{\alpha}\hat{\pi}_2 = true$ for at least one $\hat{\alpha}$ having $gyven\hat{\alpha}\hat{\omega}_1 = true$. Now note that since $known(level(\nu_2\text{-}1)\acute{\pi}_0)\langle \acute{\upsilon}_1\downarrow 1, \grave{\upsilon}_1\downarrow 1\rangle\hat{\pi}_0 = false$ and $\nu_2\text{-}1 \leq \#\rho_0[\![rec]\!]$ we have $known(level(\nu_2\text{-}1)\acute{\pi}_0)\hat{\varepsilon}_2\hat{\pi}_1 = false$ ($pat\hat{\pi}_1\nu_1\hat{\pi}_0$ being $true$) and $known(level(\nu_2\text{-}1)\acute{\pi}_1)\hat{\varepsilon}_2\hat{\pi}_2 = false$ ($pat\hat{\pi}_2\nu_2\hat{\pi}_1$ being $true$). Hence if $\hat{\omega}_1 = \hat{\varepsilon}_2$ then $\nu \geq \nu_2$ and $seen3\nu_3\hat{\omega}_0\hat{\varepsilon}_1\hat{\pi}_3 = true$; under these circumstances unless $\hat{\omega}_0 = \hat{\varepsilon}_1$ we know that $\hat{\varepsilon}_1$:L*×L* and as $found(level\nu_2\acute{\pi}_1)\hat{\varepsilon}_1\hat{\pi}_2 = true$ there is some $\hat{\alpha}$ with $gyven\hat{\alpha}\hat{\varepsilon}_1 = true$, $seen3(\nu_3\text{-}1)\hat{\omega}_0\hat{\alpha}\hat{\pi}_3 = true$ and $known(level\nu_2\acute{\pi}_1)\hat{\alpha}\hat{\pi}_2 = true$ (indeed as $\nu \geq \nu_2$ $known\acute{\pi}\hat{\alpha}\hat{\pi}_2 = true$). Consequently in all three cases either $\nu \geq \nu_2$ and $\hat{\omega}_0 = \hat{\varepsilon}_1$ or there exist $\hat{\omega}_2$ and ν_4 with $\nu_4 \leq \nu_3$ such that $seen3\nu_4\hat{\omega}_0\hat{\omega}_2\hat{\pi}_3 = true$ and $known\acute{\pi}\hat{\omega}_2\hat{\pi}_2 = true$. Applying the induction

hypothesis, $known\pi\hat{\omega}_0\pi_0\vee(\hat{\omega}_0=\hat{\varepsilon}_1\wedge\nu\geq\nu_2)=true$ and we can conclude that for all $\hat{\omega}_0$, $\hat{\omega}_1$ and ν_5

$seen3\nu_5\hat{\omega}_0\hat{\omega}_1\pi_3\wedge known\pi\hat{\omega}_1\pi_2\supset known\pi\hat{\omega}_0\pi_2\vee(\hat{\omega}_0=\hat{\varepsilon}_1\wedge\nu\geq\nu_2)$. Because $gyven\hat{\omega}\langle pop(level\nu\pi_0\downarrow 2)\upsilon_3, pop(level\nu\pi_0\downarrow 2)\upsilon_3\rangle\supset known\pi\hat{\omega}_1\pi_2$ this is enough to establish that for every $\hat{\omega}_0$

$known(level\nu\pi_0)\hat{\omega}_0\pi_3\supset known\pi\hat{\omega}_0\pi_2\vee(\hat{\omega}_0=\hat{\varepsilon}_1\wedge\nu\geq\nu_2)$; more generally for all $\hat{\omega}$ and ν

$known(level\nu\pi_3)\hat{\omega}\pi_3\supset known(level\nu\pi_3)\hat{\omega}\pi_2\vee((\hat{\omega}=\hat{\delta}_1\vee\hat{\omega}=\langle\upsilon_3\downarrow 1,\upsilon_3\downarrow 1\rangle)\wedge\nu\geq\nu_2)$.

As usual it is plain that then $\hat{\omega}$ satisfies $found(level\nu\pi_2)\hat{\omega}\pi_2=true$ either $found(level\nu\pi_3)\hat{\omega}\pi_3=true$ or $\hat{\omega}:L^*\times L^*$, and thus that when $known(level\nu\pi_3)\hat{\omega}\pi_3=true$ either $found(level\nu\pi_3)\hat{\omega}\pi_3=true$ or $\hat{\omega}:L^*\times L^*$. The fact that $found(level\nu\pi_3)\hat{\varepsilon}_1\pi_2=true$ whenever $\nu\geq\nu_2$ therefore ensures that for all $\hat{\omega}$ and ν $found(level\nu\pi_3)\hat{\omega}\pi_3=true$ if $known(level\nu\pi_3)\hat{\omega}\pi_3=true$. This shows that $p_0\pi_3=true$, and since $kent1\hat{\omega}\pi_3\supset(kent1\hat{\omega}\pi_2\vee(\hat{\omega}=\langle dummy,dummy\rangle))\ p\pi_3=true$ by 3.2.7.

Unless $\nu\geq\nu_1$ $\mathbb{P}q_0(level\nu\pi_n)=\mathbb{P}q_0(level\nu\pi_0)$ when $0\leq n\leq 3$, so for every $\hat{\alpha}$ having $known(level\nu\pi_0)\hat{\alpha}\pi_3=true$ we can deduce successively that $known(level\nu\pi_1)\hat{\alpha}\pi_2=true$, $known(level\nu\pi_0)\hat{\alpha}\pi_1=true$ and $known(level\nu\pi_1)\hat{\alpha}\pi_0=true$ (both $pat\pi_2\nu_2\pi_1$ and $pat\pi_1\nu_1\pi_0$ being $true$). Consequently $pat\pi_3\nu_0\pi_0=true$ and $\alpha\langle\zeta_0\rho_3\upsilon_3\sigma_3,\zeta_0\rho_3\upsilon_3\sigma_3\rangle=true$.

Hence every π_2 having $p\pi_2\wedge pat\pi_2\nu_2\pi_1=true$ satisfies $\alpha\langle sv\zeta_1\rho_2\upsilon_2\sigma_2, sv\zeta_1\rho_2\upsilon_2\sigma_2\rangle=true$, and in fact $j\langle sv\zeta_1, sv\zeta_1\rangle\nu_2\pi_1=true$. This in turn ensures that $\alpha\langle\mathscr{R}[\![E_1]\!]\zeta_1\rho_1\upsilon_1\sigma_1,\mathscr{R}[\![\bullet[\![E_1]\!]\psi]\!]\zeta_1\rho_1\upsilon_1\sigma_1\rangle=true$ for a typical π_1 having $p\pi_1\wedge pat\pi_1\nu_1\pi_0=true$, $known(level\nu_2\pi_0)\langle\upsilon_1\downarrow 1,\upsilon_1\downarrow 1\rangle\pi_0=false$ and $\upsilon_1\downarrow 1:L$. In accordance with our earlier contention this means that $\alpha\langle\mathscr{L}[\![E_0]\!](\mathscr{R}[\![E_1]\!]\zeta_1),\mathscr{L}[\![\bullet[\![E_0]\!]\psi]\!](\mathscr{R}[\![\bullet[\![E_1]\!]\psi]\!]\zeta_1)\rangle\pi_0=true$ and that $G[\![E_0:=E_1]\!]=true$.⊁

3.3.4. Lemma.

If $E[\![E_0]\!] \wedge E[\![E_1]\!] = true$ then $G[\![E_0 E_1]\!] = true$.

⊲As usual we assume that *mete* evaluates expressions from left to right, and we adopt certain ψ, $\hat{\zeta}_0$, ν_0 and $\hat{\pi}_0$ having $apt\psi\hat{\rho}_0 = true$, $torn[\![E_0 E_1]\!]\psi = true$, $j\hat{\zeta}_0\nu_0\hat{\pi}_0 = true$ and $g[\![E_0 E_1]\!]\nu_0\hat{\rho}_0 = true$. We provide

$$\acute{\zeta}_1 = \lambda\rho\upsilon.\upsilon{\downarrow}2 : F \rightarrow (\upsilon{\downarrow}2{\downarrow}1)\acute{\zeta}_0\rho(\langle \upsilon{\downarrow}1\rangle \S\upsilon{\dagger}2),$$
$$sv(\lambda\rho\upsilon.1 \leq \upsilon{\downarrow}1 \,|\, N \leq \#\upsilon{\downarrow}2 \,|\, L^* \rightarrow \acute{\zeta}_0\rho(\langle \upsilon{\downarrow}2{\downarrow}(\upsilon{\downarrow}1)\rangle \S\upsilon{\dagger}2), \tau)\rho\upsilon$$

and

$$\grave{\zeta}_1 = \lambda\rho\upsilon.\upsilon{\downarrow}2 : F \rightarrow (\upsilon{\downarrow}2{\downarrow}1)(\grave{\zeta}_0 \circ revert\rho)(divert\rho(\upsilon{\downarrow}2{\downarrow}2))(\langle \upsilon{\downarrow}1\rangle \S\upsilon{\dagger}2),$$
$$sv(\lambda\rho\upsilon.1 \leq \upsilon{\downarrow}1 \,|\, N \leq \#\upsilon{\downarrow}2 \,|\, L^* \rightarrow \grave{\zeta}_0\rho(\langle \upsilon{\downarrow}2{\downarrow}(\upsilon{\downarrow}1)\rangle \S\upsilon{\dagger}2), \tau)\rho\upsilon,$$

with the effect that $\mathcal{G}[\![E_0 E_1]\!]\acute{\zeta}_0 = \mathcal{R}[\![E_0]\!](\mathcal{L}[\![E_1]\!]\acute{\zeta}_1)$ and $\mathcal{G}[\![f[\![E_0 E_1]\!]\psi]\!]\grave{\zeta}_0 = \mathcal{R}[\![e[\![E_0]\!]\psi]\!](\mathcal{L}[\![e[\![E_1]\!]\psi]\!]\grave{\zeta}_1)$. We shall demonstrate that $j\langle sv(\mathcal{L}[\![E_1]\!]\acute{\zeta}_1), sv(\mathcal{L}[\![e[\![E_1]\!]\psi]\!]\grave{\zeta}_1)\rangle\nu_0\hat{\pi}_0 = true$, which together with $E[\![E_0]\!] = true$ and $e[\![E_0]\!]\nu_0\delta_0 = true$ will serve to establish that $c\langle \mathcal{G}[\![E_0 E_1]\!]\acute{\zeta}_0, \mathcal{G}[\![f[\![E_0 E_1]\!]\psi]\!]\grave{\zeta}_0\rangle\hat{\pi}_0 = true$.

Take any $\hat{\pi}_1$ having $p\hat{\pi}_1 \wedge pat\hat{\pi}_1\nu_0\hat{\pi}_0 = true$ and write $\nu_1 = \#\delta_0[\![rec]\!]$; because $E[\![E_1]\!] = true$ and $e[\![E_1]\!]\nu_1\delta_1 = true$ to show that $a\langle sv(\mathcal{L}[\![E_1]\!]\acute{\zeta}_1)\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, sv(\mathcal{L}[\![e[\![E_1]\!]\psi]\!]\grave{\zeta}_1)\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle = true$ it is enough to prove that $j\langle mv\acute{\zeta}_1, mv\grave{\zeta}_1\rangle\nu_1\hat{\pi}_2 = true$ where $\lambda\zeta.sv\zeta\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1 = \lambda\zeta.\zeta\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2$ and $\lambda\zeta.sv\zeta\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1 = \lambda\zeta.\zeta\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2$. To this end let $\hat{\pi}_3$ be any pair with $p\hat{\pi}_3 \wedge pat\hat{\pi}_3\nu_1\hat{\pi}_2 = true$; since $p\hat{\pi}_2 = true$ by the argument of 2.5.2, the existence of such pairs $\hat{\pi}_3$ is evinced by $\langle\langle\acute{\rho}_2, \langle dummy\rangle \S\acute{\upsilon}_2, \acute{\sigma}_2\rangle, \langle\grave{\rho}_2, \langle dummy\rangle \S\grave{\upsilon}_2, \grave{\sigma}_2\rangle\rangle$. Define $\hat{\pi}_4$ to be the unique pair having $\lambda\zeta.mv\zeta\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3 = \lambda\zeta.\zeta\acute{\rho}_4\acute{\upsilon}_4\acute{\sigma}_4$ and $\lambda\zeta.mv\zeta\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3 = \lambda\zeta.\zeta\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4$, and set $\hat{\varepsilon}_1 = \langle\acute{\upsilon}_4{\downarrow}1, \grave{\upsilon}_4{\downarrow}1\rangle$ and $\hat{\varepsilon}_2 = \langle\acute{\upsilon}_4{\downarrow}2, \grave{\upsilon}_4{\downarrow}2\rangle$. We shall consider the proof further when $\hat{\varepsilon}_2 : F$ only, the other situations being devoid of interest. In addition we shall presume that $p\hat{\pi}_4 \wedge pat\hat{\pi}_4\nu\hat{\pi}_2 = true$, a claim which 2.5.2 readily substantiates.

When $\acute{\varepsilon}_2$:F $\langle \acute{\zeta}_1 \acute{\rho}_4 \acute{\upsilon}_4 \acute{\sigma}_4, \grave{\zeta}_1 \grave{\rho}_4 \grave{\upsilon}_4 \grave{\sigma}_4 \rangle$ elaborates into $\langle (\acute{\varepsilon}_2 \downarrow 1) \acute{\zeta}_0 (revert \acute{\rho}_4 \acute{\rho}_5) \acute{\upsilon}_5 \acute{\sigma}_5, (\grave{\varepsilon}_2 \downarrow 1)(\grave{\zeta}_0 \circ revert \grave{\rho}_4) \grave{\rho}_5 \grave{\upsilon}_5 \grave{\sigma}_5 \rangle$ where $\hat{\pi}_5 = \langle \langle divert \acute{\rho}_4 (\acute{\varepsilon}_2 \downarrow 2), \langle \acute{\varepsilon}_1 \rangle \S \acute{\upsilon}_4 \dagger 2, \acute{\sigma}_4 \rangle, \langle divert \grave{\rho}_4 (\grave{\varepsilon}_2 \downarrow 2), \langle \grave{\varepsilon}_1 \rangle \S \grave{\upsilon}_4 \dagger 2, \grave{\sigma}_4 \rangle \rangle$. We shall show that $p\hat{\pi}_5 = true$ and that if $\hat{\pi}_6 = \langle \langle \acute{\rho}_4, \acute{\upsilon}_4 \dagger 2, \acute{\sigma}_4 \rangle, \langle \grave{\rho}_4, \grave{\upsilon}_4 \dagger 2, \grave{\sigma}_4 \rangle \rangle = true$ $j \hat{\zeta}_0 \nu_0 \hat{\pi}_6 = true$. By 3.2.7 $found\acute{\pi}_4 \hat{\omega} \hat{\pi}_4 = true$ whenever $kent1 \hat{\omega} \hat{\pi}_4 = true$; in particular, if $\hat{\omega}$:J×J and $kent1 \hat{\omega} \hat{\pi}_4 = true$ then $\#\acute{\upsilon}_4 \dagger 2 \geq \#(\acute{\omega} \downarrow 3)$. Consequently we can show by induction on ν_1 that for all $\hat{\omega}_0$, $\hat{\omega}_1$ and ν_2 $seen1 \nu_2 \hat{\omega}_0 \hat{\omega}_1 \hat{\pi}_5 \wedge kent1 \hat{\omega}_1 \hat{\pi}_4 \supset kent1 \hat{\omega}_1 \hat{\pi}_5$. Furthermore because $pat\hat{\pi}_3 \nu_1 \hat{\pi}_2 \wedge pat\hat{\pi}_2 \nu_0 \hat{\pi}_1 = true$ we may assert that $found(level\nu_0 \acute{\pi}_0) \hat{\varepsilon}_2 \hat{\pi}_4 = true$ and that for every $\hat{\omega}_1$ $hoten \hat{\omega}_1 \hat{\rho}_5 \supset kent1 \hat{\omega}_1 \hat{\pi}_4$. Hence $kent1 \hat{\omega}_0 \hat{\pi}_5 \supset kent1 \hat{\omega}_0 \hat{\pi}_4$ for all $\hat{\omega}_0$; by the same token $hoten \hat{\omega}_3 ((\mathfrak{A} q_0 \times \mathfrak{U} q_0) \hat{\rho}_5) \supset hoten \hat{\omega}_3 ((\mathfrak{A} q_0 \times \mathfrak{A} q_0) \hat{\rho}_4)$ for every $\hat{\omega}_3$ and we can infer that for every $\hat{\omega}_2$ $kent1 \hat{\omega}_2 \hat{\pi}_5 \supset kent3((q_0 \times q_0) \hat{\omega}_2)((\mathfrak{P} q_0 \times \mathfrak{P} q_0) \hat{\pi}_4)$. It is apparent from the definition of *found* in 3.2.1 that $found\acute{\pi}_5 \hat{\omega} \hat{\pi}_5 = true$ when $found\acute{\pi}_4 \hat{\omega} \hat{\pi}_4 = true$ unless $\hat{\omega}$:L*×L*, so every pair $\hat{\omega}$ satisfying $kent1 \hat{\omega} \hat{\pi}_5 = true$ is subject to $found\acute{\pi}_5 \hat{\omega} \hat{\pi}_5 = true$.

As $\#(\acute{\varepsilon}_2 \downarrow 2)[\![rec]\!] = 0$ any $\acute{\pi}$ of the form $level\nu\acute{\pi}_5$ is also of the form $level\nu\acute{\pi}_4$ when $\nu \leq \nu_1$. Accordingly we can apply 3.2.2 to such state vectors $\acute{\pi}$, with the outcome that $known\acute{\pi} \hat{\omega} \hat{\pi}_5 \supset known\acute{\pi} \hat{\omega} \hat{\pi}_4$ for all $\hat{\omega}$. Since $found\acute{\pi} \hat{\omega} \hat{\pi}_4 \supset found\acute{\pi} \hat{\omega} \hat{\pi}_5$ and $p_0 \hat{\pi}_4 = true$, for every $\hat{\omega}$ and every appropriate $\acute{\pi}$ $known\acute{\pi} \hat{\omega} \hat{\pi}_5 \supset found\acute{\pi} \hat{\omega} \hat{\pi}_5$. Routine checking now suffices to validate the other clauses of the contention that $p_0 \hat{\pi}_5 = true$. We have already pointed out that $p\hat{\pi}_4 = true$ and that for all $\hat{\omega}$ $kent1 \hat{\omega} \hat{\pi}_5 \supset kent1 \hat{\omega} \hat{\pi}_4$, so by 3.2.7 $p\hat{\pi}_5 = true$.

Given any $\hat{\alpha}$ and $\acute{\pi}$ having $known\acute{\pi} \hat{\alpha} \hat{\pi}_6 = true$ and $\acute{\pi} = level\nu\acute{\pi}_0$ for some $\nu \leq \nu_1$ we can deduce that $known\acute{\pi} \hat{\alpha} \hat{\pi}_4 = true$ and that $known\acute{\pi} \hat{\alpha} \hat{\pi}_3 = true$ by successive applications of 3.2.2. Likewise given any $\hat{\alpha}$ and $\acute{\pi}$ having $known\acute{\pi} \hat{\alpha} \hat{\pi}_2 = true$ and $\acute{\pi} = level\nu\acute{\pi}_2$ for some

$\nu \leq \nu_1$ we know that

$known\pi\hat{\alpha}(\langle \acute{\rho}_1, \langle \acute{\varepsilon}_2 \rangle \S \acute{\upsilon}_1 \dagger 1, \acute{\sigma}_1 \rangle, \langle \grave{\rho}_1, \langle \grave{\varepsilon}_2 \rangle \S \grave{\upsilon}_1 \dagger 1, \grave{\sigma}_1 \rangle) = true$; in fact as $pat\hat{\pi}_1 \nu_0 \hat{\pi}_0 = true$ we can infer from this that $known\pi\hat{\alpha}\hat{\pi}_0 = true$. Hence because $pat\hat{\pi}_3 \nu_1 \hat{\pi}_2 = true$ we can argue that

$$known(level\nu\pi_0)\hat{\alpha}\hat{\pi}_6 \supset known(level\nu\pi_0)\hat{\alpha}\hat{\pi}_3$$
$$\supset known(level\nu\pi_2)\hat{\alpha}\hat{\pi}_3$$
$$\supset known(level\nu\pi_2)\hat{\alpha}\hat{\pi}_2$$
$$\supset known(level\nu\pi_0)\hat{\alpha}\hat{\pi}_0$$

for every $\hat{\alpha}$ and ν, thereby demonstrating that $fit\hat{\pi}_6\hat{\pi}_0 = true$. To establish that $p_1\hat{\pi}_6 = true$ observe that $kent1\hat{\omega}\hat{\pi}_6 \supset kent1\hat{\omega}\hat{\pi}_4$ for all $\hat{\omega}$ and that by 3.2.2 $found(level\nu\pi_6)\hat{\omega}\hat{\pi}_6 \supset (found(level\nu\pi_4)\hat{\omega}\hat{\pi}_4 \vee \hat{\omega}:L^* \times L^*)$ and $known(level\nu\pi_6)\hat{\omega}\hat{\pi}_6 \supset known(level\nu\pi_4)\hat{\omega}\hat{\pi}_4$ for all $\hat{\omega}$ and ν. In consequence $p_1\hat{\pi}_6 \wedge fit\hat{\pi}_6\hat{\pi}_0 = true$ and $j\hat{\zeta}_0\nu_0\hat{\pi}_6 = true$.

Together with $tidy(\acute{\varepsilon}_2\downarrow 2)(level\nu_0\pi_6\downarrow 1) = true$ and $w\hat{\varepsilon}_2\hat{\pi}_3 = true$ the facts that $p\hat{\pi}_5 = true$ and $j\hat{\zeta}_0\nu_0\hat{\pi}_6 = true$ entail $a\langle \acute{\zeta}_1\acute{\rho}_4\acute{\upsilon}_4\acute{\sigma}_4, \grave{\zeta}_1\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4 \rangle = true$. In view of the definition of $\hat{\pi}_4$ this means that for every $\hat{\pi}_0$ having $p\hat{\pi}_3 \wedge pat\hat{\pi}_3\nu_1\hat{\pi}_2 = true$ $a\langle mv\acute{\zeta}_1\acute{\rho}_3\acute{\upsilon}_3\acute{\sigma}_3, mv\grave{\zeta}_1\grave{\rho}_3\grave{\upsilon}_3\grave{\sigma}_3 \rangle = true$. Hence $j\langle mv\acute{\zeta}_1, mv\grave{\zeta}_1 \rangle \nu_1\hat{\pi}_2 = true$ and $j\langle \mathscr{L}[\![E_1]\!]\acute{\zeta}_1, \mathscr{L}[\![e[\![E_1]\!]\psi]\!]\grave{\zeta}_1 \rangle \hat{\pi}_2 = true$, which ensures that for all pairs $\hat{\pi}_1$ with $p\hat{\pi}_1 \wedge pat\hat{\pi}_1\nu_0\hat{\pi}_0 = true$ we have $a\langle sv(\mathscr{L}[\![E_1]\!]\acute{\zeta}_1)\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1, sv(\mathscr{L}[\![e[\![E_1]\!]\psi]\!]\grave{\zeta}_1)\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1 \rangle = true$. Finally $c\langle \mathscr{E}[\![E_0E_1]\!]\acute{\zeta}_0, \mathscr{E}[\![g[\![E_0E_1]\!]\psi]\!]\grave{\zeta}_0 \rangle \hat{\pi}_0 = true$ for every appropriate ψ, $\hat{\zeta}_0$, ν_0 and $\hat{\pi}_0$, so $G[\![E_0E_1]\!] = true$.⊳

3.3.5. Lemma.

If $E[\![E]\!] = true$ then $G[\![val\ E]\!] \wedge G[\![res\ E]\!] \wedge G[\![goto\ E]\!] = true$.

⊲For the reasons enunciated in 2.6.4 we shall content ourselves with proving that $G[\![res\ E]\!] = true$. Let ψ, $\hat{\zeta}_0$, ν_0 and $\hat{\pi}_0$ satisfy $apt\psi\hat{\rho}_0 = true$, $rent[\![E]\!]\psi = true$, $j\hat{\zeta}_0\nu\hat{\pi}_0 = true$ and $g[\![res\ E]\!]\nu\acute{\rho}_0 = true$, and set

$\hat{\zeta}_0=\langle \lambda\rho.(ravel\rho\rho[\![res]\!]\downarrow 1\downarrow 1\downarrow 1)\rho,$

$\quad \lambda\rho\upsilon.(\rho[\![res]\!]\downarrow 1\downarrow 1)(\rho[\![res]\!]\downarrow 1\downarrow 2)(\langle \upsilon\downarrow 1\rangle \S\rho[\![res]\!]\downarrow 1\downarrow 3)\rangle$.

Then $\langle \mathscr{G}[\![res\ E]\!]\acute{\zeta},\mathscr{G}[\![g[\![res\ E]\!]\psi]\!]\grave{\zeta}\rangle=\langle \mathscr{L}[\![E]\!]\acute{\zeta}_0,\mathscr{L}[\![e[\![E]\!]\psi]\!]\grave{\zeta}_0\rangle$, $L[\![E]\!]=true$ and $e[\![E]\!]0\acute{\rho}_0=true$, so to demonstrate that $e\langle \mathscr{G}[\![res\ E]\!]\acute{\zeta},\mathscr{G}[\![g[\![res\ E]\!]\psi]\!]\grave{\zeta}\rangle\hat{\pi}_0=true$ we need only confirm that $j\hat{\zeta}_0 0\hat{\pi}_0=true$.

Suppose that $\hat{\pi}_1$ is any pair having $p\hat{\pi}_1\wedge pat\hat{\pi}_1 0\hat{\pi}_0=true$ and that $\#\rho_1[\![res]\!]>0$, the result being trivial otherwise. Define $\hat{\varepsilon}_0=\langle \upsilon_1\downarrow 1,\upsilon_1\downarrow 1\rangle$, $\hat{\varepsilon}_1=access\hat{\varepsilon}_0\hat{\pi}_1$, $\nu_7=ravel\acute{\rho}_1\acute{\rho}_1[\![res]\!]\downarrow 1\downarrow 1\downarrow 2$, $\hat{\pi}_7=level\nu_7\hat{\pi}_1$ and

$\hat{\pi}_2=\langle\langle \acute{\rho}_7,\langle \acute{\varepsilon}_0\rangle \S\acute{\upsilon}_7,restore(\acute{\varepsilon}_0:L\rightarrow update\acute{\varepsilon}_0\acute{\varepsilon}_1\acute{\sigma}_7,\acute{\sigma}_7)\acute{\sigma}_1\rangle$

$\quad \langle \grave{\rho}_1[\![res]\!]\downarrow 1\downarrow 2,\langle \grave{\varepsilon}_0\rangle \S\grave{\rho}_1[\![res]\!]\downarrow 1\downarrow 3,\grave{\sigma}_1\rangle\rangle$.

As $kent1\langle ravel\acute{\rho}_1\acute{\rho}_1[\![res]\!]\downarrow 1\downarrow 1,\grave{\rho}_1[\![res]\!]\downarrow 1\rangle\hat{\pi}_1=true$, we can write $\langle \acute{\zeta}_0\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=\langle \acute{\zeta}_1\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle$ where $\hat{\zeta}_1=\langle ravel\acute{\rho}_1\acute{\rho}_1[\![res]\!]\downarrow 1\downarrow 1\downarrow 1,\grave{\rho}_1[\![res]\!]\downarrow 1\rangle$. In order to establish that $j\hat{\zeta}_0 0\hat{\pi}_0=true$ we must simply show that for this typical pair $\hat{\pi}_1$ $a\langle \acute{\zeta}_0\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$; this goal will be attained by verifying that $p\hat{\pi}_2$, $pat\hat{\pi}_2 0\hat{\pi}_3$ and $j\hat{\zeta}_1 0\hat{\pi}_3$ are all $true$ when $\hat{\pi}_3=\langle\langle \acute{\rho}_2,\acute{\upsilon}_2\dagger 1,\acute{\sigma}_2\rangle,\langle \grave{\rho}_2,\grave{\upsilon}_2\dagger 1,\grave{\sigma}_2\rangle\rangle$. For brevity we introduce $\hat{\pi}_{n+3}=((\mathbb{P}q_0\times\mathbb{P}q_0)\hat{\pi}_n)$ when $1\leq n\leq 3$ and $\hat{\varepsilon}_{n+2}=(q_0\times q_0)\hat{\varepsilon}_n$ when $0\leq n\leq 1$, so that $\grave{\rho}_6=\grave{\rho}_5=revert\grave{\rho}_5\grave{\rho}_4$ and $\grave{\upsilon}_5=\grave{\upsilon}_4$.

As $\mathbb{P}q_0\hat{\pi}_7=level\nu_7\hat{\pi}_4$ and $p_0\hat{\pi}_4=true$ any $\hat{\omega}$ subject to $known\hat{\pi}_7\hat{\omega}\hat{\pi}_4=true$ obeys $found\hat{\pi}_7\hat{\omega}\hat{\pi}_4=true$; in particular, when $\hat{\omega}:L$ $area\hat{\omega}\acute{\sigma}_6=true$ and $access\hat{\omega}\hat{\pi}_1=access\hat{\omega}\hat{\pi}_6$. Moreover the clauses of $p_0\hat{\pi}_1$ stipulate that $\#\acute{\rho}_1[\![rec]\!]=(\nu_7-1)\vee 0$, so any $\hat{\pi}_8$ and ν_8 such that $\hat{\pi}_8=level\nu_8\hat{\pi}_6$ satisfy either $\hat{\pi}_8=level\nu_8\hat{\pi}_4$ and $\nu_8\leq(\nu_7-1)\vee 0$ or $\hat{\pi}_8=\hat{\pi}_6$ and $\nu_8\geq\nu_7$; under both possibilities $known\hat{\pi}_8=known(level(\nu_8\wedge\nu_7)\hat{\pi}_4)$. Consequently we can apply 3.2.2 to $\hat{\pi}_4$, $\hat{\pi}_6$ and $\hat{\pi}_8$, with the outcome that for all $\hat{\omega}$ and ν_8 $known(level\nu_8\hat{\pi}_6)\hat{\omega}\hat{\pi}_6=true$ if and only if $known(level(\nu_8\wedge\nu_7)\hat{\pi}_4)\hat{\omega}\hat{\pi}_4=true$. Obviously 3.2.2 is also relevant to $\hat{\pi}_5$, $\hat{\pi}_6$ and $\hat{\pi}_8$, so that for all $\hat{\omega}$ and ν_8 $known(level\nu_8\hat{\pi}_6)\hat{\omega}\hat{\pi}_6=true$

if and only if $known(level\nu_8\Pi_6)\hat{\omega}\Pi_5=true$.

Since $pat\Pi_1 0\Pi_0=true$, $found(level 0\Pi_1)\hat{\varepsilon}_1\Pi_1=true$ and for all $\hat{\omega}_0$ and ν_1 $seen1\nu_1\hat{\omega}_0\hat{\varepsilon}_2\Pi_6\supset(known(level 0\Pi_6)\hat{\omega}_0\Pi_5\vee\hat{\omega}_0=\hat{\varepsilon}_2\vee\hat{\omega}_0=\hat{\varepsilon}_3)$; hence for every $\hat{\omega}$ $known\Pi_5\hat{\omega}\Pi_5\supset(known\Pi_6\hat{\omega}\Pi_5\vee\hat{\omega}=\hat{\varepsilon}_2\vee\hat{\omega}=\hat{\varepsilon}_3)$. Combining this with the above we see that for every $\hat{\omega}$ and ν_8 $known(level\nu_8\Pi_5)\hat{\omega}\Pi_5=true$ only if $known(level(\nu_8\wedge\nu_7)\Pi_4)\hat{\omega}\Pi_4\vee((\hat{\omega}=\hat{\varepsilon}_2\vee\hat{\omega}=\hat{\varepsilon}_3)\wedge\nu_8\geq\nu_7)=true$. Thus $known(level\nu_8\Pi_5)\hat{\omega}\Pi_5=true$ only if $found(level\nu_8\Pi_5)\hat{\omega}\Pi_5=true$, because $p_0\Pi_4=true$; analogous remarks are valid for Π_6, and indeed we even know that $known(level\nu_8\Pi_n)\hat{\omega}\Pi_n=true$ only if $found(level\nu_8\Pi_n)\hat{\omega}\Pi_n=true$ when $2\leq n\leq 3$.

Select any I_0, I_1, ν_0 and ν_1 such that when $0\leq n\leq 1$ $\rho_2[\![I_n]\!]\downarrow\nu_n:L$. That $\mathfrak{M}q_0\grave{\rho}_1[\![res]\!]\downarrow 1\downarrow 2=revert(\grave{\rho}_1[\![res]\!]\downarrow 1\downarrow 2)(\mathfrak{M}q_0\grave{\rho}_1)$ is reflected in the certainty that $\grave{\rho}_2[\![I_n]\!]\downarrow\nu_n=\grave{\rho}_1[\![I_n]\!]\downarrow\nu_{n+2}$ where $\nu_{n+2}=\#\grave{\rho}_1[\![I_n]\!]-\#\grave{\rho}_2[\![I_n]\!]+\nu_n$; in addition $\acute{\rho}_2[\![I_n]\!]\downarrow\nu_n=\acute{\rho}_1[\![I_n]\!]\downarrow\nu_{n+2}$ and $ravel\acute{\rho}_2\acute{\rho}_2[\![I_n]\!]\downarrow\nu_n\downarrow 1=ravel\acute{\rho}_1\acute{\rho}_1[\![I_n]\!]\downarrow\nu_{n+2}\downarrow 1$ because $lead\nu_{n+2}(\acute{\rho}_1[\![I_n]\!])=\#\grave{\rho}_1[\![I_n]\!]-\#\grave{\rho}_2[\![I_n]\!]+lead\nu_n(\grave{\rho}_2[\![I_n]\!])$ when $0\leq n\leq 1$. Accordingly if $\grave{\rho}_2[\![I_0]\!]\downarrow\nu_0=\grave{\rho}_2[\![I_1]\!]\downarrow\nu_1$ then we can deduce in succession that $\grave{\rho}_1[\![I_0]\!]\downarrow\nu_2=\grave{\rho}_1[\![I_1]\!]\downarrow\nu_3$, that $I_0=I_1$ and $\acute{\rho}_1[\![I_0]\!]\downarrow\nu_2\dagger 1=\acute{\rho}_1[\![I_1]\!]\downarrow\nu_3\dagger 1$ (as $p_0\Pi_1=true$), and that $\acute{\rho}_2[\![I_0]\!]\downarrow\nu_0\dagger 1=\rho_2[\![I_1]\!]\downarrow\nu_1\downarrow 1$. This completes those parts of the proof that $p_0\Pi_2\wedge p_0\Pi_3=true$ which are worth giving.

By induction on ν_1 we can readily ratify the statement that when $2\leq n\leq 3$ for all $\hat{\omega}_0$, $\hat{\omega}_1$ and ν_1 $seen1\nu_1\hat{\omega}_0\hat{\omega}_1\Pi_n\wedge kent1\hat{\omega}_1\Pi_1\wedge known\Pi_7((q_0\times q_0)\hat{\omega}_1)\Pi_4\supset kent1\hat{\omega}_0\Pi_1$; as a result $kent1\hat{\omega}\Pi_2\supset kent1\hat{\omega}\Pi_1$ and $kent1\hat{\omega}\Pi_3\supset kent1\hat{\omega}\Pi_1$ for all $\hat{\omega}$. From 3.2.7 and the property $p_0\Pi_2\wedge p_0\Pi_3=true$ it now follows that $p\Pi_2\wedge p\Pi_3=true$. Moreover for every $\hat{\alpha}$ and ν_8 $known(level\nu_8\Pi_3)\hat{\alpha}\Pi_2=true$ only if $known(level\nu_8\Pi_3)\hat{\alpha}\Pi_3=true$, so $pat\Pi_2 0\Pi_3=true$. That $j\hat{\zeta}_1 0\Pi_3=true$ is now a direct consequence of the knowledge that $w\langle ravel\acute{\rho}_1\acute{\rho}_1[\![res]\!]\downarrow 1\downarrow 1,\grave{\rho}_1[\![res]\!]\downarrow 1\rangle\Pi_1=true$.

Tracing back through the steps of the argument, $\alpha\langle\acute{\zeta}_1\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2,\grave{\zeta}_1\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2\rangle=true$, $j\hat{\zeta}_0 0\Pi_0=true$ and

$c\langle\mathscr{G}[\![\text{res } E]\!]\zeta', \mathscr{G}[\![g[\![\text{res } E]\!]\psi]\!]\zeta''\rangle\pi_0'' = true$. This being so for all the apposite ψ, $\hat{\zeta}$, ν and π_0'' we can conclude that $G[\![\text{res } E]\!] = true$.⊁

We cannot weaken the condition $e[\![\text{res } E]\!] = \lambda\nu\rho.e[\![E]\!]0\rho$ to $e[\![\text{res } E]\!] = \lambda\nu\rho.e[\![E]\!](\rho[\![\text{res}]\!]\downarrow 1\downarrow 2)\rho$ because this would admit such programs as l: val (res l), in which control leaves a block immediately after executing a res statement. Although less confining constraints on res statements could be provided by using $[\text{Ide}\mapsto[D\times N\times N]^*]\times[N\times N\times N]^*\times[P\times N\times N]^*$ instead of U in 3.1.4, the outcome would be too meagre to pay for the effort involved.

3.3.6. Lemma.

If $L[\![E]\!] \wedge D[\![\Delta]\!] = true$ then $G[\![\Delta \text{ inside } E]\!] = true$.

⊲Suppose that ψ_0, $\hat{\zeta}_0$, ν_0 and π_0'' satisfy $apt\psi_0\hat{\rho}_0 = true$, $torn[\![\Delta \text{ inside } E]\!]\psi = true$, $j\hat{\zeta}_0\nu_0\pi_0'' = true$ and $g[\![\Delta \text{ inside } E]\!]\nu_0\rho_0' = true$; we shall show that $c\langle\mathscr{G}[\![\Delta \text{ inside } E]\!]\zeta_0', \mathscr{G}[\![g[\![\Delta \text{ inside } E]\!]\psi]\!]\zeta_0''\rangle\pi_0'' = true$. To this end we define $\psi_1 = \psi_0[false^*/\mathscr{I}[\![\Delta]\!]][opts(\mathscr{K}[\![\Delta]\!])\psi/\mathscr{K}[\![\Delta]\!]]$, $\nu_1 = \nu_0\wedge(\#\rho_0'[\![\text{rec}]\!]+1)$ and $\hat{\zeta}_1 = \langle remit\zeta_0', \zeta_0''\circ revert\rho_0''\rangle$, and given any π_1'' having $p\pi_1''\wedge fit\pi_1''\pi_0'' = true$ we set $\pi_2'' = \langle\langle\rho_1'[\pi_1''//\text{rec}], \upsilon_1', \sigma_1'\rangle, \pi_1''\rangle$. Now $\mathscr{G}[\![\Delta \text{ inside } E]\!]\zeta_0'\rho_1'\upsilon_1'\sigma_1' = \mathscr{D}[\![\Delta]\!](\mathscr{L}[\![E]\!]\zeta_1')\rho_2'\upsilon_2'\sigma_2'$ while $\mathscr{G}[\![g[\![\Delta \text{ inside } E]\!]\psi_0]\!]\zeta_0''\rho_1''\upsilon_1''\sigma_1'' = \mathscr{A}[\![d[\![\Delta]\!]\psi_0]\!](\mathscr{L}[\![e[\![E]\!]\psi_1]\!]\zeta_1'')\rho_2''\upsilon_2''\sigma_2''$.

Patently $p\pi_2''\wedge fit\pi_2''\pi_2'' = true$, so as $D[\![\Delta]\!] = true$ and $d[\![\Delta]\!]0\rho_2' = true$ to show that $c\langle\mathscr{G}[\![\Delta \text{ inside } E]\!]\zeta_0'\rho_1'\upsilon_1'\sigma_1', \mathscr{G}[\![g[\![\Delta \text{ inside } E]\!]\psi_0]\!]\zeta_0\rho_1''\upsilon_1''\sigma_1''\rangle = true$ we need only demonstrate that $c\langle\mathscr{L}[\![E]\!]\zeta_1', \mathscr{L}[\![e[\![E]\!]\psi_1]\!]\zeta_1''\rangle\pi = true$ for all π having $spun[\![\Delta]\!]0\psi_0\pi_2''\pi = true$. There certainly are such π, for if $\langle\alpha^*, \alpha''^*\rangle = \langle news(\#\mathscr{I}[\![\Delta]\!])\sigma_2', news(\#\mathscr{I}[\![\Delta]\!])\sigma_2''\rangle$ we may take π to be $\langle\rho_2'[\alpha^*//\mathscr{I}[\![\Delta]\!]][dummy^*/\mathscr{K}[\![\Delta]\!]], \upsilon_2', updates\alpha^*dummy^*\sigma_2''\rangle$, dealing similarly with π''. Select one particular π_3 having $spun[\![\Delta]\!]0\psi_0\pi_2''\pi_3 = true$; because $g[\![\Delta \text{ inside } E]\!]\nu_0\rho_0' = true$ 3.1.4 makes it evident that $e[\![E]\!]\nu_1\rho_3 = true$. In addition we know that $L[\![E]\!] = true$,

so $c\langle\mathscr{L}[\![\mathrm{E}]\!]\acute{\zeta}_1,\mathscr{L}[\![\mathrm{e}\ [\![\mathrm{E}]\!]\psi_1]\!]\grave{\zeta}_1\rangle\hat{\pi}_3$ will be *true* if $j\hat{\zeta}_1\nu_1\hat{\pi}_3$ is *true*.

As $p_1\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$, $j\hat{\zeta}_0\nu_0\hat{\pi}_1=true$; in addition $\#\acute{\rho}_3[\![\mathrm{rec}]\!]=\#\acute{\rho}_1[\![\mathrm{rec}]\!]+1$, $\mathbb{q}_0\acute{\rho}_3[\![\mathrm{rec}]\!]\downarrow 1\downarrow 1=\mathbb{P}q_0\hat{\pi}_1$ and $p_1\hat{\pi}_3=true$ from the definition of *spun* in 3.2.4. Writing $\hat{\pi}_4=\langle\langle revert\acute{\rho}_1\acute{\rho}_3,\acute{\upsilon}_3,\acute{\sigma}_3\rangle,\langle revert\grave{\rho}_1\grave{\rho}_3,\grave{\upsilon}_3,\grave{\sigma}_3\rangle\rangle$, once we have proved that $known(level\nu\hat{\pi}_1)\hat{\alpha}\hat{\pi}_4\supset known(level\nu\hat{\pi}_1)\hat{\alpha}\hat{\pi}_1$ for all $\hat{\alpha}$ and ν 3.2.8 will assure us that $j\hat{\zeta}_1\nu_1\hat{\pi}_3=true$.

Let $\hat{\pi}$ be of the form $level\nu\hat{\pi}_1$ for some ν, so that $\mathbb{P}q_0\hat{\pi}=\mathbb{P}q_0(level(\nu\wedge\nu_1)\hat{\pi}_2)$. By 3.2.2 for all $\hat{\omega}$ $known\hat{\pi}\hat{\omega}\hat{\pi}_4=true$ if and only if $known\hat{\pi}\hat{\omega}\hat{\pi}_3=true$. In particular any $\hat{\alpha}$ satisfying $known\hat{\pi}\hat{\alpha}\hat{\pi}_4=true$ is subject to $known\hat{\pi}\hat{\alpha}\hat{\pi}_3=true$ also; hence as $spun[\![\Delta]\!]0\psi_0\hat{\pi}_2\hat{\pi}_3=true$ $known\hat{\pi}\hat{\alpha}\hat{\pi}_2=true$ and $known\hat{\pi}\hat{\alpha}\hat{\pi}_1=true$, giving $j\hat{\zeta}_1\nu_1\hat{\pi}_3=true$. This establishes that $a\langle\mathscr{G}[\![\Delta\ \mathrm{inside}\ \mathrm{E}]\!]\acute{\zeta}_0\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{G}[\![\mathfrak{g}[\![\Delta\ \mathrm{inside}\ \mathrm{E}]\!]\psi_0]\!]\grave{\zeta}_0\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$ and that $c\langle\mathscr{G}[\![\Delta\ \mathrm{inside}\ \mathrm{E}]\!]\acute{\zeta}_0,\mathscr{G}[\![\mathfrak{g}[\![\Delta\ \mathrm{inside}\ \mathrm{E}]\!]\psi_0]\!]\grave{\zeta}_0\rangle\hat{\pi}_0=true$. As ψ_0, $\hat{\zeta}_0$, ν_0 and $\hat{\pi}_0$ can be chosen at will we must have $G[\![\Delta\ \mathrm{inside}\ \mathrm{E}]\!]=true$.⊁

The proofs needed for those expressions having labels with propagated scopes resemble such calculations as 2.5.7 too closely to deserve attention.

3.3.7. Lemma.

Let Δ be $\mathrm{I}=\mathrm{E}$, $\mathrm{I}==\mathrm{E}$, $\mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}$ or $\mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}$ for some I or $\mathrm{I}_1,\ldots,\mathrm{I}_n$ and some E such that $E[\![\mathrm{E}]\!]=true$; then $D[\![\Delta]\!]\wedge T[\![\Delta]\!]=true$.

⊀We first outline the proof that $D[\![\Delta]\!]=true$ when Δ is of the form $\mathrm{I}==\mathrm{E}$ and when $E[\![\mathrm{E}]\!]=true$. Let ψ_0, $\hat{\zeta}_0$ and $\hat{\pi}_0$ be such that $d[\![\Delta]\!]0\acute{\rho}_0=true$ and $\bigwedge\{c\hat{\zeta}_0\hat{\pi}\mid spun[\![\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}\}=true$; taking $\hat{\zeta}_0$ to be $\langle\perp,\perp\rangle$ confirms that such entities exist. When $\nu_0=\#\acute{\rho}_0[\![\mathrm{rec}]\!]+1$ and

$\hat{\zeta}_1=\langle sv(\lambda\rho\upsilon.\acute{\zeta}_0\rho[\upsilon\downarrow 1//\mathrm{I}](\upsilon\dagger 1)),$

$(\lambda\zeta.opt[\![\mathrm{I}]\!]\psi_0\rightarrow sv(mv\zeta),sv\zeta)(\lambda\rho\upsilon.\grave{\zeta}_0\rho[\upsilon\downarrow 1/\mathrm{I}](\upsilon\dagger 1))\rangle$

we know that $e[\![E]\!] \nu_0 \acute{\rho}_0 = true$ and that $\langle \mathscr{D}[\![\Delta]\!] \acute{\zeta}_0, \mathscr{D}[\![\mathscr{d} [\![\Delta]\!] \psi_0]\!] \grave{\zeta}_0 \rangle = \langle \mathscr{E}[\![E]\!] \acute{\zeta}_1, \mathscr{E}[\![\mathscr{e} [\![E]\!] \psi_0]\!] \grave{\zeta}_1 \rangle$. Since $E[\![E]\!] = true$ the conclusion $c\langle \mathscr{D}[\![\Delta]\!] \acute{\zeta}_0, \mathscr{D}[\![\mathscr{d} [\![\Delta]\!] \psi_0]\!] \grave{\zeta}_0 \rangle \hat{\pi}_0 = true$ will follow immediately once we have established that $j \hat{\zeta}_1 \nu_0 \hat{\pi}_0 = true$. Suppose that $\hat{\pi}_1$ is any pair having $p \hat{\pi}_1 \wedge pat \hat{\pi}_1 \nu_0 \hat{\pi}_0 = true$, and introduce

$\hat{\varepsilon}_0 = access \langle \acute{\upsilon}_1 \downarrow 1, \grave{\upsilon}_1 \downarrow 1 \rangle \hat{\pi}_1, \hat{\varepsilon}_1 = \langle \acute{\varepsilon}_1, (opt[\![I]\!] \psi_0 \rightarrow new \grave{\sigma}_1, \grave{\varepsilon}_0) \rangle$ and

$\hat{\pi}_2 = \langle \langle \acute{\rho}_1 [\acute{\varepsilon}_1 /// I], \acute{\upsilon}_1 \dagger 1, \acute{\sigma}_1 \rangle$,

$\langle \grave{\rho}_1 [\grave{\varepsilon}_1 / I], \grave{\upsilon}_1 \dagger 1, (opt[\![I]\!] \psi_0 \rightarrow update \grave{\varepsilon}_1 \grave{\varepsilon}_0 \grave{\sigma}_1, \grave{\sigma}_1) \rangle \rangle$.

By the method of 3.3.2 we can readily show that $p \hat{\pi}_2 = true$; note in particular that as $\grave{\varepsilon}_1 = new \grave{\sigma}_1$ (when $\grave{\varepsilon}_1$ is a location) and as $p_0 \hat{\pi}_1 = true$ any I_6 and ν_6 having $\grave{\rho}_2[\![I]\!] \downarrow 1 = \grave{\rho}_2[\![I_6]\!] \downarrow \nu_6$ must have $I = I_6$ and $\nu = \nu_6$. We can also show quite easily that $spun[\![\Delta]\!] 0 \psi_0 \hat{\pi}_0 \hat{\pi}_2 = true$, so $a \langle \acute{\zeta}_0 \acute{\rho}_2 \acute{\upsilon}_2 \acute{\sigma}_2, \grave{\zeta}_0 \grave{\rho}_2 \grave{\upsilon}_2 \grave{\sigma}_2 \rangle = true$ and $j \hat{\zeta}_1 \nu_0 \hat{\pi}_0 = true$. Hence $c \langle \mathscr{D}[\![\Delta]\!] \acute{\zeta}_0, \mathscr{D}[\![\mathscr{d} [\![\Delta]\!] \psi_0]\!] \grave{\zeta}_0 \rangle \hat{\pi}_0 = true$ for any suitable ψ_0, $\hat{\zeta}_0$ and $\hat{\pi}_0$, thereby ensuring that $D[\![\Delta]\!] = true$.

⊲We now turn to the proof that $T[\![\Delta]\!] = true$ when Δ is a declaration of the form I==E for which $E[\![E]\!] = true$. Let ψ_3, $\hat{\zeta}_3$ and $\hat{\pi}_3$ be such that $t[\![\Delta]\!] 0 \acute{\rho}_3 = true$ and $\bigwedge \{ c \hat{\zeta}_3 \hat{\pi} \mid spun[\![\Delta]\!] 1 \psi_3 \hat{\pi}_3 \hat{\pi} \} = true$; then $\psi_3[\![I]\!] \downarrow 1 = true$, and writing $\nu_3 = \# \acute{\rho}_3 [\![rec]\!] + 1$ and

$\hat{\zeta}_4 = \langle sv(\lambda \rho \upsilon . \acute{\zeta}_3 \rho [\upsilon \downarrow 1 /// I](\upsilon \dagger 1)),$

$sv(\lambda \rho \upsilon . \rho[\![I]\!] \downarrow 1 : L \rightarrow \grave{\zeta}_3 \rho (\upsilon \dagger 1) \circ update(\rho[\![I]\!] \downarrow 1)(\upsilon \downarrow 1), \tau) \rangle$

gives $e[\![E]\!] \nu_3 \acute{\rho}_3 = true$ and $\langle \mathscr{T}[\![\Delta]\!] \acute{\zeta}_3, \mathscr{T}[\![\mathscr{t} [\![\Delta]\!] \psi_4]\!] \grave{\zeta}_3 \rangle = \langle \mathscr{E}[\![E]\!] \acute{\zeta}_4, \mathscr{E}[\![\mathscr{e} [\![E]\!] \psi_3]\!] \grave{\zeta}_4 \rangle$. As $E[\![E]\!] = true$ to show that $c \langle \mathscr{T}[\![\Delta]\!] \acute{\zeta}_3, \mathscr{T}[\![\mathscr{t} [\![\Delta]\!] \psi_3]\!] \grave{\zeta}_3 \rangle \hat{\pi}_3 = true$ it suffices to verify that $j \hat{\zeta}_4 \nu_3 \hat{\pi}_3 = true$. Accordingly take any $\hat{\pi}_4$ having $p \hat{\pi}_4 \wedge pat \hat{\pi}_4 \nu_3 \hat{\pi}_3 = true$, and set $\hat{\varepsilon}_3 = access \langle \acute{\upsilon}_4 \downarrow 1, \grave{\upsilon}_4 \downarrow 1 \rangle \hat{\pi}_4$ and

$\hat{\pi}_5 = \langle \langle \acute{\rho}_4 [\acute{\varepsilon}_3 /// I], \acute{\upsilon}_4 \dagger 1, \acute{\sigma}_4 \rangle, \langle \grave{\rho}_4, \grave{\upsilon}_4 \dagger 1, update(\grave{\rho}_4[\![I]\!] \downarrow 1) \grave{\varepsilon}_3 \grave{\sigma}_4 \rangle \rangle$.

⊲There is some $\hat{\pi}$ for which $spun[\![\Delta]\!] 1 \psi_3 \hat{\pi}_3 \hat{\pi} = true$, so $\grave{\rho}_3[\![I]\!] \downarrow 1 : L$, $\grave{\rho}_4[\![I]\!] \downarrow 1 : L$ and $\langle \acute{\zeta}_4 \acute{\rho}_4 \acute{\upsilon}_4 \acute{\sigma}_4, \grave{\zeta}_4 \grave{\rho}_4 \grave{\upsilon}_4 \grave{\sigma}_4 \rangle = \langle \acute{\zeta}_3 \acute{\rho}_5 \acute{\upsilon}_5 \acute{\sigma}_5, \grave{\zeta}_3 \grave{\rho}_5 \grave{\upsilon}_5 \grave{\sigma}_5 \rangle$. Likewise $ravel \acute{\rho}_3 \acute{\rho}_3 [\![I]\!] \downarrow 1 \downarrow 1 : V$ and $ravel \acute{\rho}_4 \acute{\rho}_4 [\![I]\!] \downarrow 1 \downarrow 1 : V$, so, $p_0 \hat{\pi}_4$ being $true$, we can presume that $\grave{\rho}_4[\![I]\!] \downarrow 1 = \grave{\rho}_4[\![I_6]\!] \downarrow \nu_6$ only if $I = I_6$ and $lead1(\acute{\rho}_4[\![I]\!]) = lead \nu_6 (\acute{\rho}_4[\![I]\!])$. The description of $\acute{\rho}_4 [\acute{\varepsilon}_3 /// I]$ given

in 3.1.4 therefore ensures that for all ρ and $\hat{\omega}$

$hoten\hat{\omega}(revert\rho\rho_5, revert\rho\grave{\rho}_5) = (hoten\hat{\omega}(revert\rho\rho_4, revert\rho\grave{\rho}_4) \wedge \sim\grave{\omega}=\grave{\delta}_3)$
$\vee(\hat{\omega}=\hat{\delta}_3 \wedge (lead1(\rho_3[\![I]\!]) > \#\rho_3[\![I]\!] - \#\rho[\![I]\!]))$

where $\hat{\delta}_3 = (\hat{\varepsilon}_3, \grave{\rho}_4[\![I]\!]\downarrow 1)$. The fact that $spun[\![\Delta]\!]1\psi_3\hat{\pi}_3\hat{\pi}=true$ for some $\hat{\pi}$ also implies that $\rho_3[\![I]\!]\downarrow 1\downarrow 2=\nu_3$ so, $tidy(level\nu\pi_3\downarrow 1)(level\nu\pi_3\downarrow 1)$ being *true* for all ν, we must have $\nu\geq\nu_3$ whenever ν satisfies $lead1(\rho_3[\![I]\!]) > \#\rho_3[\![I]\!] - \#(level\nu\pi_3)[\![I]\!]$.

Let π be of the form $level\nu\pi_3$ for some ν; a trivial induction on ν_1 demonstrates that for all $\hat{\omega}_0$, $\hat{\omega}_1$ and ν_1 $(seen3\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_5 \wedge known\pi\hat{\omega}_1\hat{\pi}_4 \wedge \sim\grave{\omega}_1=\grave{\delta}_3) \supset (known\pi\hat{\omega}_0\hat{\pi}_4 \wedge \sim\grave{\omega}_0=\grave{\delta}_3)$. Since $pat\hat{\pi}_4\nu\hat{\pi}_3=true$ we know that $found(level\nu_3\pi_3)\hat{\varepsilon}_3\hat{\pi}_4=true$; in particular if $\hat{\varepsilon}_3$:L*×L* any $\hat{\alpha}$ having $gyven\hat{\alpha}\hat{\varepsilon}_3=true$ is subject to $known(level\nu_3\pi_3)\hat{\alpha}\hat{\pi}_4=true$. No $\hat{\alpha}$ such that $known(level\nu_3\pi_3)\hat{\alpha}\hat{\pi}_4=true$ can be such that $\grave{\alpha}=\grave{\delta}_3$, as $known(level\nu_3\pi_3)(ravel\rho_4\rho_4[\![I]\!]\downarrow 1\downarrow 1, \grave{\delta}_3)\hat{\pi}_4=true$ and $p\hat{\pi}_4=true$. Hence for all $\hat{\omega}_0$ and ν_1
$seen3\nu_1\hat{\omega}_0\hat{\delta}_3\hat{\pi}_5 \supset (\hat{\omega}_0=\hat{\delta}_3 \vee \hat{\omega}_0=\hat{\varepsilon}_3 \vee (known(level\nu_3\pi_3)\hat{\omega}_0\hat{\pi}_4 \wedge \sim\grave{\omega}_0=\grave{\delta}_3))$. Whenever $gyven\hat{\omega}_1(pop(level\nu_3\pi_3\downarrow 2)\upsilon_5, pop(level\nu_3\pi_3\downarrow 2)\grave{\upsilon}_5)=true$, $\hat{\omega}_1$ is in L×L or V×V because $pat\hat{\pi}_4\nu_3\hat{\pi}_3=true$, and again we never have $\grave{\omega}_1=\grave{\delta}_3$.

Collecting these assertions together we see that for all ν, π and $\hat{\omega}$ with $\pi=level\nu\pi_3$
$hoten\hat{\omega}(revert\rho\rho_5, revert\rho\grave{\rho}_5) \supset (known\pi\hat{\omega}\hat{\pi}_4 \wedge \sim\grave{\omega}=\grave{\delta}_3) \vee (\hat{\omega}=\hat{\delta}_3 \wedge \nu\geq\nu_3)$ while $gyven\hat{\omega}(pop\upsilon\upsilon_5, pop\upsilon\grave{\upsilon}_5) \supset (known\pi\hat{\omega}\hat{\pi}_4 \wedge \sim\grave{\omega}=\grave{\delta}_3)$. From the definition of *known* in 3.2.1 it follows that
$known\pi\hat{\omega}\hat{\pi}_5 \supset (known\pi\hat{\omega}\hat{\pi}_4 \wedge \sim\grave{\omega}=\grave{\delta}_3) \vee ((\hat{\omega}=\hat{\delta}_3 \vee \hat{\omega}=\hat{\varepsilon}_3) \wedge \nu\geq\nu_3)$ for all ν, π and $\hat{\omega}$ having $\pi=level\nu\pi_3$ or $\pi=level\nu\pi_5$. An awareness that $p_0\hat{\pi}_4=true$ and that $found(level\nu_3\pi_3)\hat{\varepsilon}_3\hat{\pi}_4=true$ now allow us to assert that for all ν and $\hat{\omega}$ such that $known(level\nu\pi_5)\hat{\omega}\hat{\pi}_5=true$ we have $found(level\nu\pi_5)\hat{\omega}\hat{\pi}_5=true$. Furthermore when $known(level\nu\pi_3)\hat{\alpha}\hat{\pi}_5=true$ we have $known(level\nu\pi_3)\hat{\alpha}\hat{\pi}_4=true$ and as $pat\hat{\pi}_4\nu_3\pi_3=true$ $known(level\nu\pi_3)\hat{\alpha}\hat{\pi}_3=true$. Thus $p_0\hat{\pi}_5=true$ and if

$\wedge\{w\hat{\omega}\hat{\pi}_5 \mid kent1\hat{\omega}\hat{\pi}_5\}=true$ we shall have $p\hat{\pi}_5 \wedge spun[\![\Delta]\!]1\psi_3\hat{\pi}_3\hat{\pi}_5=true$.⊁

A mundane induction on ν_1 establishes that for all $\hat{\omega}_0$, $\hat{\omega}_1$ and ν_1 $seen1\nu_1\hat{\omega}_0\hat{\omega}_1\hat{\pi}_5 \wedge (kent1\hat{\omega}_1\hat{\pi}_4 \vee \hat{\omega}_1=\hat{\delta}_3) \supset (kent1\hat{\omega}_0\hat{\pi}_4 \vee \hat{\omega}_0=\hat{\delta}_3)$, so for every $\hat{\omega}$ $kent1\hat{\omega}\hat{\pi}_5 \supset (kent1\hat{\omega}\hat{\pi}_4 \vee \hat{\omega}=\hat{\delta}_3)$. When $kent1\hat{\omega}\hat{\pi}_4=true$, however, 3.2.6 shows that $kent1((q_0\times q_0)\hat{\omega})((\mathbb{P}q_0\times\mathbb{P}q_0)\hat{\pi}_4)=true$ and consequently that $found(level\nu_3\hat{\pi}_3)\hat{\omega}\hat{\pi}_4=true$, while the fact that $p\hat{\pi}_4=true$ assures us that $w\hat{\omega}\hat{\pi}_4=true$. From 3.2.7 we can now deduce that $w\hat{\omega}\hat{\pi}_5=true$ for all $\hat{\omega}$ having $kent1\hat{\omega}\hat{\pi}_5=true$, and accordingly $p\hat{\pi}_5 \wedge spun[\![\Delta]\!]1\psi_3\hat{\pi}_3\hat{\pi}_5=true$.

Hence $a\langle \acute{\zeta}_3\acute{\rho}_5\acute{\upsilon}_5\acute{\sigma}_5, \grave{\zeta}_3\grave{\rho}_5\grave{\upsilon}_5\grave{\sigma}_5\rangle=true$ and, more generally, $a\langle \acute{\zeta}_4\acute{\rho}_4\acute{\upsilon}_4\acute{\sigma}_4, \grave{\zeta}_4\grave{\rho}_4\grave{\upsilon}_4\grave{\sigma}_4\rangle=true$ for all $\hat{\pi}_4$ having $p\hat{\pi}_4 \wedge pat\hat{\pi}_4\nu_3\hat{\pi}_3=true$. This means that $j\hat{\zeta}_4\nu_3\hat{\pi}_3=true$ and that $c\langle \mathcal{T}[\![\Delta]\!]\acute{\zeta}_3, \mathcal{T}[\![\acute{\Delta}]\!]\psi_3]\!]\grave{\zeta}_3\rangle\hat{\pi}_3=true$ for all ψ_3, $\hat{\zeta}_3$ and $\hat{\pi}_3$ such that $t[\![\Delta]\!]0\acute{\rho}_3=true$ and $\wedge\{c\hat{\zeta}_3\hat{\pi} \mid spun[\![\Delta]\!]1\psi_3\hat{\pi}_3\hat{\pi}\}=true$, which in turn implies $T[\![\Delta]\!]=true$.⊁

The same result can be obtained by similar means when Δ is I=E, $I_1,\ldots,I_n$=E or $I_1,\ldots,I_n$==E. For none of these does the degree of complexity exceed that required above, and indeed for two of them it is considerably smaller.⊁

By methods closely related to those of 2.6.6 we can show that for any Δ_0 and Δ_1 if $D[\![\Delta_0]\!] \wedge D[\![\Delta_1]\!]=true$ then $D[\![\Delta_0$ within $\Delta_1]\!]=true$ whilst if $D[\![\Delta_0]\!] \wedge T[\![\Delta_1]\!]=true$ then $T[\![\Delta_0$ within $\Delta_1]\!]=true$. Likewise a minor variant of 2.6.7 serves to establish that for all $\Delta_1,\ldots,\Delta_n$ $D[\![\Delta_1]\!] \wedge \ldots \wedge D[\![\Delta_n]\!] \supset D[\![\Delta_1$ and...and $\Delta_n]\!]$ and $T[\![\Delta_1]\!] \wedge \ldots \wedge T[\![\Delta_n]\!] \supset T[\![\Delta_1$ and...and $\Delta_n]\!]$. Both proofs demand no arguments other than those introduced already, so they can be safely omitted. Were we to adopt the alternative to *trim* alluded to in 2.1.5 we would be reduced to asserting that $D[\![\Delta_0$ within $\Delta_1]\!]=true$ if $cramped[\![\Delta_0$ within $\Delta_1]\!](\lambda I.\langle 2\rangle)=true$, which would be inimical to the motivation underlying within declarations.

3.3.8. Lemma.

If $T[\![\Delta]\!]=true$ then $T[\![\text{rec }\Delta]\!]=true$ and, when $opts(\mathscr{H}[\![\Delta]\!])=\lambda\psi.true^*$ also, $D[\![\text{rec }\Delta]\!]=true$.

⊲Suppose that ψ_0, $\hat{\pi}_0$, $\hat{\pi}_1$ and $\hat{\zeta}$ satisfy $opts(\mathscr{H}[\![\Delta]\!])\psi_0=true^*$, $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$ and $\bigwedge\{c\hat{\zeta}\hat{\pi}\mid spun[\![\text{rec }\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}\}=true$; let ψ_1 be $\psi_0[false^*/\mathscr{I}[\![\Delta]\!]][true^*/\mathscr{H}[\![\Delta]\!]]$. Because $p_0\hat{\pi}_1=true$, if $\langle\alpha^*,\alpha^*\rangle=\langle news(\#\mathscr{I}[\![\Delta]\!])\acute{\sigma}_1,news(\#\mathscr{I}[\![\Delta]\!]\grave{\sigma}_1)\rangle$ and $\theta_2=\langle updates\acute{\alpha}^*dummy^*\acute{\sigma}_1,updates\grave{\alpha}^*dummy^*\grave{\sigma}_1\rangle$, θ_2 is proper. Define $\hat{\upsilon}_2=\hat{\upsilon}_1$ and $\hat{\rho}_2=\langle\acute{\rho}_1[\acute{\alpha}^*//\mathscr{I}[\![\Delta]\!]][dummy^*/\mathscr{H}[\![\Delta]\!]],\grave{\rho}_1[\grave{\alpha}^*/\mathscr{I}[\![\Delta]\!]\mathscr{H}[\![\Delta]\!]]\rangle$, in terms of which $\mathscr{D}[\![\text{rec }\Delta]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1$ and $\mathscr{D}[\![d[\![\text{rec }\Delta]\!]\psi_0]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1$ are $\mathscr{T}[\![\Delta]\!]\acute{\zeta}\acute{\rho}_2\acute{\upsilon}_2\acute{\sigma}_2$ and $\mathscr{T}[\![\mathscr{I}[\![\Delta]\!]\psi_1]\!]\grave{\zeta}\grave{\rho}_2\grave{\upsilon}_2\grave{\sigma}_2$ respectively. If $\hat{\pi}$ satisfies $spun[\![\Delta]\!]1\psi_1\hat{\pi}_2\hat{\pi}=true$ then $spun[\![\text{rec }\Delta]\!]0\psi_0\hat{\pi}_1\hat{\pi}=true$ since if $\hat{\pi}=\mathbb{R}q_0\hat{\pi}$ and $I:\mathscr{H}[\![\Delta]\!]$ $\mathbb{R}q_0\acute{\rho}_0[\![I]\!]=\mathbb{R}q_0\acute{\rho}_2[\![I]\!]\dagger lead1(\acute{\rho}_2[\![I]\!])=\acute{\rho}[\![I]\!]\dagger lead1(\acute{\rho}[\![I]\!])=revert\acute{\rho}_2\acute{\rho}[\![I]\!]$. Hence when $d[\![\text{rec }\Delta]\!]0\acute{\rho}_0=true$ (and $T[\![\Delta]\!]0\acute{\rho}_2=true$ also) to show that $a\langle\mathscr{D}[\![\text{rec }\Delta]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{D}[\![d[\![\text{rec }\Delta]\!]\psi_0]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$ it suffices to verify that $p\hat{\pi}_2\wedge fit\hat{\pi}_2\hat{\pi}_2=true$.

If $\hat{\pi}$ is of the form $level\nu\hat{\pi}_2$ for some ν either $\hat{\pi}=level\nu\hat{\pi}_1$ or $\hat{\pi}=\hat{\pi}_2$; in the first case $known\hat{\pi}\hat{\omega}\hat{\pi}_2\supset known\hat{\pi}\hat{\omega}\hat{\pi}_1$ for all $\hat{\omega}$ from 3.2.2. The second case is disposed of by noting that induction shows that $kent\nu\hat{\omega}\hat{\pi}_2\supset(hoten\hat{\omega}\hat{\rho}_2\vee\hat{\omega}=\langle dummy,dummy\rangle\vee kent\nu\hat{\omega}\hat{\pi}_1)$ for all ν and $\hat{\omega}$. When $\acute{\alpha}:\acute{\alpha}^*$ and $\grave{\alpha}:\grave{\alpha}^*$ $area\acute{\alpha}\acute{\sigma}_1\vee area\grave{\alpha}\grave{\sigma}_1=false$ so $kent3\langle\acute{\alpha},\grave{\epsilon}\rangle\hat{\pi}_1\wedge kent3\langle\acute{\epsilon},\grave{\alpha}\rangle\hat{\pi}_1=false$ for all $\hat{\epsilon}$ as $p_0\hat{\pi}_1=true$; consequently $p_0\hat{\pi}_2=true$. Plainly $w\hat{\omega}\hat{\pi}_2=true$ for all $\hat{\omega}$ having $w\hat{\omega}\hat{\pi}_1=true$, and thus $\bigwedge\{w\hat{\omega}\hat{\pi}_2\mid kent1\hat{\omega}\hat{\pi}_2\}=true$ as $\bigwedge\{w\hat{\omega}\hat{\pi}_1\mid kent1\hat{\omega}\hat{\pi}_1\}=true$; in conjunction with $p_0\hat{\pi}_2=true$ this ensures that $p\hat{\pi}_2\wedge fit\hat{\pi}_2\hat{\pi}_2=true$.

Hence for any ψ_0, $\hat{\pi}_0$, $\hat{\pi}_1$ and $\hat{\zeta}$ having $opts(\mathscr{H}[\![\Delta]\!])\psi_0=true^*$, $p\hat{\pi}_1\wedge fit\hat{\pi}_1\hat{\pi}_0=true$ and $\bigwedge\{c\hat{\zeta}\hat{\pi}\mid spun[\![\text{rec }\Delta]\!]0\psi_0\hat{\pi}_0\hat{\pi}\}=true$ we have $a\langle\mathscr{D}[\![\text{rec }\Delta]\!]\acute{\zeta}\acute{\rho}_1\acute{\upsilon}_1\acute{\sigma}_1,\mathscr{D}[\![d[\![\text{rec }\Delta]\!]\psi_0]\!]\grave{\zeta}\grave{\rho}_1\grave{\upsilon}_1\grave{\sigma}_1\rangle=true$ or $d[\![\text{rec }\Delta]\!]0\acute{\rho}_0=false$. In accordance with 3.2.4 this means that $D[\![\text{rec }\Delta]\!]=true$.⊳

3.3.9. Theorem.

The meanings accorded by *new* stack semantics to a Mal program and by *novel* store semantics to its transform are comparable, provided that the program obeys the constraints of 3.1.4, the lattice of locations is infinite and every recursive declaration rec Δ_0 embedded in the program satisfies the equality $opts(\mathscr{M}[\![\Delta_0]\!])=\lambda I.true^*$.

⊰The insistence that all declarations of the form rec Δ_0 embedded in the program be subject to $opts(\mathscr{M}[\![\Delta_0]\!])=\lambda I.true^*$ ensures not merely that we can apply 3.3.8 but that the transform of the program is devoid of recursive declarations by incidence and satisfies the conditions of 2.6.9. Thus although 3.3.1, for instance, is expressed in terms of *new* store semantics it holds for *novel* store semantics as well. Moreover the absence of both G and P from the domain W defined in 3.2.1 makes it possible to assert in a similar manner that *new* stack semantics is exactly cquivalent to *novel* stack semantics. In consequence the meaning of the program according to stack semantics is comparable with that of its transform (under the rules of 1.4.6) according to store semantics whichever means of storage allocation is implicit in the relevant equations.

Should the program satisfy the conditions under which 2.7.6 is applicable rather more than this can be said. Under these circumstances, the program and its transform are equivalent when both are evaluated using *new* store semantics, so in fact the outcome of the program suggested by stack semantics is comparable with the outcome suggested by store semantics. From 2.3.9 it therefore follows that the answer obtained by implementing the language with the aid of stacks is precisely that predicted by standard semantics.⊱

3.4. Different control structures for languages.

3.4.1. Surrogate routines.

Elaborate though it may be, Ma1 does not evince the essentials of all computer languages. Its lack of format statements and file-handling facilities is unimportant, however, since they could readily be embedded in its semantic equations. Likewise the introduction of arrays and structures (which are akin to members of L*) would merely require us to complicate the technical details of appendix 1 without disturbing its core. More significant is the omission from Ma1 of any means of checking types and encoding implicit coercions during compilation, which will be treated in 3.6.1. In this section we shall extend the formalism of 1.3.4 to cope with methods of controlling the execution sequence of a program which differ from those of Pa1.

The use of labels is frequently opposed on the ground that it transports the intended meaning of a program beyond the bounds of human and computer comprehension. Even when labels are permitted languages tend to bar entry to a block after it has already been left by imposing the constraints of 3.1.4. The worst effects of doing this can be mitigated by introducing coroutines, which sometimes achieve less wayward transfers of control than are afforded by the assignment and invocation of stored label entry points. Accordingly we begin by augmenting the syntax of Ma1 with a set of primitives which exhibit the salient aspects of coroutines.

Those languages which provide coroutines give them a mode of definition which resembles that for subroutines rather than that for labels. We follow this convention in our adoption of cr()E, which by analogy with fn()E is deemed to be a coroutine taking the list nil as a parameter. Unlike the corresponding subroutine, however, the point to which E returns its result is determined not

when cr()E is applied but when it is set up; in 3.4.2 for brevity we shall use ρ⟦res⟧ to provide this point, but plainly there are many more plausible possibilities. Whereas parameters cannot be passed to labels by goto statements, here we can exploit the extra freedom allowed by abstractions in order to create crI.E, $crI_1,\ldots,I_n$.E, crI..E and $crI_1,\ldots,I_n$..E, which require the value of bound variables to be supplied when the coroutine is first activated. Such activations occur on encountering resume E_0 with E_1, when E_1 is passed as an argument to a coroutine E_0 which then takes over control from the current one; should the latter ever be invoked again execution will continue from the textual position following this resumption. This represents the most significant distinction between coroutines and denoted label entry points, in that only the continuation signified by the former can be altered as the execution of the program proceeds.

There is one further difference between our coroutines and labels: we permit the suspension of coroutines so that when a resume instruction demands that control pass to a suspended coroutine ρ⟦res⟧ is summoned instead. Thus cancel E will suspend E while adjoin E will append E once more to the set of coroutines which can be resumed. This apparatus is introduced merely to add force to our claim that the technique to be developed in 3.4.2 can supply the formal semantics of any of the constructs discussed by Dahl [3] by modelling the 'sequencing set' or 'activation list' of a simulation language.

As intimated above a subroutine is subordinate to the program which calls it but a coroutine, like a label entry point, need not be. The declaration rec I==cr()E is therefore more closely linked with I::E than with rec I==fn()E; the declaration I==cr()E corresponds not to setting a label by incidence but to evaluating the expression E in an environment which binds I to

a value created in an outer block. Yet though we may permit the identifier I set by I::E to denote a continuation we would be unwise to allow the same liberty to the identifier declared by rec I==cr()E, since the environment would then need to be modified explicitly at every dynamic occurrence of a resume instruction. Moreover we would have to alter all the environments attached to coroutines as well as the current one, for if

rec *g*==cr*z*.resume *g* with (*z*+1) inside resume *g* with 0

is to return 1 as its result the same should be true of

rec *g*==cr*z*.resume *h* with (*z*+1) and *h*==cr*z*.resume *h* with (*z*+1)

inside resume *g* with 0,

which returns 2 unless these alterations are effected. The formalism of 2.1.1 could perhaps be made to accommodate such alterations but standard semantics cannot, so we are obliged not to take the denotation of a coroutine to be a continuation.

The obvious means of ensuring that the variables *g* and *h* in the program above change their continuations simultaneously would be to make them both denote a location containing the appropriate continuation. On resuming the execution of a coroutine the appropriate linkage information could then be assigned to the location corresponding to the coroutine the execution of which was being halted. We shall not adopt this expedient because it would corrupt the intention of declarations by incidence, according to which *g*==cr()dummy within *g*:=dummy should not modify the value of *g*. Though it may be that languages should provide only coroutines stored in this manner, no conventional simulation language does so and hence we must cater for existing constructs less obliquely. Underlying a coroutine resumption is a concept as different from that of an assignment as it is from that of a subroutine application, so we shall isolate it in a separate mechanism.

There is perhaps an analogy between the status of stored coroutines and on conditions. One behemoth of a language permits on conditions to be declared at points other than block entry, thereby altering dynamically the environment bound to subroutines and label entry points. It is possible to describe such silliness formally either by supplying part of the environment to a subroutine only when it is applied or by adopting a stratagem akin to that of 3.4.2. The proper course of action, however, is to design an entirely different language in which on conditions can be declared only at the heads of blocks; similarly, then, it may be that coroutines should always be explicitly stored objects which cannot be denoted and which do not require the contrivance which we shall now discuss.

3.4.2. Controlled queues of processes.

The considerations of 3.4.1 drive us to provide an abstract version of the sequencing set, a feature of all implementations of simulation languages which is not needed by other languages. Akin to the lattice L is a flat lattice of processes, I, which we shall take to be a summand of D. Whenever an identifier is declared to be a coroutine it denotes not a continuation but a process through which is channelled each request for knowledge about the coroutine. Associated with any program is a queue taken from a domain Q in such a way that if τ:Q all that is known about the process ι:I can be inferred from $\tau\iota$. For Mal we take this domain of queues to be I$\mapsto$[T×T×K°], factoring out the undesirable elements as in 1.3.1. The lattice K° provides the continuation corresponding to the coroutine while the second constituent truth value lattice indicates whether or not it has been suspended by applying cancel. The first constituent serves a purpose very similar to that of *area* in

that it establishes which processes are in use; to acquire a fresh process the semantic equations invoke *near*:Q→S→I, which is taken to be any continuous function satisfying

$\lambda\tau\sigma.\tau(near\tau\sigma)\downarrow 1 \vee \tau(near\tau\sigma)\downarrow 2 = \lambda\tau\sigma.\bigwedge\{\tau\iota\downarrow 1 \vee \tau\iota\downarrow 2 \mid \iota : I\} \rightarrow \bot, false.$

Somewhat similar to *update* is *impose*:I→[T×T×K°]→Q→Q, which can be defined by

$impose = \lambda\iota\omega\tau.(\lambda\iota'.\iota' = \iota \rightarrow \omega, \tau\iota).$

In consequence the iterated version of *near*, *nears*:N→Q→S→I*, is given by

$nears = \lambda\nu\tau\sigma.\nu = 0 \rightarrow \langle\rangle, (\lambda\iota.\langle\iota\rangle \S nears(\nu-1)(impose\iota(\langle true\rangle \S \tau\iota\dagger 1)\tau)\sigma)(near\tau\sigma)$

At any point during the execution of a program the name of the current process must be available alongside a sequencing set in Q holding the continuations which supplant the current one when other coroutines are resumed. This sequencing set cannot be built into the continuations because it may have been modified by the time they are set in motion; the relevant process names, however, are immutable and can therefore be incorporated in members of C. Thus the logical development of 1.3.2 involves setting C=Q→S→A, K=E→C and X=U→C, where A is a suitable domain of answers. Before an expression or a declaration can be supplied with a continuation both the environment and the process on which it operates must be determined, so typical valuations are now $\mathscr{E}$:Exp→U→I→K→C and $\mathscr{D}$:Dec→U→I→X→C, while the value domains are provided by the equalities V=I+B+L*+J+F, E=L+V and D=E+G.

As subroutines do not have continuations sealed into them, they are not associated with particular processes and $\mathscr{F}$:Abs→U→F can continue in use; now F=[E→I→K→C]° so that $\mathscr{F}[\![\text{fn}()\text{E}]\!]$ is

$\lambda\rho\iota\kappa\tau\sigma.\kappa(\lambda\alpha\iota'\kappa'.rv(\lambda\beta.\beta \mid L^* = \langle\rangle \rightarrow \mathscr{L}[\![E]\!]\rho\iota'\kappa', \tau)\alpha)\tau\sigma$ and $\mathscr{F}[\![E_0 E_1]\!]$ is

$\lambda\rho\iota\kappa.(\lambda\psi.run\langle\mathscr{F}[\![E_0]\!]\rho\iota, \mathscr{L}[\![E_1]\!]\rho\iota\rangle\ \psi)$

$(\lambda\varepsilon^*.\varepsilon^*\downarrow 1 : F \rightarrow (\varepsilon^*\downarrow 1)(\varepsilon^*\downarrow 2)\iota\kappa, rv(1 \le \beta \mid N \le \#\varepsilon^*\downarrow 1 \mid L^* \rightarrow \kappa(\varepsilon^*\downarrow 1\downarrow\beta), \tau)(\varepsilon^*\downarrow 2)),$

where *run*, *rv* and *lv* are identical with their counterparts in 1.3.5 except for the presence of a member of Q which plays no part in the evaluation. Labels, on the other hand, contain process names in their values, with the effect that when a jump is made the current process reverts to that at the time of definition of the relevant label; this is made plain by the fact that $\mathscr{P}[\![\mathrm{I:E}]\!]\rho\iota\kappa\downarrow 1$ and $\mathscr{L}[\![\mathrm{I::E}]\!]\rho\iota\kappa\downarrow 1$ are both $\mathscr{E}[\![\mathrm{E}]\!]\rho\iota\kappa$. For reasons similar to those adduced in 1.4.1 recursive declarations must be dealt with by means of $\mathscr{S}[\![\Delta]\!]\rho\iota\tau\sigma\downarrow\nu$, which is $\lambda\kappa'\tau'\sigma'.\mathscr{T}[\![\Delta]\!]\rho\iota(\lambda\rho''\tau''\sigma''.\kappa'(\rho''[\![\mathscr{H}[\![\Delta]\!]\downarrow\nu]\!]\,|\,\mathrm{E})\tau'\sigma')\tau\sigma$ when $1\leq\nu\leq\#\mathscr{H}[\![\Delta]\!]$, thereby implying that $\mathrm{G}=[\mathrm{K}\to\mathrm{C}]^\circ$ and that $\mathscr{E}[\![\mathrm{I}]\!]$ is $\lambda\rho\iota\kappa\tau\sigma.(\lambda\delta.\delta:\mathrm{G}\to\delta\kappa\tau\sigma,\kappa\delta\tau\sigma)(\rho[\![\mathrm{I}]\!]\downarrow 1)$. The remaining semantic equations in appendix 1 are left almost unscathed by the introduction of coroutines, the sole changes being the intrusion of two extra arguments which do not interfere with the evaluation; $\mathscr{E}[\![\mathrm{E}_0:=\mathrm{E}_1]\!]$, for instance, becomes

$\lambda\rho\iota\kappa.run\langle\mathscr{L}[\![\mathrm{E}_0]\!]\rho\iota,\mathscr{R}[\![\mathrm{E}_1]\!]\rho\iota\rangle(\lambda\varepsilon^*\tau\sigma.\kappa(dummy)\tau(update(\varepsilon^*\downarrow 1)(\varepsilon^*\downarrow 2)\sigma))$.

When we create a new coroutine we select a process which is not in use and impose on it the appropriate continuation together with tokens which reveal that the process is active. Hence the outcome of $\mathscr{E}[\![\mathrm{cr()E}]\!]$ is

$\lambda\rho\iota\kappa\tau\sigma.(\lambda\omega.\kappa(near\tau\sigma)(impose(near\tau\sigma)\omega\tau)\sigma)$

$\quad\langle true,true,rv(\lambda\beta.\beta\,|\,\mathrm{L}^*=\langle\rangle\to\mathscr{L}[\![\mathrm{E}]\!]\rho(near\tau\sigma)(\rho[\![\mathrm{res}]\!]\downarrow 1),\tau)\rangle$,

which in view of our remarks about labels ensures that after running E control reverts to the process creating the coroutine. The equations for the other kinds of coroutine are related to those for fnI.E, $\mathrm{fnI}_1,\ldots,\mathrm{I}_n$.E, fnI..E and $\mathrm{fnI}_1,\ldots,\mathrm{I}_n$..E in precisely the same way as that for cr()E is related to that for the corresponding subroutine.

The connection between a process resumption and a subroutine application is only rendered more devious by the insis-

tence that a passive process never be resumed unless by chance it is that corresponding to ρ⟦res⟧. In consequence we take

$\mathscr{E}$⟦resume E_0 with E_1⟧ to be

λρικ.(λψ.*run*⟨$\mathscr{R}$⟦E_0⟧ρι,$\mathscr{L}$⟦E_1⟧ρι⟩ψ)

(λε*τσ.(λτ'.(τ'(ε*↓1|I)↓2→τ'(ε*↓1|I)↓3,ρ⟦res⟧↓1)(ε*↓2)τ'σ)

(*impose*ι⟨τι↓1,τι↓2,κ⟩τ)).

We can formulate the other properties of our coroutines in an even more trivial manner, for $\mathscr{E}$⟦cancel E⟧ is

λρικ.$\mathscr{R}$⟦E⟧ρι(λετ.ε:I→κε(*impose*ε⟨τε↓1,*false*,τε↓3⟩τ),τ) while

$\mathscr{E}$⟦adjoin E⟧, its opposite, is

λρικ.$\mathscr{R}$⟦E⟧ρι(λετ.ε:I→κε(*impose*ε⟨τε↓1,*true*,τε↓3⟩τ),τ). Notice that in regarding I as a summand of V we are clearing the way for some rather strange sharing patterns between coroutines which do not arise in more normal languages. Thus in

g=0 inside *h*==cr()dummy inside *g*:=*h*, for example, assigned to *g* is not a continuation but a process marking a continuation which is altered whenever the continuation tallying with *h* is altered.

3.4.3. Variant formulations.

The queues introduced for standard semantics naturally have counterparts in store semantics. The motivation given in 2.1.1 for converting K° into Z°×U×Y remains valid in the new situation provided a reasonable version of Z is adopted, so for the store semantic equations governing the coroutines of 3.4.2 we can take Q to be I↦[T×T×[Z°×U×Y]]. We might expect to require valuations belonging to the lattices Exp→Z→I→U→Y→Q→S→A and Dec→Z→I→U→Y→Q→S→A but luckily they are unnecessary. The environment and stack involved when an expression or a declaration is evaluated need not be supplied as arguments alongside the queue τ because they are already embedded in it. In addition the name of the current process ι can be kept in τ by tagging it so that

it alone among the processes satisfies $\sim\tau\iota\downarrow 1\wedge\tau\iota\downarrow 2=\mathit{true}$; doing so is inelegant but perhaps more in keeping with the interpretive nature of store semantics. Accordingly we extract the name of the current location with the aid of last:Q→S→I, which is defined by $\mathit{last}=\lambda\tau\sigma.\bigsqcup\{\iota\,|\sim\tau\iota\downarrow 1\wedge\tau\iota\downarrow 2\}$.

For store semantics Z is therefore Q→S→A, and the valuations are exemplified by $\mathscr{E}$:Exp→O and $\mathscr{D}$:Dec→O where O is Z→Z as before. The value domains continue to have the forms imposed on them in 3.4.2, but now J=Z°×U×Y, F=O°×U and G=O°×Q×S. The environment lattice U undergoes a slight change because now ρ⟦rec⟧ must tuck away not the current state vector but the queue and the store; consequently U is still [Ide↛D°*]×J*×P* provided that P is Q×S. Moreover the definitions of *novel*, *replace* and *recur* must be altered to take account of the locations accessible from processes other than the current one; after doing this we can demand that $\mathscr{E}$⟦I⟧ be

$$\lambda\tau\sigma.(\lambda\iota.(\lambda\delta.\delta\text{:G}\to(\lambda\iota.(\lambda\beta.(\delta\downarrow 1)\zeta(\mathit{impose}\iota\langle \mathit{false},\mathit{true},\beta\rangle(\delta\downarrow 2))(\delta\downarrow 3))$$
$$\langle(\delta\downarrow 2)\iota\downarrow 3\downarrow 1,((\delta\downarrow 2)\iota\downarrow 3\downarrow 2)[\langle\tau,\sigma\rangle/\mathbf{rec}],(\delta\downarrow 2)\iota\downarrow 3\downarrow 3\rangle)$$
$$(\mathit{last}(\delta\downarrow 2)(\delta\downarrow 3)),$$
$$\zeta(\mathit{impose}\iota\langle \mathit{false},\mathit{true},\langle\tau\iota\downarrow 3\downarrow 1,\tau\iota\downarrow 3\downarrow 2,\langle\delta\rangle\,\S\tau\iota\downarrow 3\downarrow 3\rangle\rangle\tau)\sigma)$$
$$((\tau\iota\downarrow 3\downarrow 3)⟦\text{I}⟧\downarrow 1))(\mathit{last}\tau\sigma))$$

and that $\mathscr{S}$⟦Δ⟧τσ↓ν be

$$\langle\lambda\zeta'\tau'\sigma'.\mathscr{T}⟦\Delta⟧(\lambda\tau''\sigma''.(\lambda\iota.(\lambda\beta.\mathit{recur}\zeta'(\mathit{impose}\iota\langle \mathit{false},\mathit{true},\beta\rangle\tau'')\sigma'')$$
$$\langle\tau''\iota\downarrow 3\downarrow 1,\tau''\iota\downarrow 3\downarrow 2,\langle(\tau''\iota\downarrow 3\downarrow 2)⟦\mathscr{E}⟧\Delta⟧\downarrow\nu⟧\downarrow 1\,|\,\text{E}\rangle\rangle)$$
$$(\mathit{last}\tau''\sigma'')),$$
$$\tau,\sigma\rangle$$

when $1\le\nu\le\#\mathscr{E}⟦\Delta⟧$. We shall not provide more of the equations needed by the extended version of store semantics because they can readily be obtained by subjecting appendix 2 to the refinements of 3.4.2.

The formulation of stack semantics in terms of queues can likewise be put into effect by translating the equations of appendix 3. Since each process must have access to its own private environment and stack Q is again $I \to [T \times T \times [Z^\circ \times U \times Y]]$ although now $J = Z^\circ$ and $F = 0^\circ$. We choose not to conflate Q and S because to us the principles beneath the components of S are very different from the ones underlying Q, which is concerned with the flow of control through programs rather than with their results. The queue is regarded as housing all the information private to a process whilst the store provides a channel for communication between processes. In other languages the extent to which a process can send or receive data using this channel will be determined by flags in T attached to the relevant entry in the queue. Hence stack semantics can continue to invoke valuations $\mathscr{E}: Exp \to 0$ and $\mathscr{D}: Dec \to 0$, for which $\mathscr{E}[\![I]\!]$ is now taken to be

$$\lambda\zeta\tau\sigma.(\lambda\iota.(\lambda\delta.\zeta\ impose\iota\langle false, true, \langle \tau\iota\downarrow 3\downarrow 1, \tau\iota\downarrow 3\downarrow 2, (\delta) \S \tau\iota\downarrow 3\downarrow 3\rangle\rangle \tau)\sigma)$$
$$(ravel(\tau\iota\downarrow 3\downarrow 2)(\tau\iota\downarrow 3\downarrow 2)[\![I]\!]\downarrow 1\downarrow 1))(last\tau\sigma)),$$

as $V = I + B + L^* + J + F$, $E = L + I + B + L^* + J + F$ and $D = L + I + B + L^* + J + F$.

When the semantic equations are set up correctly it is possible to prove an analogue of 2.3.9 relating standard and stack semantics in the presence of queues. Besides the predicates of 2.2.5 this involves

$$q = \lambda\hat{\tau}.\bigwedge\{\acute{\tau}\iota\downarrow 1 = \grave{\tau}\iota\downarrow 1 \wedge \acute{\tau}\iota\downarrow 2 = \grave{\tau}\iota\downarrow 2 \wedge k\langle \acute{\tau}\iota\downarrow 3, \grave{\tau}\iota\downarrow 3\rangle \mid \iota : I\}_\wedge (last\grave{\tau}_\perp : I),$$

so that it deals only with a queue τ if there is exactly one process ι having $\sim\tau\iota\downarrow 1 \wedge \tau\iota\downarrow 2 = true$. In addition we can define a version of the function $seen: N \to N \to [W^\circ \times W^\circ] \to [W^\circ \times W^\circ] \to [P^\circ \times P^\circ] \to T$ more appropriate than that of 2.1.6 to store semantics with queues, in which $W = L + I + B + L^* + J + F + J + P + [T \times T \times [Z^\circ \times U \times Y]]$ and $P = Q \times S$; for instance $seen\nu_0(\nu_1+1)\hat{\omega}_0\hat{\omega}_1\hat{\pi}$ is $seen\nu_0\nu_1\hat{\omega}_0\langle(\acute{\pi}\downarrow 1)\acute{\omega}_1, (\grave{\pi}\downarrow 1)\grave{\omega}_1\rangle\hat{\pi}$ if $\hat{\omega}_1 : I \times I$ and is

$\forall$*iseen*$\nu_0\nu_1\hat{\omega}_0\hat{\omega}_2\Uparrow$

$\wedge$(*hoten*$\hat{\omega}_2$($\vec{\omega}_1$↓3↓2,$\tilde{\omega}_1$↓3↓2)$\vee$(*gyven*$\hat{\omega}_2$($\vec{\omega}_1$↓3↓3,$\tilde{\omega}_1$↓3↓3)$\wedge\nu_0$<2))|$\hat{\omega}_2$:W×W}

if $\hat{\omega}_1$:[T×T×[Z°×U×Y]]×[T×T×[Z°×U×Y]]. With the assistance of this function we can establish theorems like 2.5.9 and 2.6.9 whilst a version suitable for stack semantics can be used for an analogue of 3.3.9. By extending the definition of *crushed* given in 1.5.4 so that any expression satisfying it is devoid of coroutine resumptions we can even validate 1.5.8 and 2.7.7 for semantic equations with queues.

It is frequently claimed that stored free variable lists can perform all the tasks which coroutines can carry out. Here we shall briefly indicate how to show that this is so for our choice of coroutines by providing an explicit conversion procedure which erases them from a program. We associate a member of L* with every member of I called upon by the program in such a way that changes in the contents of the constituent locations reflect changes in the flags and continuation attached to the process. To achieve this we pick out identifiers I_0, I_1 and I_2 occurring nowhere in the program and define

move:[Exp+Dec]→Ide→[Exp+Dec] by such equations as

move⟦cr()E⟧=λI.I_0=(true,true,fn()res *move*⟦E⟧⟦I_0⟧) inside I_0;

move⟦resume E_0 with E_1⟧=λI.I_0=*move*⟦E_0⟧⟦I⟧ and I_1=*move*⟦E_1⟧⟦I⟧

inside I_2=fnI.res I

inside val (I_3:=fn .res I;

(if $I_0$2 then $I_0$3 else I_2)I_1);

move⟦cancel E⟧=λI.(*move*⟦E⟧⟦I⟧)2:=false;

move⟦adjoin E⟧=λI.(*move*⟦E⟧⟦I⟧)2:=true.

The other kinds of coroutine are converted in like manner to cr()E whereas the expressions inherent in Mal are not altered; for example, *move*⟦E_0; E_1⟧ is λI.(*move*⟦E_0⟧⟦I⟧; *move*⟦E_1⟧⟦I⟧). The problem now becomes that of proving that when I_0 does not occur

at all in E or Δ $\mathscr{E}[\![E]\!]$ and $\mathscr{D}[\![\Delta]\!]$ compute the same answers as $\mathscr{E}[\![move[\![E]\!][\![I_0]\!]]\!]$ and $\mathscr{D}[\![move[\![E]\!][\![I_0]\!]]\!]$ respectively. We therefore translate these programs using standard semantics, obtaining entities in which I_0, I_1 and I_2 do not appear (by virtue of 1.5.2), and then take members of Z, Q and S suitable for store semantics which can be shown to be equivalent to them by using the extension of 2.3.9 alluded to above. This reduces the problem to considering pairs in $[Q\times S]^\circ\times[Q\times S]^\circ$ for which we take $yclept\hat{\omega}(\langle \acute{f},\acute{\sigma}\rangle,\langle \grave{\tau},\grave{\sigma}\rangle)$ to be

$\forall\{(\acute{f}\iota\downarrow 1 \vee \acute{f}\iota\downarrow 2 \rightarrow hoten\hat{\omega}(\acute{f}\iota\downarrow 3\downarrow 2,\grave{\tau}\iota\downarrow 3\downarrow 2) \vee gyven\hat{\omega}(\acute{f}\iota\downarrow 3\downarrow 3,\grave{\tau}\iota\downarrow 3\downarrow 3), false)$

$\vee gyven\hat{\omega}(\acute{\sigma}\downarrow 2 \S \acute{\sigma}\downarrow 3,\grave{\sigma}\downarrow 2 \S \grave{\sigma}\downarrow 3) \vee(\hat{\omega}=\langle last\acute{f}\acute{\sigma},(\grave{\tau}(last\grave{\tau}\grave{\sigma})\downarrow 3\downarrow 2)[\![I_0]\!]\downarrow 1\rangle)\,|\,\iota:I\}.$

Using this and the function *seen* mentioned above we extract all the values witnessed by corresponding states $\langle \acute{f},\acute{\sigma}\rangle$ and $\langle \grave{\tau},\grave{\sigma}\rangle$ so that we can build up inclusive predicates resembling those of 2.4.5. Finally these are applied in a proof akin to that of 2.5.9 establishing that when $k\hat{\zeta}(\langle \acute{f},\acute{\sigma}\rangle,\langle \grave{\tau},\grave{\sigma}\rangle)=true$ and $rent[\![E]\!](\acute{f}\,(last\acute{f}\acute{\sigma})\downarrow 3\downarrow 2)=true$ we must have $c(\mathscr{E}[\![E]\!]\zeta,\mathscr{E}[\![move[\![E]\!][\![I_0]\!]]\!]\grave{\zeta})\langle\langle \acute{f},\acute{\sigma}\rangle,\langle \grave{\tau},\grave{\sigma}\rangle\rangle=true$.

3.5. Parallel programming.

3.5.1. Indivisible operations.

The utility of the technique introduced in 3.4.2 is neither confined to the description of coroutines nor vitiated by the ugliness of some language constructs; indeed, we have already alluded to its role in highlighting the obscurities of on conditions. Here we shall model parallel programs by adapting process queues so that they act in a non-deterministic fashion outside the control of the programmer. Our approach will differ from that of Milner [11] mainly because of our preference for composing a sequence of operations as a continuation instead of dissecting it.

We demand that the workings of computers can be analysed into discrete operations which cannot be interrupted by one another; it is fortunate that every implementation which avoids chaos satisfies this requirement, for a continuum of operations would need a different treatment. Though a typical computer might debar assignments to a location while its content is being extracted, we do not have to presume that the hardware locks arise at such a macroscopic level. We shall nonetheless do so to enable us to retain a model for storage akin to that of 1.3.1, but there would be no difficulty about adapting all that follows to computers permitting greater liberties: for instance, were the individual bits in a word protected from being overwritten during the examination of its content, we would replace $L \rightarrow [T \times V]$ by $L \rightarrow [T \times [\{0\}^\circ + \{1\}^\circ]^*]$ and encode the members of B and L as bit patterns in $[\{0\}^\circ + \{1\}^\circ]^*$.

The 'indivisible operations' or 'basic steps' pertaining to a particular implementation are also such that any two instances of them which are not protected from one another can be performed simultaneously. Thus if at most one assignment can

be made to any location at a given moment, *update* defines an indivisible operation such that for all proper α_1, α_2, β_1 and β_2 $update\alpha_1\beta_1 \circ update\alpha_2\beta_2$ is the effect of interleaving $update\alpha_1\beta_1$ and $update\alpha_2\beta_2$ arbitrarily unless $\alpha_1=\alpha_2$. Consequently should the implementation be executing several processes in parallel we can align them arbitrarily to obtain a sequential computation with the same effect; all that we must insist upon is that the temporal order of the basic steps in each individual process be preserved by the sequential computation.

In the mathematical model the counterpart to the mingling of indivisible operations will be the composition of primitive functions (like $update\alpha_1\beta_1$ and $update\alpha_2\beta_2$) in a certain order. These functions will be attached to processes by τ, a member of a domain Q similar to that of 3.4.2. After applying the function given by some process ι:I another process will be chosen while the remaining primitives required by ι will be preserved in τ as the continuation corresponding to ι. The choice of process will be made by *next*, which will tentatively be assumed to belong to I→Q→S→I; thus $next\iota\tau\sigma$ will be the process which usurps control after a function provided by ι:I produces τ:Q and σ:S. The nature of *next* will not preclude the possibility that one particular process might be selected at every appeal to it, and in fact when τ contains only one active process this will inevitably be the case. Nevertheless the means whereby processes take control is constrained by a hierarchy quite different from that discussed in 3.4.1, so the present τ will be distinguished from the earlier one by calling it a 'sequel'; the semantics for a language with both coroutines and parallel programming would therefore be endowed with both a queue and a sequel.

For reasons elucidated in 3.5.3 we now take $\tau\iota\downarrow 7$ (rather than $\tau\iota\downarrow 3$) to be the continuation associated with ι:I in τ:Q,

and we tacitly presume that the definition of *impose* given in 3.4.2 is modified so that $\tau\iota$ is permitted to have seven components. Before invoking a process *impose* is used to put the current continuation back into the sequel, and then the fresh continuation is retrieved. This suggests that to switch from one process to another we introduce a function *do* having the form

$\lambda\iota\theta\tau\sigma.(\lambda\tau'.(\tau'(next\,\iota\tau\sigma)|C)\tau'\sigma)$

$\quad(impose\,\iota\langle\tau\iota\downarrow1,\tau\iota\downarrow2,\tau\iota\downarrow3,\tau\iota\downarrow4,\tau\iota\downarrow5,\tau\iota\downarrow6,\theta\rangle\,\tau).$

In fact the current continuation need not be put back into the sequel if *next* does not call for a change of process, so it is more efficient to let *do* be

$\lambda\iota\theta\tau\sigma.(\lambda\tau'.\iota=next\,\iota\tau\sigma\rightarrow\theta\tau\sigma,(\tau(next\,\iota\tau\sigma)\downarrow7|C)\tau'\sigma)$

$\quad(impose\,\iota\langle\tau\iota\downarrow1,\tau\iota\downarrow2,\tau\iota\downarrow3,\tau\iota\downarrow4,\tau\iota\downarrow5,\tau\iota\downarrow6,\theta\rangle\,\tau).$

Minor variants of this are equally plausible: we could, for instance, select the next process on the assumption that the sequel preserves the current continuation, thereby giving *do* the value

$\lambda\iota\theta\tau\sigma.(\lambda\tau'.\iota=next\,\iota\tau'\sigma\rightarrow\theta\tau\sigma,(\tau(next\,\iota\tau'\sigma)\downarrow7|C)\tau'\sigma)$

$\quad(impose\,\iota\langle\tau\iota\downarrow1,\tau\iota\downarrow2,\tau\iota\downarrow3,\tau\iota\downarrow4,\tau\iota\downarrow5,\tau\iota\downarrow6,\theta\rangle\,\tau).$

To extend the formalism of 1.3.4 by introducing parallel processing we append extra arguments ι and τ to the equations of appendix 1 in the positions suggested by 3.4.2, and we then insert *do* at every basic step in a computation. As mentioned above, precisely what constitutes the set of such steps depends on the implementation; we shall include in it changes to sequels and stores (which provide paths of communication between processes), but doing so by no means exhausts it. The comments of 3.4.3 establish that associated with every process are members of U and Y, so when the environment is altered or when an expressed value is supplied to a continuation the sequel is implicitly being influenced. Hence that process ι to which a

given semantic equation refers can be supplanted by another process in the wake of evaluating a declaration or an expression. As a result $\mathscr{S}[\![I]\!]$ is $\lambda\rho\iota\kappa\tau\sigma.(\lambda\delta.\delta:G\rightarrow do\iota(\delta\kappa)\tau\sigma, do\iota(\kappa\delta)\tau\sigma)(\rho[\![I]\!]\downarrow 1)$ and $\mathscr{S}[\![\Phi]\!]$ is $\lambda\rho\iota\kappa\tau\sigma.do\iota(\kappa(\mathscr{S}[\![\Phi]\!]\rho))\tau\sigma$. Since $\mathscr{S}[\![E_0;\ E_1]\!]$ entails the deletion of an element from the stack we take it to be $\lambda\rho\iota\kappa.\mathscr{S}[\![E_0]\!]\rho\iota(\lambda\varepsilon.do\iota(\mathscr{S}[\![E_1]\!]\rho\iota\kappa))$ instead of $\lambda\rho\iota\kappa.\mathscr{S}[\![E_0]\!]\rho\iota(\lambda\varepsilon.\mathscr{S}[\![E_1]\!]\rho\iota\kappa)$; analogous remarks apply to conditional expressions and while loops.

The argument ι in the semantic equations does not represent the only process which is currently executing; in contrast to 3.4.2 here it is merely the process on which attention happens to be fixed at one particular moment. Accordingly ι will be said to be the 'present' process, while a 'current' process will be any process which could become the present process without impediment.

In order to clarify what constitutes such an impediment Mal will be extended by the addition of semaphores. Thus the expressions notice E and ignore E will determine whether or not the semaphore signified by E is to affect any future selection of the present process by *next*, while raise E and lower E will respectively add 1 to and subtract 1 from the value of the semaphore. As no process can even be current, far less present, unless the value of every semaphore influencing it exceeds 0, *next* must be suitably constrained. Consequently the information with which *next* is supplied must include a record of which semaphores can affect a given process. This is kept in the sequel τ:Q by letting $\tau\iota\downarrow 3$ be a list of the locations holding semaphores which have been made to influence ι; $\tau\iota\downarrow 1$ and $\tau\iota\downarrow 2$ remain truth values with much the same roles as in 3.4.2. The component $\tau\iota\downarrow 7$ contains a continuation, but the facilities for parallel programs to be provided in 3.5.3 will be such that this continuation need not be in C. Before describing these facilities, however, we must investigate the properties of *next*.

3.5.2. Scheduling algorithms.

The selection of the present process resembles that of a new location in that while it may depend on the particular implementation involved it is also subject to certain restrictions. These can be encapsulated in a continuous predicate, $able: I \to Q \to S \to T$, such that $able\iota\tau\sigma$ is *true* only if ι can currently be running; hence $\lambda\iota\tau\sigma.able(next\iota\tau\sigma)\tau\sigma = \lambda\iota\tau\sigma.\bigvee\{able\iota'\tau\sigma \mid \iota':I\} \to true, \bot$. For the language we are considering

$able = \lambda\iota\tau\sigma.\tau\iota\downarrow 1 \wedge \tau\iota\downarrow 2 \wedge \bigwedge\{\alpha:\tau\iota\downarrow 3 \to hold\alpha\sigma \mid N>0, true \mid \alpha:L\} \wedge (\tau\iota\downarrow 7:C^\circ)$

in accordance with 3.5.1, but here it is immaterial how the current processes are constrained. We are content merely to observe that if τ_1 and τ_2 are proper sequels such that $\tau_1 \sqcup \tau_2$ is proper then the continuity of *next* ensures that $\lambda\iota\sigma.next\iota\tau_1\sigma = \lambda\iota\sigma.next\iota\tau_2\sigma$. This apparently innocuous equality has disastrous implications which we shall now analyse.

It seems plausible that the values of $\mathscr{E}[\![B]\!]$ and $\mathscr{E}[\![\textbf{while } E_0 \textbf{ do } E_1]\!]$ should be taken to be $\lambda\rho\iota\kappa\tau\sigma.do\iota(\kappa(\mathscr{E}[\![B]\!]))\tau\sigma$ and $\lambda\rho\iota\kappa.fix(\lambda\theta.\mathscr{E}[\![E_0]\!]\rho\iota(\lambda\varepsilon.\varepsilon \to do\iota(\mathscr{E}[\![E_1]\!]\rho\iota(\lambda\varepsilon.do\iota\theta)), do\iota(\kappa(dummy))))$ respectively, so that $\mathscr{E}[\![\textbf{while true do dummy}]\!]$ must be $\lambda\rho\iota\kappa\tau\sigma.fix(do\iota)\tau\sigma$. Suppose that an attempt is made to execute a process ι_0 embodying the expression while true do dummy in parallel with another process and that by some chance when the execution of ι_0 commences the sequel τ_0 and the store σ_0 are such that $\tau_0\iota_0\downarrow 7:C^\circ$ and $next\iota_0\tau_0\sigma_0 = \iota_0$. The equality above implies that for all $n \geq 0$

$next\iota_0(impose\iota_0\langle \tau_0\iota_0\downarrow 1, \tau_0\iota_0\downarrow 2, \tau_0\iota_0\downarrow 3, \tau_0\iota_0\downarrow 4, \tau_0\iota_0\downarrow 5, \tau_0\iota_0\downarrow 6, \theta_n\rangle\tau_0)\sigma_0 = \iota_0$

where $\theta_0 = \bot$ and $\theta_{n+1} = do\iota_0\theta_n$ when $n \geq 0$. Consequently for each version of *do* suggested in 3.5.1 induction on $n \geq 0$ will establish that $\theta_n\tau_0\sigma_0 = \bot$ and thus that $fix(do\iota_0)\tau_0\sigma_0 = \bot$. Intuitively one would not expect this to be the case, for even when $next\iota_0\tau_0\sigma_0 = \iota_0$ the operating system should have the opportunity of breaking out of

the infinite loop by choosing another process after performing two indivisible operations taken from ι_0.

To remedy this situation it is necessary to provide *next* with a parameter the value of which will alter whenever *do* is invoked. Part of the outcome of applying *next* will have to be a new value of this parameter which will then be passed on in such a way that even if the present process, the sequel and the store are not changed successive calls of *next* need not produce the same result. This suggests that if H signifies this extra domain of parameters *next* should be sought among the members of $I \to H \to Q \to S \to [I \times H]$.

It would be possible to provide an all-embracing version of H by taking it to be a flat lattice; the elements of this could then be put in correspondence with those $\phi: I \to Q \to S \to [I \times H]$ such that $\lambda\iota\tau\sigma.\mathit{able}(\phi\iota\tau\sigma\downarrow 1)\tau\sigma = \lambda\iota\tau\sigma.\bigvee\{\mathit{able}\iota'\tau\sigma \mid \iota': I\} \to \mathit{true}, \bot$. Inherent in this interpretation of H are two more specialized (but equally satisfactory) ones, which will be termed 'descriptive' and 'prescriptive'. The first of these views elements of H as recording how the processes have been executed so far; thus I^* is appropriate as its H, $\lambda\iota^*.\#\iota^*$ is a rudimentary clock, and for all $\iota: I$, $\eta: H$, $\tau: Q$ and $\sigma: S$ $\mathit{next}\iota\eta\tau\sigma\downarrow 2$ can be taken to be either $\langle \mathit{next}\iota\eta\tau\sigma\downarrow 1\rangle \S\eta$ or $\langle \iota \rangle \S\eta$, depending on whether η is deemed to be amended when a process comes into play or when it is halted. In the prescriptive approach a typical $\eta: H$ provides the order in which processes will be executed in the future, so that under the convention of 1.2.8 H is I^∞ or possibly I^ν. There is no reason why H should not play both descriptive and prescriptive roles by assuming the form $I^* \times I^\infty$, but we shall concentrate on its prescriptive powers; however to stress the underlying generality of H we shall call a typical member of H a 'roster' instead of giving it some more evocative title. As intimated above we can

arrange for every descriptive roster to contain the name of the present process; we shall organize each prescriptive roster in a similar fashion by taking its top element to be the present process. This will enable us to discard ι (the present process) from the parameter lists of our equations, and to regard *next* as an element of H→Q→S→H; thus henceforth for any roster η in a suitable domain H η↓1 will be assumed to be the present process.

No matter how H is constructed *next* must be restricted by a postulate like

λητσ.*able*(*next*ητσ↓1)=λητσ.⋁{*able*ιτσ|ι:I}→*true*,⊥.

Consequently the next process to be executed cannot be selected from a prescriptive roster η simply by taking the first available element, which is η↓2↓1; rather *next* must trace back through η until it encounters a suitable process. This can be achieved by writing

next=λητσ.*able*(η↓2↓1)τσ→η↓2,*next*(η↓2)τσ.

Not every sensible scheduling policy can be viewed as a member of I* from which this particular *next* function selects a succession of processes. One policy which requires a different *next* function is that which demands that execution of the present process continue for as long as possible; for this we might set

next=λητσ.*able*(η↓1)τσ→η,*next*(η↓2)τσ.

Further policies are conceivable: one could, for instance, insist that the siblings of the current process be completely executed before any other process takes control. Thus to retain full generality either we must let *next* be any continuous function subject to the postulate above, perhaps together with

λητσ.*fix*(λφη'.*next*ητσ=η'→*true*,φ(η'↓2))η=λητσ.(*next*ητσ↓1:I),

or we must adopt the more general treatment mentioned above in which H is a flat lattice.

Prescriptive rosters perhaps provide the most natural

framework in which to discuss such problems as whether two processes deadlock or how to ensure that a program is allowed to execute only for a certain length of time: the latter issue, for instance, can be resolved by using a member of I^{∞} in which only the operating system processes occur infinitely often. In this connection it should be noted that a prescriptive roster in I^{∞} corresponds with a time-slicing algorithm only if the computer which is being modelled mathematically has precisely one processor; under other circumstances the roster is concerned to arbitrate between all the processes which are currently running by deeming one of them to be the present process. Thus it might be preferable to adopt $I^{*\infty}$ instead of I^{∞} so that for any $\eta:H$ $\eta\downarrow 1$ would be the list of all the processes which perform some action 'during a particular cycle'. Within this cycle the first remaining operation of one process, $\eta\downarrow 1\downarrow\nu_0$, would be applied before that of another process, $\eta\downarrow 1\downarrow\nu_1$, only if $\nu_0<\nu_1$; at the end of the cycle *next* would replace $\eta\downarrow 1$ with a different list, which might be $\eta\downarrow 2\downarrow 1$. This approach will not be pursued here, as it yields little extra insight.

Whichever sort of roster is employed the lattice of command continuations, C, is $H\to Q\to S\to A$ and the activities set in motion by choosing a process can be embodied in a suitable function *do*. When $\eta\downarrow 1$ signifies the present process and ψ is a variable ranging over a continuation domain having C° as a summand this function is given by

$$do=\lambda\psi\eta\tau\sigma.(\lambda\tau'.(\lambda\eta'.\eta\downarrow 1=\eta'\downarrow 1\to(\psi|C)\eta'\tau\sigma,(\tau(\eta'\downarrow 1)\downarrow 7|C)\eta'\tau'\sigma)(next\eta\tau'\sigma))$$
$$((\lambda\iota.imposer\langle\tau\iota\downarrow 1,\tau\iota\downarrow 2,\tau\iota\downarrow 3,\tau\iota\downarrow 4,\tau\iota\downarrow 5,\tau\iota\downarrow 6,\psi\rangle\tau)(\eta\downarrow 1)).$$

Under this definition of *do*, should $\theta_0:C$, $\eta_0:H$, $\tau_0:Q$ and $\sigma_0:S$ satisfy $(\tau_0(\eta_0\downarrow 1)\downarrow 7:C^{\circ})\vee\vee\{able\,\iota\tau_0\sigma_0|\iota:I\}=false$ then $do\theta_0\eta_0\tau_0\sigma_0$ need not yield a faulty computation with answer τ but instead

may be $\theta_0\eta_1\tau_0\sigma_0$ (for a suitable η_1), whereas setting

$do=\lambda\psi\eta\tau\sigma.(\lambda\tau'.(\lambda\eta'.\eta\downarrow 1=\eta'\downarrow 1\rightarrow(\psi|C)\eta'\tau\sigma,(\tau(\eta\downarrow 1)\downarrow 7|C)\eta'\tau'\sigma)(next\eta\tau\sigma))$

$((\lambda\iota.impose\iota\langle\tau\iota\downarrow 1,\tau\iota\downarrow 2,\tau\iota\downarrow 3,\tau\iota\downarrow 4,\tau\iota\downarrow 5,\tau\iota\downarrow 6,\psi\rangle\tau)(\eta\downarrow 1))$

must produce an error stop when $do\theta_0\eta_0\tau_0\sigma_0$ is encountered.

Besides being suitable for describing parallel processes which are explicitly set up by programs such as operating systems sequels can also provide an account of relevant actions external to the computer on which the program is executed. These actions might include supplying an input tape, which will be discussed in 3.5.3, or, more drastically, turning off the power. The latter case merely needs an appropriate function $halt:C^\circ$ attached by the sequel to a process $\iota_0:I$, together with a version of *next* which applies to prescriptive rosters and which satisfies $next\eta_0\tau_0\sigma_0\downarrow 1=\iota$ whenever $\eta_0:H$, $\tau_0:Q$ and $\sigma_0:S$ are such that $able\iota_0\tau_0\sigma_0=true$, $\eta_0\downarrow 1=\iota_0$ and $\tau_0\iota_0\downarrow 7=halt$.

An alternative way to represent some external actions entails taking H to be $I^\infty\times[Q\rightarrow S\rightarrow[Q\times S]]^\infty$ rather than I^∞. The function specified by the top element of the second component of $\eta:H$ is applied before a fresh process is selected by *next*, so *do* is

$\lambda\psi\eta\tau\sigma.(\lambda\langle\tau',\sigma'\rangle.(\lambda\iota.(\lambda\tau''.(\lambda\eta'.\iota=\eta'\downarrow 1\downarrow 1\rightarrow(\psi|C)\eta'\tau'\sigma',$

$(\tau'(\eta'\downarrow 1\downarrow 1)\downarrow 7|C)\eta'\tau''\sigma')$

$(next\eta\tau''\sigma'))$

$(impose\iota\langle\tau'\iota\downarrow 1,\tau'\iota\downarrow 2,\tau'\iota\downarrow 3,\tau'\iota\downarrow 4,\tau'\iota\downarrow 5,\tau'\iota\downarrow 6,\psi\rangle\tau')$

$(\eta\downarrow 1\downarrow 1))$

$((\eta\downarrow 2\downarrow 1)\tau\sigma)$

and $\lambda\eta\tau\sigma.next\eta\tau\sigma\downarrow 2$ is probably $\lambda\eta\tau\sigma.\eta\downarrow 2\downarrow 2$. As only certain members of $Q\rightarrow S\rightarrow[Q\times S]$ are likely to arise the component $[Q\rightarrow S\rightarrow[Q\times S]]^\infty$ could be replaced by R^∞ where R is a flat lattice of representations on which can be defined a mapping on to those $\phi:Q\rightarrow S\rightarrow[Q\times S]$ obeying suitable constraints. Among these constraints might be

$\lambda\alpha\tau\sigma.hold\alpha\sigma=\lambda\alpha\tau\sigma.hold\alpha(\phi\tau\sigma\downarrow 2)$, which would ensure that the only parts of the store to be altered would be the input and output buffers. We shall not pursue this topic further, since an adequate description of parallel programs seems to be provided by taking H to be I^{∞}. To confirm that this is so we shall now give a formal description of the language features mentioned in 3.5.1.

3.5.3. Typical equations for parallel processes.

Whereas the processes of 3.4.1 can be manipulated by textual statements those appropriate to parallel execution are not explicitly mentioned in a program. Consequently I is not usually part of the expressed value domain of a language; even conventional languages in which operating systems are written prefer to let summands like L masquerade as I. Thus when extended as in 3.5.1 Mal retains the lattices V=B+L*+J+F, E=L+V and D=E+G required by 1.4.5. Providing rosters in addition to sequels necessitates taking C to be H→Q→S→A (not Q→S→A or S→A), but K and X remain E→C and U→C respectively, so the valuations are exemplified by $\mathscr{E}$:Exp→U→K→C and by $\mathscr{D}$:Dec→U→X→C.

In accordance with the argument of 3.5.1 the semantic equations are built up by inserting *do* at all their basic steps. The roster η:H supplied to these equations serves solely to affect the outcome of *do* and to indicate which is the present process; even the sequel τ:Q affects only a few of the equations given below. Thus $\mathscr{E}[\![I]\!]$ is $\lambda\rho\kappa\eta\tau\sigma.(\lambda\delta.\delta:G\rightarrow do(\delta\kappa)\eta\tau\sigma, do(\kappa\delta)\eta\tau\sigma)(\rho[\![I]\!]\downarrow 1)$ (and G is $[K\rightarrow C]^{\circ}$), while $\mathscr{E}[\![B]\!]$ and $\mathscr{E}[\![\Phi]\!]$ are $\lambda\rho\kappa\eta\tau\sigma.do(\kappa(\mathscr{B}[\![B]\!]))\eta\tau\sigma$ and $\lambda\rho\kappa\eta\tau\sigma.do(\kappa(\mathscr{F}[\![\Phi]\!]\rho))\eta\tau\sigma$ respectively. Since ι:I need no longer appear as a parameter in the equations, $\mathscr{E}[\![E_0;\ E_1]\!]$ is $\lambda\rho\kappa.\mathscr{E}[\![E_0]\!]\rho(\lambda\varepsilon.do(\mathscr{E}[\![E_1]\!]\rho\kappa))$, $\mathscr{E}[\![\text{if } E_0 \text{ then } E_1 \text{ else } E_2]\!]$ is $\lambda\rho\kappa.\mathscr{R}[\![E_0]\!]\rho(\lambda\varepsilon.\varepsilon\rightarrow do(\mathscr{E}[\![E_1]\!]\rho\kappa), do(\mathscr{E}[\![E_2]\!]\rho\kappa))$ and $\mathscr{E}[\![\text{while } E_0 \text{ do } E_1]\!]$

is $\lambda\rho\kappa.fix(\lambda\theta.\mathscr{R}[\![E_0]\!]\rho(\lambda\varepsilon.\varepsilon\rightarrow do(\mathscr{G}[\![E_1]\!]\rho(\lambda\varepsilon.do\theta)),do(\kappa(dummy))))$; now, however, the definition of *do* given in 3.5.2 ensures that the execution of while true do dummy need not continue indefinitely if two of its basic steps are performed in quick succession. Just as alterations to the stack influence the sequel implicitly so do alterations to the environment; hence $\mathscr{G}[\![\Delta$ inside $E]\!]$ is taken to be $\lambda\rho\kappa.\mathscr{G}[\![\Delta]\!]\rho(\lambda\rho'.do(\mathscr{L}[\![E]\!](divert\rho\rho')\kappa))$. By the same token $\mathscr{D}[\![I==E]\!]$ and $\mathscr{T}[\![I==E]\!]$ are both equal to $\lambda\rho\chi.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon.do(\chi(arid[\varepsilon/I])))$, while $\mathscr{D}[\![I=E]\!]$ is $\lambda\rho\chi.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon.do(\chi(arid[\varepsilon/I])))$ and $\mathscr{T}[\![I=E]\!]$ is $\lambda\rho\chi.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon\eta\tau\sigma.(\lambda\delta.\delta:L\rightarrow do(\chi(arid[\delta/I]))\eta\tau(update\delta\varepsilon\sigma),\tau)(\rho[\![I]\!]\downarrow 1))$.

For any $\tau:Q$ and $\iota:I$ $\tau\iota\downarrow 3$ is a member of L^* which lists the semaphores which help to determine whether ι can be used as the next process. Thus $\mathscr{G}[\![$notice $E]\!]$, which is intended to adjoin the location represented by E to the member of L^* corresponding to the present process, is simply $\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\eta\tau.(\lambda\iota.do(\kappa\varepsilon)\eta(impose\iota(\langle\tau\iota\downarrow 1,\tau\iota\downarrow 2,\langle\varepsilon\rangle\S\tau\iota\downarrow 3\rangle\S\tau\iota\dagger 3)\tau))(\eta\downarrow 1))$ whilst $\mathscr{G}[\![$ignore $E]\!]$, which is intended to delete the location from the list is

$$\lambda\rho\kappa.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon\eta\tau.(\lambda\alpha^*.(\lambda\iota.do(\kappa\varepsilon)\eta(impose\iota(\langle\tau\iota\downarrow 1,\tau\iota\downarrow 2,\alpha^*\rangle\S\tau\iota\dagger 3)\tau))(\eta\downarrow 1))$$
$$(fix(\lambda\phi\beta.\beta=\langle\rangle\rightarrow\langle\rangle,(\varepsilon=\beta\downarrow 1\rightarrow\langle\rangle,\langle\beta\downarrow 1\rangle)\S\phi(\beta\dagger 1))(\tau(\eta\downarrow 1)\downarrow 3))).$$

The operation of increasing (or of decreasing) the value of a semaphore is indivisible but that of evaluating the expression E which gives rise to it need not be; this fact is incorporated in the semantic equations by letting $\mathscr{G}[\![$raise $E]\!]$ and $\mathscr{G}[\![$lower $E]\!]$ be $\lambda\rho\kappa.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon\eta\tau\sigma.do(\kappa\varepsilon)\eta\tau(update\varepsilon(hold\varepsilon\sigma|N+1)\sigma))$ and $\lambda\rho\kappa.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon\eta\tau\sigma.do(\kappa\varepsilon)\eta\tau(update(hold\varepsilon\sigma|N-1)\sigma))$, in which no opportunity is given to carry out any other actions between isolating ε and modifying its content.

Although the function *run* set up in 1.3.5 allows expressions

and declarations to be evaluated in an indeterminate order, it does not permit one evaluation to be entangled with another. Thus according to appendix 1 the execution of $E_1,\ldots,E_n$ entails executing E_1 either before E_2 or after E_2, so that switching back and forth between E_1 and E_2 is not envisaged. Now that the formalism can handle parallel programs this can be viewed in a fresh light Rather than reflecting a permutation of a list of expressions or declarations *run* will set up processes which will evaluate the constituents of the list. The answers calculated will then be returned to the parent process ι so that it can resume execution. While its offspring are running the continuation ψ attached to ι by the sequel will therefore be in E*→C or U*→C, so that the definition of *able* in 3.5.2 will preclude the possibility that ι could proceed in parallel with its descendants. This situation does not prevail in all languages and is not typical of operating systems, which can be executed at the same time as their offspring, but its variants could obviously be modelled equally well. For Mal, however, it is necessary merely to introduce the functions *run*:[K→C]°*→[E*→C]→C and *run*:[X→C]°*→[U*→C]→C. When H is taken to be I" and when $\eta\downarrow 1$ is always regarded as the present process corresponding to η:H, for every γ*:[K→C]°*, ψ:[E*→C]°, η:H, τ:Q and σ:S *run*γ*$\psi\eta\tau\sigma$ equals

$$(\lambda\iota^*.(\lambda\xi.(\lambda\tau'.do\psi\eta\tau'\sigma)(fix(\lambda\phi\nu.\nu>\#\gamma^*\rightarrow\tau,impose(\iota^*\downarrow\nu)(\xi\nu)(\phi(\nu+1)))1))$$
$$(\lambda\nu.(\lambda\kappa.\langle true,true,\tau(\eta\downarrow 1)\downarrow 3,\iota^*,\tau(\eta\downarrow 1)\downarrow 5,\tau(\eta\downarrow 1)\downarrow 6,(\gamma^*\downarrow\nu)\kappa\rangle)$$
$$(\lambda\varepsilon\eta'\tau'\sigma'.(\lambda\tau".(\lambda\psi'.(\lambda\beta.do(\beta\rightarrow\psi',\psi'(dummy^*))\langle\eta\downarrow 1,\eta'\rangle\tau'\sigma')$$
$$(\bigvee\{\tau"(\iota^*\downarrow\nu')\downarrow 1\mid 1\leq\nu'\leq\#\gamma^*\}))$$
$$(\lambda\varepsilon^*.(\tau"\iota\downarrow 7)(\langle\varepsilon^*\downarrow 1,\ldots,\varepsilon^*\downarrow(\nu-1),\varepsilon\rangle\,\S\varepsilon^*\dagger\nu))$$
$$(impose(\iota^*\downarrow\nu)(\langle false\rangle\,\S\tau'(\iota^*\downarrow\nu)\dagger 1)\tau'))))$$
$$(nears(\#\gamma^*)\tau\sigma).$$

Unappealing though this formula is, it does at least have the merit of being adequate to describe a realistic language feature; moreover it is sufficiently general for all our present purposes, in that

the definition of *run* appropriate to declarations differs only in the domains required by γ^* and ψ.

The complicated nature of *run* is inevitable, for it is designed to embody a rather devious course of action. On embarking upon the evaluation of $E_1,\ldots,E_n$, say, *run* not only preserves the given $\psi:[E^*\rightarrow C]^\circ$ as the continuation of $\eta\downarrow 1$, the present process, but also obtains n new processes using *near*, which obeys an axiom like that of 3.4.2, and endows them with fresh entries in the sequel; the roster could, of course, also be altered but this would be superfluous. Each fresh entry inherits $\tau(\eta\downarrow 1)\downarrow 3$ (the list of semaphores attached to $\eta\downarrow 1$) and catalogues the siblings of the relevant process as its fourth component; any one of these siblings may be invoked by *do* because they all have continuations belonging to C. These continuations incorporate the code for some E_m having $1\leq m\leq n$ as well as some $\kappa:K^\circ$ which describes what is to happen once the evaluation of E_m is complete. In fact the effect of κ depends on whether any of the siblings of the process corresponding to E_m have not yet finished executing: if $\vee\{\tau''(\iota^*\downarrow\nu')\downarrow 1 \mid 1\leq\nu'\leq\#\gamma^*\}$ is *true* there are siblings which must proceed further so the result returned by E_m is given to the continuation of $\eta\downarrow 1$ but this continuation remains in $[E^*\rightarrow C]$, whereas if $\vee\{\tau''(\iota^*\downarrow\nu') \mid 1\leq\nu'\leq\#\gamma^*\}$ is *false* the continuation of $\eta\downarrow 1$ is put into C by supplying it with a dummy argument list of length n. Thus after all the E_m have been exhausted none of the descendants of $\eta\downarrow 1$ can be called upon by *next* to become the present process since each has *false* as the first component of entry in the sequel; $\eta\downarrow 1$, on the other hand, can resume control because its continuation is a member of C. Much of the clumsiness in the definition of *run* stems from the need to return the result of E_m to $\eta\downarrow 1$ immediately it has been found, so a simpler function could be used were the sequel to follow the queue of 3.4.3 by

keeping explicit environments and stacks.

When parallel programming is introduced the value of $\mathscr{E}[\![E_1,\ldots,E_n]\!]$ given in appendix 1 can therefore be replaced by $\lambda\rho\kappa\eta\tau\sigma.run\langle\mathscr{L}[\![E_1]\!]\rho,\ldots,\mathscr{L}[\![E_n]\!]\rho\rangle(\lambda\varepsilon^*.\kappa(\varepsilon^*|E))\eta\tau\sigma$, while $\mathscr{D}[\![\Delta_1$ and...and $\Delta_n]\!]$ and $\mathscr{T}[\![\Delta_1$ and...and $\Delta_n]\!]$ can be assumed to be $\lambda\rho\chi\eta\tau\sigma.run\langle\mathscr{D}[\![\Delta_1]\!]\rho,\ldots,\mathscr{D}[\![\Delta_n]\!]\rho\rangle(\chi\circ conserve)\eta\tau\sigma$ and $\lambda\rho\chi\eta\tau\sigma.run\langle\mathscr{T}[\![\Delta_1]\!]\rho,\ldots,\mathscr{T}[\![\Delta_n]\!]\rho\rangle(\chi\circ conserve)\eta\tau\sigma$ respectively. Likewise $\mathscr{E}[\![E_0:=E_1]\!]$ becomes

$\lambda\rho\kappa.run\langle\mathscr{L}[\![E_0]\!]\rho,\mathscr{R}[\![E_1]\!]\rho\rangle(\lambda\varepsilon^*\eta\tau\sigma.do(\kappa(dummy))\eta\tau(update(\varepsilon^*\downarrow 1)(\varepsilon^*\downarrow 2)\sigma))$.

The siblings of a process are listed in the sequel to avoid executing two parts of a program in parallel unless the syntax indicates that this should happen. Sequential standard semantics suggests that 3 is the outcome of val (0+1,(2,res 3)); however unless the process executing 0+1 is terminated on jumping out of res 3 parallel semantics may not reach the same conclusion. Loosely speaking, were this process allowed to stay active after the jump the stack associated with its parent might grow to the wrong size; this phenomenon would be expressed in the formalism by an attempt to apply a member of C to an argument in E. Consequently whenever a process is left by jumping the descendants of its siblings must be terminated using

$hang=\lambda\iota^*\tau.\iota^*=\langle\rangle\rightarrow\tau,$

$(\lambda\iota.hang(\iota^*\dagger 1)(hang(\tau\iota\downarrow 4)(impose\iota(\langle false\rangle\,\S\tau\iota\dagger 1)\tau)))(\iota^*\downarrow 1)$.

Thus in parallel semantics $\mathscr{E}[\![$val E$]\!]$ is taken to be $\lambda\rho\kappa\eta.\mathscr{L}[\![E]\!]\rho[\lambda\varepsilon\eta'\tau'\sigma'.do(\kappa\varepsilon)\langle\eta\downarrow 1,\eta'\rangle(\eta\downarrow 1=\eta'\downarrow 1\rightarrow\tau',hang\langle\eta'\downarrow 1\rangle\tau')\sigma'/res]\kappa\eta$ and $\mathscr{E}[\![$res E$]\!]$ is taken to be $\lambda\rho\kappa.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon.do(\rho[\![res]\!]\varepsilon))$. The analagous equations for labels require knowledge of the prescriptive roster pertaining at the time the labels are declared in order to identify the present process. Hence $\mathscr{P}[\![I:E]\!]\rho\kappa\eta\downarrow 1$ and $\mathscr{L}[\![I::E]\!]\rho\kappa\eta\downarrow 1$ are both

$\lambda\eta'\tau'\sigma'.do(\mathscr{G}[\![E]\!]\rho\kappa)(\eta\downarrow 1,\eta')(\eta\downarrow 1=\eta'\downarrow 1\rightarrow\tau',hang(\eta'\downarrow 1)\tau')\sigma'$,

although $\mathscr{G}[\![$goto E$]\!]$ is simply $\lambda\rho\kappa.\mathscr{A}[\![E]\!]\rho(\lambda\varepsilon.do(\varepsilon|C))$ and J is C°. Slightly different formulae would be necessary were the present process treated as an additional parameter of the semantic equations rather than as a distinguished part of the roster.

By contrast with the situation for labels the ostensible meaning of an abstraction in parallel standard semantics closely resembles that given by sequential semantics. Thus $\mathscr{F}[\![$fn()E$]\!]$, for instance, is $\lambda\alpha\kappa.rv(\lambda\beta.\beta|L^*=\langle\rangle\rightarrow do(\mathscr{L}[\![E]\!]\rho\kappa),\tau)\alpha$ and $\mathscr{G}[\![E_0E_1]\!]$ is

$\lambda\rho\kappa.(\lambda\psi.run\langle\mathscr{L}[\![E_0]\!]\rho,\mathscr{R}[\![E_1]\!]\rho\rangle\psi)$

$(\lambda\varepsilon^*.\varepsilon^*\downarrow 1:F\rightarrow do(\varepsilon^*\downarrow 1)(\varepsilon^*\downarrow 2)\kappa,$

$rv(\lambda\beta.1\leq\beta|N\leq\#\varepsilon^*\downarrow 1|L^*\rightarrow do(\kappa(\varepsilon^*\downarrow 1\downarrow\beta)),\tau)(\varepsilon^*\downarrow 2))$

so long as *rv* and *sv* are modified to take account of the presence of H and Q.

In a computer which permits concurrent operations it is natural to view locations as being accessible to particular processes instead of to the entire computation. Consequently an element of $[L\rightarrow T]$ must be attached to every process in the sequel. Actually two such elements are attached in order to segregate the heap storage heaving infinite extent from the stack storage, which is discarded on leaving the block wherein it is allocated; Q is therefore taken to be

$I\rightarrow[T\times T\times L^*\times I^*\times[L\rightarrow T]\times[L\rightarrow T]\times[C^{\circ}+[E^*\rightarrow C]^{\circ}+[U^*\rightarrow C]^{\circ}]]$. By introducing further syntax we could ensure that certain locations required by a Mal program would be retrieved not by garbage collection but with the aid of

$$restore=\lambda\tau_0\tau_1.\lambda\iota.(\lambda\varepsilon.\langle\tau_1\iota\downarrow 1,\tau_1\iota\downarrow 2,\tau_1\iota\downarrow 3,\tau_1\iota\downarrow 4,\varepsilon,\tau_1\iota\downarrow 6,\tau_1\iota\downarrow 7\rangle)$$
$$(\lambda\alpha.(\tau_0\iota\downarrow 5)\alpha\wedge(\tau_1\iota\downarrow 5)\alpha).$$

Just as this replaces the function *restore* of 3.1.1 so the stipulation on *new* provided by 1.3.1 is superseded by one

concerning *new*:H→Q→S→L, to wit

$\lambda\eta\tau\sigma.\bigvee\{((\tau\iota\downarrow 5)(new\eta\tau\sigma)\vee(\tau\iota\downarrow 6)(new\eta\tau\sigma))\wedge\tau\iota\downarrow 1 \mid \iota:I\}$

$=\lambda\eta\tau\sigma.\bigwedge\{\bigvee\{((\tau\iota\downarrow 5)\alpha\vee(\tau\iota\downarrow 6)\alpha)\wedge\tau\iota\downarrow 1 \mid \iota:I\} \mid \alpha:L\}\rightarrow\perp, false.$

An interesting demonstration of the interaction of a process with actions external to it is provided by an input buffer. Though S, the domain of stores, may be $[L\rightarrow V]\times V^*\times V^*$ the second component of a store σ may not list all the members of V which are ever going to be input. Thus an attempt to evaluate get E when $\sigma\downarrow 2$ is the empty list need not lead to an error; it may just halt the process until $\sigma\downarrow 2$ is not empty. This effect could be expressed by taking $\mathscr{G}[\![\text{get E}]\!]$ to be

$\lambda\rho\kappa.\mathscr{E}[\![E]\!]\rho(\lambda\varepsilon.fix(\lambda\theta\eta\tau\sigma.\#\sigma\downarrow 2>0\rightarrow\phi_0\kappa\varepsilon\eta\tau\sigma, do\,\theta\eta\tau\sigma))$ where

$\phi_0=\lambda\kappa\varepsilon\eta\tau\sigma.do(\kappa\varepsilon)\eta\tau(update\varepsilon(\sigma\downarrow 2\downarrow 1)\langle\sigma\downarrow 1,\sigma\downarrow 2\dagger 1,\sigma\downarrow 3\rangle)$, but this involves making the process to resume execution when it tests to see whether $\sigma\downarrow 2$ has been filled. A more efficient version of $\mathscr{G}[\![\text{get E}]\!]$ would be

$\lambda\rho\kappa.\mathscr{E}[\![E]\!]\rho(\lambda\varepsilon\eta\tau\sigma.(\lambda\tau'.\#\sigma\downarrow 2>0\rightarrow\phi_0\kappa\varepsilon\eta\tau\sigma, do(\phi_0\kappa\varepsilon)\eta\tau'\sigma)$

$(impose(\eta\downarrow 1)(\langle\tau(\eta\downarrow 1)\downarrow 1, false\rangle\,\S\,\tau(\eta\downarrow 1)\dagger 2)\tau));$

the second component of $\eta\downarrow 1$ would then need to be altered when the computer operator supplied more input. Though the person concerned might believe himself to have free-will, the sequence of values provided by him can be assumed in retrospect at least to be ω_0, a predetermined member of V^* (which is defined in 1.2.8). The process ι_0 corresponding in τ to his actions would therefore have at its genesis a continuation θ_0 such that

$\theta_0=fix(\lambda\phi\omega\eta\tau\sigma.(\lambda\langle\tau',\sigma'\rangle.\omega=\langle\rangle\rightarrow halt\eta\tau\sigma, do(\phi(\omega\downarrow 2))\eta\tau'\sigma'),$

$\langle\lambda\iota.\langle\tau\iota\downarrow 1, true\rangle\,\S\,\tau\iota\dagger 2, \langle\sigma\downarrow 1,\sigma\downarrow 2\,\S\langle\omega\downarrow 1\rangle,\sigma\downarrow 3\rangle\rangle)\omega_0.$

This formulation is not entirely satisfactory because θ_0 is obliged to alter $\tau\iota\downarrow 2$ for every $\iota:I$, but more realistic equations could readily be given. Among these would be a group in which $\tau\iota\downarrow 2$

represented not a member of T but a member of I*; this member of I* would be the list of processes which could not become current until ι had carried out some action. Hence $\mathscr{G}[\![$get E$]\!]$ would be

$\lambda\rho\kappa.\mathscr{P}[\![E]\!]\rho(\lambda\varepsilon\eta\tau\sigma.(\lambda\tau'.\#\sigma\downarrow 2>0\rightarrow\phi_0\kappa\varepsilon\eta\tau\sigma,do(\phi_0\kappa\varepsilon)\eta\tau'\sigma)$

$(impose\iota_0(\langle\tau\iota_0\downarrow 1,\langle\eta\downarrow 1\rangle\S\tau\iota_0\downarrow 2\rangle\S\tau\iota_0\dagger 2)\tau))$

and θ_0 would be given by

$\theta_0=fix(\lambda\phi\omega\eta\tau\sigma.(\lambda\langle\tau',\sigma'\rangle\omega=\langle\rangle\rightarrow halt\eta\tau\sigma,do(\phi(\omega\downarrow 2))\eta\tau'\sigma')$

$\langle impose\iota_0(\langle\tau\iota_0\downarrow 1,\langle\rangle\rangle\S\tau\iota_0\dagger 2)\tau,\langle\sigma\downarrow 1,\sigma\downarrow 2\S\langle\omega\downarrow 2\rangle,\sigma\downarrow 3\rangle\rangle)\omega_0.$

A model in which $\tau\iota\downarrow 2$ belonged to I* might constrain the selection of the next process by

$able=\lambda\iota\tau\sigma.\tau\iota\downarrow 1\wedge\sim(\iota:\tau\iota_0\downarrow 2\vee\ldots\vee\iota:\tau\iota_m\downarrow 2)$

$\wedge\bigwedge\{\alpha:\tau\iota\downarrow 3\rightarrow hold\alpha\sigma\,|\,N>0,true\,|\,\alpha:L\}\wedge(\tau\iota\downarrow 7:C^\circ),$

where $\iota_0,\ldots,\iota_m$ are suitable operating system processes.

By following the lines of 3.4.3 we could introduce a form of parallel store semantics which would be equivalent to parallel standard semantics in the sense of 2.3.9 provided that *do* was inserted at the appropriate points. Our other theorems could also be established under the additional assumption that only one member of I can become the present process (which entails giving *run* the same meaning as in 1.3.5). Thus, for instance, all suitable ρ:U, η:H, τ:Q and E:Exp satisfy $\lambda\kappa\sigma.\mathscr{G}[\![E]\!]\rho\kappa\eta\tau\sigma=\lambda\kappa\sigma.((\lambda\varepsilon.do(\kappa\varepsilon)\eta\tau)*\P\mathscr{G}[\![E]\!]\rho)\sigma$ when 1.5.4 supplies the conjugate valuation.

There is no difficulty about setting up sequels which describe other features of parallel programming and 'real time' systems, but the domain Q tends to become complex. However the structure imposed above on Q is sufficiently large for our purposes, as it could be made applicable to Algol 68 with the aid of minor extensions. Chief among these would be the addition of components listing the opened and closed input and output streams attached to particular processes. We shall not

detail these but will end the section with an account of the conceptual basis for our treatment of input and output.

3.5.4. Computed state sequences.

It has been claimed that a deficiency of the theory as developed in appendix 1 is its inability to provide such programs as while true do put 1 with any answer other than ⊥. This can be overcome by interpreting programs as families of parallel processes obeying the equations of 3.5.3; here, however, without introducing rosters or sequels we shall try to analyse those computations which in sequential semantics yield the answer ⊥ and appear not to terminate.

There is a widespread belief that a program which never halts should be given a meaning by the semantic equations which reflects whether or not it prints any results. To achieve this we could set $S=[L\to[T\times V]]\times V^*$ and $A=V^\vee$ (taking $V^\vee$ to be as in 1.2.8), furnish every entire program with the final continuation $\lambda\varepsilon\sigma.\langle\rangle\,|A$, and view $\mathscr{C}[\![\text{put } E]\!]$ as $\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\sigma.\langle\varepsilon,\kappa\varepsilon\sigma\rangle)$ instead of $\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\sigma.\kappa\varepsilon\langle\sigma\downarrow 1,\sigma\downarrow 2,\langle\varepsilon\rangle\,\S\sigma\downarrow 3\rangle)$. When the remainder of appendix 1 is preserved intact this device makes while true do put 1 and put 0 have as their respective outcomes $fix(\lambda o.\langle 1,o\rangle)$ (an unending stream of numbers) and $\langle 0,\langle\rangle\rangle$. Moreover at the cost of specifying the domain A the output channel would be removed from the store, where it gives a misleading impression of influencing the future course of the computation by being supplied as an argument to the continuation. Nevertheless modifying $\mathscr{C}[\![\text{put } E]\!]$ has implications which the considerations below lead us to regard as being both too extensive and too confined.

In accordance with the precepts of 1.4.1 we wish the side effects of a recursive declaration by incidence to be performed

at the time of declaration only. However were we to adopt the modification suggested above the operators of appendix 1 would give to rec f==(put 1; fn$z$.put 2; fz)) inside $f0$ the value $\langle 1,fix(\lambda o.\langle 1,\langle 2,o\rangle\rangle)\rangle$ instead of the more fitting $\langle 1,fix(\lambda o.\langle 2,o\rangle)\rangle$. Unfortunately there is no natural way to amend our treatment of recursion to make it compatible with an approach which binds the output more tightly to the continuation than to the store.

Perhaps it is unfair to berate one artifice on the ground that it invalidates another, but there are in any case further unsatisfactory aspects to every model which preserves only the output from computations. Since the input is not saved in a similar fashion, bilateral streams (which can be both written on and read) must be handled by the formalism in the manner chosen for the input streams, although they partake equally of the nature of output streams. In Algol 68 there is a library function which will test any stream to see whether it can be written on; as streams are not tied to particular mechanical devices for transmitting information, we can readily conceive of another function which would transform an output stream into a bilateral stream. Such a function would play havoc with the semantics of output proposed above, but within the more conventional model of 1.3.1 (extended by the means mentioned in 3.5.3) it would simply confirm the essential uniformity of streams: each is an object not unlike a process or a location containing a list in V* or H* together with certain truth values which determine such trivial matters as whether output can be sent along it.

The argument for distinguishing the outcome of while true do dummy from that of while true do put 1 but not from that of while true do 0:=1 seems to rest upon the assumption that sending something along an output stream is irrevocable whereas assignments to locations can be overwritten. This confuses

physical reality with mathematical calculations involving lattices. Certainly the marks made upon line-printer paper may be indelible and the holes punched in tapes cannot be filled, but this hardly seems to bear upon the algorithm realized by a program. Indeed for such devices as traffic lights the physical manifestations of output are purely ephemeral and can be interpreted equally well as the updating of locations. It is no more satisfactory to argue that the difference between the output and the store resides in the irreversibility of the act of sending a number along an output stream, for the act of assignment likewise cannot be undone: once we have updated a location we do not annul the fact that an assignment has taken place when we later restore the content to its original value. Furthermore the significant products of non-terminating programs like operating systems are sequences of acts with evanescent effects, and whether or not to construe them as 'output' is a matter of taste.

Given that we ought to distinguish the outcome of while true do dummy from that of an unending loop which writes numbers there can be little sense in identifying it with the outcome of such an expression as while true do get 1. If the output is not to vanish when a program fails to terminate then neither should the input, for it cannot be affected at all by the algorithm which uses it. In fact whereas the infinite list of numerals which is printed might reasonably be thought to deserve the formal interpretation $\perp$ like the store, the input stream should not be violated by the computation. Accordingly our earlier suggestion must give way to one in which A, the domain of answers, is $V^* \times V^*$ and the final continuation is $\lambda\varepsilon\sigma.\langle\langle\rangle|A,\langle\rangle|A\rangle$. Now we take $\mathscr{C}[\![\text{put } E]\!]$ to be $\lambda\rho\kappa.\mathscr{E}[\![E]\!]\rho(\lambda\varepsilon\sigma.\langle\kappa\varepsilon\sigma\downarrow 1,\langle\varepsilon,\kappa\varepsilon\sigma\downarrow 2\rangle\rangle)$ and $\mathscr{C}[\![\text{get } E]\!]$ to be

$\lambda\rho\kappa.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon\sigma.(\lambda\sigma'.\langle\langle\sigma\downarrow 2\downarrow 1, \kappa\varepsilon\sigma'\downarrow 1\rangle, \kappa\varepsilon\sigma'\downarrow 2\rangle)$

$(update\varepsilon(\sigma\downarrow 2\downarrow 1)\langle\sigma\downarrow 1, \sigma\downarrow 2\dagger 1\rangle)).$

We cannot keep merely the residual dregs of the input by letting $\mathscr{S}[\![$ get E$]\!]$ be

$\lambda\rho\kappa.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon\sigma.(\lambda\sigma'.\langle\kappa\varepsilon\sigma'\downarrow 1\downarrow 2, \kappa\varepsilon\sigma'\downarrow 2\rangle)(update\varepsilon(\kappa\varepsilon\sigma\downarrow 1\downarrow 1)\sigma)),$

because this would provide while true do get 1 with the answer and would make a mockery of the intention of continuations. Yet once we have changed $\mathscr{S}[\![$ get E$]\!]$ symmetry dictates that we turn $\mathscr{S}[\![$ put E$]\!]$ into the analogous form, which is

$\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\sigma.(\lambda\sigma'.\langle\kappa\varepsilon\sigma'\downarrow 1, \langle\varepsilon, \kappa\varepsilon\sigma'\downarrow 2\rangle\rangle)\langle\sigma\downarrow 1, \sigma\downarrow 2, \sigma\downarrow 3\dagger 1\rangle)$

where $S=[L\mapsto[T\times V]]\times V^*\times V^*$ as before.

Having introduced the principle that the semantic equations can duplicate an entity (in our case the input) we can grasp what underlies the original modification to the meaning of $\mathscr{S}[\![$ put E$]\!]$: essentially it is concerned not to remove the output from a realm where it does not belong but to keep a record of some of the actions indulged in by the program. There is no reason why other actions might not be of equal interest; we might, for instance, wish to examine the way in which the content of a particular location changes as the computation proceeds. In the situations above we are simply inspecting what happens at some of the basic steps (in the sense of 3.5.1) and ignoring what happens at others. Considerations of elegance and consistency suggest that instead of doing this we preserve all the changes to the store by setting $S=[L\mapsto[T\times V]]\times V^*\times V^*$ and $A=S^\vee$. Each program is supplied with the final continuation $\lambda\varepsilon\sigma.\langle\sigma, \langle\rangle \mid A\rangle$ and at every basic step we adjoin the current store to the list in A. Accordingly $\mathscr{S}[\![E_0:=E_1]\!]$ is now

$\lambda\rho\kappa.run\langle\mathscr{L}[\![E_0]\!]\rho, \mathscr{R}[\![E_1]\!]\rho\rangle(\lambda\varepsilon^*\sigma.\langle\sigma, \kappa(dummy)(update(\varepsilon^*\downarrow 1)(\varepsilon^*\downarrow 2)\sigma)\rangle),$

$\mathscr{S}[\![E_0;\ E_1]\!]$ is $\lambda\rho\kappa.\mathscr{S}[\![E_0]\!]\rho(\lambda\varepsilon\sigma.\langle\sigma, \mathscr{S}[\![E_1]\!]\rho\kappa\sigma\rangle)$ and $\mathscr{S}[\![I]\!]$ is

$\lambda\rho\kappa\sigma.(\lambda\delta.\delta:G\rightarrow\langle\sigma,\delta\kappa\sigma\rangle,\langle\sigma,\kappa\delta\sigma\rangle)(\rho[\![I]\!]\downarrow 1)$; more generally, wherever *do* would appear in collateral semantics this sort of equation inserts a further store instead. Obvious retractions enable us to extract the final output and input from the store sequence arrived at by the end of the computation, thereby providing all the information obtainable from the earlier versions of the equations for get E and put E.

Needless to say there are many variants of the scheme proposed here; in particular both store and stack semantics would allow the retention of a sequence of state vectors in the domain $A=[U\times Y\times S]^*$, while their parallel counterparts might give rise to members of $A=[H\times Q\times S]^*$. Moreover it is even possible to adapt the scheme in order to capture the execution sequence when semantic equations without continuations are provided. It seems that whatever formulation of semantics is appropriate to the problem in hand should be used so long as it can be proved to be equivalent to the standard semantics of 1.3.4 (enhanced as in 3.5.3 when parallel programming is permitted). In the present case such a proof can be couched in terms of $halt:A\rightarrow S$, which is defined by $halt=\lambda o.o\downarrow 2=()\rightarrow o\downarrow 1,halt(o\downarrow 2|A)$. We take the domain A of appendix 1 to be S and regard a typical pair $\hat{o}$ as having $\acute{o}$ drawn from standard semantics and $\grave{o}$ drawn from the above. If $c=\lambda\hat{\theta}.\bigwedge\{a\langle\acute{\theta}\acute{\sigma},\grave{\theta}\grave{\sigma}\rangle\mid s\theta\}$, $k=\lambda\hat{\kappa}.\bigwedge\{c\langle\acute{\kappa}\acute{\epsilon},\grave{\kappa}\grave{\epsilon}\rangle\mid e\hat{\epsilon}\}$, $x=\lambda\hat{\chi}.\bigwedge\{c\langle\acute{\chi}\acute{\rho},\grave{\chi}\grave{\rho}\rangle\mid u\hat{\rho}\}$ and $a=\lambda\delta.(\lambda\theta.\theta=\langle\perp,\perp\rangle\vee\theta=\langle\top,\top\rangle\rightarrow true,s\theta)\langle\acute{\delta},halt\grave{\delta}\rangle$ (and if s has the same form as in 2.2.5) we can assert that for every E:Exp and Δ:Dec $c\langle\mathscr{E}[\![E]\!]\acute{\rho}\acute{\kappa},\mathscr{E}[\![E]\!]\grave{\rho}\grave{\kappa}\rangle\wedge c\langle\mathscr{D}[\![\Delta]\!]\acute{\rho}\acute{\chi},\mathscr{D}[\![\Delta]\!]\grave{\rho}\grave{\chi}\rangle=true$ for all $\hat{\rho}$, $\hat{\kappa}$ and $\hat{\chi}$ having $u\hat{\rho}\wedge k\hat{\kappa}\wedge x\hat{\chi}=true$. The proof of this follows the usual lines and has no features of interest.

Two adverse criticisms might be levelled against this approach to the output of non-terminating programs. Firstly, our semantic equations do not make plain that the numbers in the output

stream cannot influence the course of a program. Yet much the same is true of the environment, which may contain denotations for identifiers occurring nowhere in the program, and just as 1.5.2 can cut the environment down to size so a more complex induction can serve to show the irrelevance of the output to the remainder of the program. Secondly, the equations for which $A=S^{\nu}$ fall foul of recursive declarations in the same manner as those for which $A=V^{\nu}$; however this is immaterial, for the member of A obtained at the end of the program now represents not the final result but a record of all that the store has undergone.

3.6. Manifest types.

3.6.1. Compiled coercions.

Many computer languages adopt manifest types in order to arrange that precisely the right storage is allocated to particular members of V and to facilitate type-checking during a computation. Though these aims are connected they are not identical, for it is easy to conceive of implementations in which all the stored values occupy the same space but type-checking is desirable or in which locations can differ in size but no errors are found until programs are executed. Consequently our discussion of types will be couched in terms of Algol 68 [26], which distinguishes clearly between allocating storage and coercing values (the concomitant of checking their types during compilation). Stored values and locations play contrasting roles in Pal, which extends more readily to the incidence and reference features of Mal than to a language with a hierarchy of types where any object having type M may be kept in a location of type ref M. Accordingly even the syntax used henceforth will be that of strict Algol 68, although recourse will not be made to the more unpleasant aspects of its terminology.

To ensure that every location contains a value which fits it exactly we introduce a lattice M, comprising the meanings of modes, and take the first component of the store domain S to be $L \to [[M \to T] \times V]$. Whenever a new location suitable for holding a given value is required the locations attached only to defunct processes are examined to find out whether they are compatible with the mode of the value. In terms of the domain Q of 3.5.3 *new*:$M \to H \to Q \to S \to L$ must therefore satisfy a more complicated constraint than hitherto; a suitable candidate is

λμηησ.(λα.⋁{((τι↓5)α∨(τι↓6)α)∧τι↓1 | ι:I}∨~*area*ασμ)(*new*μηησ)

=λμηησ.⋀{⋁{((τι↓5)α∨(τι↓6)α)∧τι↓1 | ι:I}∨~*area*ασμ | α:L}→ι,*false*,

where as in 1.3.1 *area*ασ is (σ↓1)α↓1. The member of M→T associated with a location is not regarded as an attribute of the content, which continues to be obtained by taking *hold*ασ to be (σ↓1)α↓2. In general because types can be checked during compilation the mode of an entity need not be incorporated in the corresponding stored value. The only exception to this rule arises in the case of an object having a mode united from several other modes, when the existence of conformity relations entails keeping the original mode of the value along with it; thus the declaration union(bool, char) *c*=true, for instance, makes *c* denote not *true* but $\langle \mu_1, true \rangle$, where μ_1 is some member of M which represents the syntactic entity bool.

Sufficient information is made available during compilation for the types in a program to be checked by a valuation which links each identifier to the type it is declared to possess and thus creates a static environment drawn from the domain Ide→M* before executing the program. For convenience this environment is amalgamated with the dynamic environment in Ide→D°*, thereby providing U with Ide→[M×D°]* as its first component. In practice a member of M will be paired only with those denotations which fall into certain summands of D (so that $\langle \mu_2, true \rangle$, say, will not occur if μ_2 represents char), but notwithstanding this Ide→[M×D°]* is adequate for our present purposes. Strictly speaking we should set up U in a manner resembling that of 3.1.1 in order to ensure that any compiled program obeys the context conditions, which like those of 3.1.4 arrange that no location subject to the stack discipline is passed out beyond its extent, but we shall follow existing implementations by ignoring this

refinement. After responding to the declaration above an environment ρ therefore becomes $\rho[\langle \mu_0, \langle \mu_1, true\rangle\rangle/c]$ where union (bool, char) is represented in M by μ_0.

Just as V need not be identical with D so there is no reason why the lattices M found in $L \mapsto [[M \to T] \times V]$ and in $Ide \mapsto [M \times D^\circ]^*$ should coincide, and indeed those appropriate to Algol 68 do not. More precisely, since selectors $\Sigma_1, \ldots, \Sigma_n$ lying in the flat domain Sel can obtain from an object having mode ref struct (M_1 Σ_1,...,M_n Σ_n) objects of modes ref M_1,...,ref M_n to which assignments may be made independently, the storage occupied by a structure is taken to be a list of locations (in one to one correspondence with the fields of the structure) instead of an indivisible location. Accordingly the lattice M present in $Ide \mapsto [M \times D^\circ]^*$ includes a component which signifies the modes yielded by structures whereas the lattice M in $L \mapsto [[M \to T] \times V]$ does not; our main concern being with type-checking, from now on we shall consider only the former version of M.

In keeping with its insistence that an expressed value belongs to one (and only one) of the lattices L and V Pal permits expressions to be evaluated in three ways, embodied in the valuations $\mathscr{E}$, $\mathscr{L}$ and $\mathscr{R}$. In Algol 68, however, there is a considerable overlap between D and V; in particular it is possible to store any sort of location or even a list of locations. It would be possible to provide syntactic operators which would extract the contents of these locations, but in fact Algol 68 allows the context of an expression to determine when its result is to be forced to become a value of another given type. In principle different sets of coercions could be provided for every production rule in the syntax, but in fact no more than seven such sets are required; as even Pal needs two sets

(comprising *lv* and *rv*) this total is not excessive. To compensate for this paucity of contexts there is a wealth of potential ambiguities arising from the fact that sometimes the type into which the expression must be coerced is not fully known since all that is available is some predicate in M→T which it satisfies.

From this it would appear that we could view a coercion as a member of [M→T]→[M×E]→[M×E] mapping a predicate in M→T and an initial pair ⟨μ,ε⟩:M×E to a final pair. Were this the case there would be no need to build any coercions into a program during its compilation, and we could happily proceed to execute it using an interpreter which would first work out the result of an expression and would then apply a suitable coercion. In order to check the type of an expression we would need a domain of expressed values akin to our present M×E, but nevertheless this approach would be somewhat less complex than the one we shall actually adopt. Regrettably it is invalidated by 'balancing', which we shall now describe.

An expression like a conditional clause or a case clause contains more than one possible exit (in the sense of 1.5.3) and hence more than one possible mode for the expressed value which it would return in the absence of coercions. To arrive at a common final type all but one of these exits are subjected to the strongest possible coercion, thereby being forced to take on the type attained by applying the coercion induced by the context to the remaining exit; should there be more than one way of choosing the exits to be coerced strongly they must all lead to the same final type. The program

ref ref int *u*=loc ref int; ref int *u*=loc int; (true|*u*|*v*):=*v*,

for instance, provides the type ref int for the result of the conditional clause by taking the content of *u* (thereby coercing

it strongly) and by applying a nugatory soft coercion to v; its net effect is therefore to update the location contained in u with the integer contained in v. Were the coercion appropriate to the left hand side of an assignment not applied until the conditional clause had been evaluated u would be updated with the location v because an object of mode ref ref int is not affected by a soft coercion.

Balancing thus obliges the coercions to act on the code which produces expressed values rather than on the values themselves. Setting C=H→Q→S→A, K=E→C and X=U→C as in 3.5.3 we take the coercions to be members of [M→T]→[M×G]→[M×G] where G=[K→C]°. The semantic equations governing an expression must now be supplied with yet two more arguments, one being the predicate π which the type of the ultimate expressed value must satisfy and the other indicating the context o wherein the expression is found. For simplicity we shall give this context the same name as the coercion to which it tallies, although in a more formal treatment we would not do so. Because o maps π and a pair $\langle \mu,\gamma \rangle$:M×G to some other pair in M×G these extra arguments are supplied to the equations after determining the environment and the current process. Accordingly the valuations are exemplified by $\mathscr{U}$:Exp→U→O→P→[M×G] where O=P→[M×G]→[M×G] and P=M→T. The method of floating labels to the head of a block discussed in 1.3.4 requires that we also introduce a valuation $\mathscr{V}$:Exp→U→O→P→[M×G] which assumes that the labels are already known to the environment. Our earlier versions of $\mathscr{E}$ and $\mathscr{I}$ must accordingly be displaced by ones such that for all E:Exp we have $\mathscr{E}[\![E]\!]=\lambda\rho o\pi\kappa\eta\tau\sigma.(\mathscr{U}[\![E]\!]\rho o\pi\downarrow 2\,|\,[K\to C])\eta\tau\sigma$ and also $\mathscr{I}[\![E]\!]=\lambda\rho o\pi\kappa\eta\tau\sigma.(\mathscr{V}[\![E]\!]\rho o\pi\downarrow 2\,|\,[K\to C])\kappa\eta\tau\sigma$.

When the component of U dealing with identifiers is arranged as above the value of $\mathscr{V}[\![I]\!]$ for any I:Ide must therefore

be λρoπ.(λ⟨μ,δ⟩.oπ⟨μ,λκ.(δ:G→*do*(δκ),δ:J→*do*δ,δ:E→*do*(κδ),τ)⟩)(ρ⟦I⟧↓1). The test for members of G in the formula stems from the adoption of recursive declarations akin to those of 1.4.1, not from the presence of procedures without parameters (which are only applied at the instigation of the context). Contrariwise the test for members of J arises because jumps can be executed whether a goto statement occurs or not; more precisely, for every E:Exp 𝒱⟦goto E⟧ is 𝒱⟦E⟧. We can take across to Algol 68 the description of the effect of a jump given in 3.5.3 (even including the definition of *hang*), so J=C° as before; the sole difference is that J is no longer a summand of E or of V despite being one of D.

The semantic equation for 𝒱⟦I⟧ is fairly typical in that o and π appear only in the combination oπ. Indeed were it not for the existence of balancing and of the switches between two forms of strong context in the expressions known as 'confrontations' we could amalgamate these arguments, supplying the valuations with mappings in [M×G]→[M×G] instead. Before clarifying the nature of these mappings we must elucidate the structure of M, and so it is to this task that we now turn.

3.6.2. Declarations of recursive modes.

Because Pal permits any location to contain any stored value it offers no obstacle to the construction of programs in which the constituent locations of certain members of L* may hold further members of L*. In a manifest type language, however, such programs may require a list of location with the same type, and this cannot be achieved in the framework set up so far. Thus although the Pal declaration rec *x*=nil aug false can be written in the cumbersome forms of strict Algol 68 as

ref struct (ref bool *t*) *x*=heap struct (ref bool *t*); *t* of *x*:=false

the fragment rec x=nil aug x calls for a more elaborate treatment. This is provided in Algol 68 by allowing new modes to be created out of those already known; the relevant declarations are recursive, for if they were not nothing could be accomplished with their aid which could not be accomplished without it at the cost of extra writing. In terms of them the fragment above could be translated as the verbose program

mode p=ref struct (p t); p x=heap struct (p t); t of x:=x.

The existence of mode declarations ensures that we cannot automatically identify the syntactic lattice [18] of modes, Mod, with its semantic counterpart, M; instead we must relate them through an environment which binds a meaning to each symbol signifying a mode name in the language. These symbols are the proper elements of a flat lattice of indications, Ind, so during compilation an element of Ind↦M* must be constructed alongside the member of Ide↦M* mentioned in 3.6.1.

Types such as struct (M_1 Σ_1,...,M_n Σ_n) provide only a partial parallel to the members of L* employed by Pal, since without executing an Algol 68 program we can discover all the points in it where a given component of a structure is selected. In this respect members of L* are more closely followed by arrays than by structures; moreover some arrays share with members of L* the further property of having flexible bounds. These bounds can be set to their initial values by supplying expressions as part of the declarer M:Mod in an Algol 68 identity declaration M I=E. For the sake of consistency bounds must also be permitted in mode T=M, where T:Ind is declared to have the same meaning as M; under these circumstances, however, the bounds are not evaluated when T is declared but only when T is later used in an identity declaration. We shall not dwell on the nature of array bounds in Algol 68 or

on the valuations required by their semantics but will presume, rather inaccurately, that a list of such bounds is an expressed value in E. Were this so the continuation appropriate to a declarer would belong to K and the valuation determining its outcome would be in Mod→U→K→C; in consequence the effect of a type declaration mode T=M would be to include in the denotation of T a member of K→C which would not be immediately applied. Accordingly a dynamic environment lying in Ind↛G* must be introduced; as the declarations encountered during the compilation of a program tally with those encountered during its execution this environment can be combined with Ind↛M* to give Ind↛[M×G]*, the second component of U. To handle declarations of operators U must even be supplied with a third component, but this raises no new questions and will be therefore be ignored. The domain U will thus be [Ide↛[M×D°]*]×[Ind↛[M×G]*], while any ρ:U will yield lists ρ⟦I⟧ and ρ⟦T⟧ for all I:Ide and T:Ind.

Note in passing that a language in which lists of array bounds actually did form part of E would in one sense be more general and cohesive than Algol 68. Yet the plethora of specialized semantic domains which is found in Algol 68 does provide a means of making programs less prone to errors by restricting the positions in them where particular 'parts of speech' can occur, and it is not clear that reducing syntactic restrictions inevitably (or even frequently) gives rise to a wider class of useful programs. The evolution of programming languages has often entailed trading in a rather special construction for one with wider applicability but less security: this has happened, for instance, in the replacement of I:=E by E_0:=E_1 and indeed in the move from languages with manifest types to those without. Whereas removing petty rules usually leads to more elegant semantic equations, allowing too much liberty may not do so and

will hinder proofs about particular programs. The extent to which language designs resolve this conflict is an important criterion for their success.

To describe the influence of modes on coercions we do not need to discuss the valuation in Mod→U→K→C, as it is relevant only during the execution of the program. We shall therefore examine only $\mathcal{N}$:Mod→U→M, which ignores array bounds when describing types. The production rules reveal that a syntactic mode must be an indication or a basic type like bool or char unless it combines other modes in a representation of a complex entity such as a procedure without parameters or a structure. Thus we take the lattice Mod to be

Ind+{bool}°+{char}°+...+{proc}°×Mod+{struct}°×[Mod×Sel]*+...,

where for simplicity we have detailed only a few of the components and have admitted the non-existent type signifying structures with no selectors.

The lattice M is roughly homologous to Mod in that the semantic quantities *bool* and *char*, for instance, must be double atoms. Being intended to provide the 'absolute' meanings of modes without reference to environments or evaluating mechanisms M does not contain a version of Ind, the domain of indications. Notwithstanding this, such distinctions as that between procedures and structures must carry over from Mod to M so that corresponding to every other summand of Mod must be one of M. To replace {proc}°×Mod by {*proc*}°×M would be incorrect, however, for although proc ⊥ cannot be written in a program intuitively proc p should tally with ⟨*proc*,⊥⟩ in an environment where p indicates the type ⊥. Adopting the coalesced product of 1.2.2 therefore entails substituting {*proc*}°×M° for {proc}°×Mod; similar alterations must be made to the other summands, so

$\{bool\}^\circ+\{char\}^\circ+\ldots+\{proc\}^\circ\times M^\circ+\{struct\}^\circ\times[M^\circ\times Sel]^*+\ldots$ gives the structure of M. Such constants as *proc* and *struct* are of course irrelevant to the formulation of M, but their presence has some pedagogical value.

The valuation $\mathcal{M}$:Mod→U→M is defined for every T:Ind by taking $\mathcal{M}[\![T]\!]$ to be $\lambda\rho.\rho[\![T]\!]\downarrow 1\downarrow 1$ when $\lambda\rho.\rho[\![T]\!]$ belongs to U→[M×G]*. The basic modes must be given meanings which take over to M the roles fulfilled by the types in Mod; hence $\mathcal{M}[\![\text{bool}]\!]$ and $\mathcal{M}[\![\text{char}]\!]$, for instance, must be $\lambda\rho.bool$ and $\lambda\rho.char$ respectively. Since the principle underlying the interpretation of the basic types applies equally to the higher types $\mathcal{M}[\![\text{proc M}]\!]$ is $\lambda\rho.\langle proc,\mathcal{M}[\![M]\!]\rho\rangle$ and $\mathcal{M}[\![\text{struct }(M_1\ \Sigma_1,\ldots,M_n\ \Sigma_n)]\!]$ is $\lambda\rho.\langle struct,\langle\langle\mathcal{M}[\![M_1]\!]\rho,\Sigma_1\rangle,\ldots,\langle\mathcal{M}[\![M_n]\!]\rho,\Sigma_n\rangle\rangle\rangle$. Only if M is taken to be $\{bool\}^\circ+\{char\}^\circ+\ldots+\{proc\}^\circ\times M^\circ+\{struct\}^\circ\times[M^\circ\times Sel]^*+\ldots$, not $\{bool\}^\circ+\{char\}^\circ+\ldots+\{proc\}^\circ\times M+\{struct\}^\circ\times[M\times Sel]^*+\ldots$, does this recursive construction of $\mathcal{M}$ give such modes as proc p a value other than $\perp$.

As every Algol 68 declaration is recursive, in accordance with the remarks of 1.3.4 the meaning of any Δ:Dec can be obtained only by making its constituents of the form mode T=M interact. This can be done by collecting up the identifiers and indications declared, as well as the modes to which they correspond, using $\mathcal{A}$:Dec→[[Mod×Ide]+[Mod×Ind]]*. The construction of this valuation requires taking $\mathcal{A}[\![\text{M I=E}]\!]$ to be $\langle\langle M,I\rangle\rangle$, $\mathcal{A}[\![\text{mode T=M}]\!]$ to be $\langle\langle M,T\rangle\rangle$, $\mathcal{A}[\![\text{E; }\Delta]\!]$ to be $\mathcal{A}[\![\Delta]\!]$ and $\mathcal{A}[\![\Delta_1,\ldots,\Delta_n]\!]$ to be $\mathcal{A}[\![\Delta_1]\!]\,\S\ldots\S\,\mathcal{A}[\![\Delta_n]\!]$.

Although the four kinds of declaration mentioned in the definition of $\mathcal{A}$ are only a sample of what Algol 68 has to offer they provide clear guidance as to how the others are to be understood. On this basis it is possible to introduce $\mathcal{X}$:Dec→U→U,

which yields the part of the meaning of a declaration that can be extracted during compilation. Given any Δ:Dec $\mathscr{D}[\![\Delta]\!]$ is $\lambda\rho.fix(\lambda\rho'.fix(\lambda\phi\nu.\nu>\#\mathscr{A}[\![\Delta]\!]\rightarrow\rho,\phi(\nu+1)[\mathscr{C}[\![\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow1]\!]\rho',\perp)/\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow2])1)$, for declarations of operators would complicate the definition of $\mathscr{A}$, not that of $\mathscr{D}$. The presence of Ide$\mapsto$D°* and Ind$\mapsto$G* in the domain U is irrelevant to the roles of these valuations, so the entity $\perp$:[K$\rightarrow$C]° is put into the environment set up by $\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow2$ purely for the sake of convenience.

The lattice M can readily provide all the members of O required by the valuations of 3.6.1. In a soft context, for example, an expression must be left unchanged by the coercion unless it is of mode proc M_0 for some M_0:Mod, when the code compiled for the expression may include an invocation of the corresponding procedure. This process may be repeated until an object having a type satisfying the contextual predicate π:P is at last obtained; thus should M_0 itself be proc M_1 for some M_1:Mod a second invocation may be inserted by the compiler. An element of O appropriate to soft contexts is therefore given by $soften=\lambda\pi\langle\mu,\gamma\rangle.\pi\mu\rightarrow\langle\mu,\gamma\rangle,\mu:\{proc\}°\times M°\rightarrow soften\pi\langle\mu\downarrow2,\lambda\kappa.\gamma(\lambda\varepsilon.(\varepsilon|G)\kappa)\rangle,\perp$. The other coercions can be constructed in a similar fashion although tiresome complications must be introduced to handle such phenomena as uniting, where one mode can be converted into many union modes each containing it.

Unfortunately M is less satisfactory as a source of elements of P. Many of the contexts in which an expression can stand require only that after performing the appropriate coercion the type of the expression satisfy a simple predicate like $\lambda\mu.\mu:\{struct\}°\times[M°\times Sel]*$. Strong contexts, however, demand that the final type of the expression be tested by $\lambda\mu.\mu=\mu_0$ for some μ_0:M. Since M is not a flat lattice no continuous test for

equality can be imposed on it; however, we know that in practice if $\mathfrak{m}$ is the set of meanings of modes that arise in programs then the members of m are mutually incomparable. We can also introduce a function, *equal*, such that

$$
\begin{aligned}
equal = \lambda\nu\mu_0\mu_1.\ & \nu<1 \to true,\\
& (\mu_0 = bool)\wedge(\mu_1 = bool) \to true,\\
& (\mu_0 = char)\wedge(\mu_1 = char) \to true,\\
& \ldots,\\
& \bigwedge\{\mu_n : \{proc\}^\circ \times M^\circ \mid 0 \le n \le 1\} \to equal(\nu-1)(\mu_0 \downarrow 2)(\mu_1 \downarrow 2),\\
& \bigwedge\{\mu_n : \{struct\}^\circ \times [M^\circ \times Sel]^* \mid 0 \le n \le 1\} \to\\
& \qquad (\sim(\#(\mu_0 \downarrow 2) = \#(\mu_1 \downarrow 2)) \to false,\\
& \qquad \sim\bigwedge\{\mu_0 \downarrow 2 \downarrow \nu \downarrow 2 = \mu_1 \downarrow 2 \downarrow \nu \downarrow 2 \mid 1 \le \nu \le \#(\mu_0 \downarrow 2)\} \to false,\\
& \qquad \bigwedge\{equal(\nu-1)(\mu_0 \downarrow 2 \downarrow \nu' \downarrow 1)(\mu_1 \downarrow 2 \downarrow \nu' \downarrow 1) \mid 1 \le \nu' \le \#(\mu_0 \downarrow 2)\}),\\
& \ldots,\\
& false.
\end{aligned}
$$

For every ν $equal\nu$ is continuous, but there are many sequences $\{\mu_n \mid n \ge 0\}$ for which $\mu_{n+1} \sqsupseteq \mu_n$ when $n \ge 0$, $\bigsqcup\mu_n$ can arise from a valid Algol 68 declaration and $\bigwedge\{equal\nu(\bigsqcup\mu_n)(\bigsqcup\mu_m) \mid \nu : N\} = true$ though $\bigwedge\{equal\nu(\bigsqcup\mu_n)\mu_m \mid \nu : N\} = \bot$ for all $m \ge 0$; such sequences can be constructed using the method of 2.1.6. Thus it is possible to incorporate *equal* in the parameters of the semantic equations only by introducing discontinuous functions and using the subset m of M or by providing equations for every value of $equal\nu$. This is sad but perhaps not unacceptable; intuitively it is to be expected that for any valid program there can be derived an integer $\nu : N$ such that any two modes required by the program can be distingished from one another by $equal\nu$. This result will be established in 3.6.5 with the aid of a treatment of modes more in keeping with the style of the Algol 68 report than the one developed above.

3.6.3. Interpreted roles for modes.

The unsatisfactory nature of *equal* arises from the fact that M is not flat; yet for the reasons outlined in 2.1.6 to adopt a flat lattice in place of M is to abandon all hope of using *fix* to determine the outcome of a recursive type declaration (though we could use m to factor out the part of M that we need). Here we shall define a valuation on a flat lattice which will be equivalent to the valuation $\mathcal{N}$ of 3.6.2 and which will also model the obvious implementation technique for recursive types.

Underlying the semantic equations mentioned in 3.6.2 is the convention that the value given to any T:Ind by an environment ρ:U can be obtained simply by inspecting the entry for T in ρ; thus $\mathcal{N}[\![T]\!]$ is $\lambda\rho.\rho[\![T]\!]\downarrow 1\downarrow 1$. In an implementation, however, the compiled 'call by value' implicit in this gives way to an interpreted 'call by name' which replaces T by the right hand side of its declaration. This operation could be expressed mathematically by regarding the component of ρ concerned with indications not as a member of Ind↦[M×G]* but as a member of Ind↦[Mod×G]*, then a valuation $\mathcal{M}$:Mod→U→M, for which $\mathcal{M}[\![T]\!]$ would be $\lambda\rho.\mathcal{M}[\![\rho[\![T]\!]\downarrow 1\downarrow 1]\!]\rho$, might be needed instead of $\mathcal{N}$.

To define U in this way would be to succumb to a temptation like that of giving denotations to the free variables of a procedure when applying it rather than when declaring it. This snare could be avoided by adopting Ind↦[Mod×U×G]* rather than Ind↦[Mod×G]* and by taking $\mathcal{M}[\![T]\!]$ to be $\lambda\rho.\mathcal{M}[\![\rho[\![T]\!]\downarrow 1\downarrow 1]\!](\rho[\![T]\!]\downarrow 1\downarrow 2)$; however the coalesced product of 1.2.2 is such that were Ind↦[Mod×U×G]* to be a component of U then the version of U derived from the minimal fixed point of a functor would contain at most two elements. The use of Ind↦[Mod×U°×G]* instead would provide a sufficiently large domain U only at the cost of depending crucially on lattices other than flat ones, and thus

cannot be tolerated.

Fortunately 3.1.1 offers a hope of salvation: as part of the value of an indication will be kept not an environment but a cluster of pointers. This cluster will be a certain ψ:Ind↔N such that for any T:Ind $\psi[\![T]\!]$ will show where to find the appropriate occurrence of T in the entire environment. Thus the part of dealing with Ind will be Ind↔[Mod×[Ind N]×G]* and $\mathscr{M}[\![T]\!]$ will be roughly

$\lambda\rho.\mathscr{M}[\![\rho[\![T]\!]\downarrow 1\downarrow 1]\!](\lambda I.\langle\rangle, \lambda T'.\rho[\![T']\!]\dagger(\#\rho[\![T']\!]-(\rho[\![T]\!]\downarrow 1\downarrow 2)[\![T']\!]))$.

Because a declaration of the form M I=E necessitates representing M in the environment alongside the denotation of I the component of U concerned with identifiers is taken to be Ide↔[Mod×[Ind↔N]×D°]*. As U is now [Ide↔[Mod×[Ind↔N]×D°]*]×[Ind↔[Mod×[Ind↔N]×G]*] the valuation $\mathscr{E}$ must be superseded by a certain $\mathscr{D}$:Dec→U→U, just as $\mathscr{N}$ is superseded by $\mathscr{M}$. For any Δ:Dec $\mathscr{D}[\![\Delta]\!]$ can be written in terms of the valuation $\mathscr{A}$ of 3.6.2 as

$\lambda\rho.(\lambda\psi.fix(\lambda\phi\nu.\nu>\#\mathscr{A}[\![\Delta]\!]\rightarrow\rho,\phi(\nu+1)[\langle\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow 1,\psi,\perp\rangle/\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow 2])1)$

$(\lambda T.(\bigvee\{1\leq\nu\leq\#\mathscr{A}[\![\Delta]\!]\rightarrow(T=\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow 2), false \mid \nu:N\}\rightarrow\#\rho[\![T]\!]+1,\#\rho[\![T]\!]))$;

if declarations were not recursive $\mathscr{D}[\![\Delta]\!]$ would be

$\lambda\rho.fix(\lambda\phi\nu.\nu>\#\mathscr{A}[\![\Delta]\!]\rightarrow\rho,\phi(\nu+1)[\langle\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow 1,\lambda T.\#\rho[\![T]\!],\perp\rangle/\mathscr{A}[\![\Delta]\!]\downarrow\nu\downarrow 2])1.$

The member of Ind↔N attached to an identifier or an indication by a declaration can be used to extract the pertinent entries in the environment even when further layers have been piled on top. To emphasize the irrelevance of the third component of Mod×U×G to the determination of the value of a type the environment formed by the process of extraction will keep $\perp$ in each such component. The function *dip*:[Ind↔N]→U→U will therefore be given by

dip=λψρ.(λφ.⟨λI.⟨⟩,λT.φ(ρ⟦T⟧↑(#ρ⟦T⟧-ψ⟦T⟧))⟩)

(fix(λφω*.#ω*=0→⟨⟩,⟨⟨ω*↓1↓1,ω*↓1↓2,⊥⟩⟩§φ(ω*↑1))).

The complexity of dip could of course be reduced by allowing it to yield answers in Ind→[Mod×[Ind↦N]]*instead of in U; only a desire for conceptual economy runs counter to this.

Even when the portion of U concerned with Ind is Ind↦[Mod×[Ind↦N]×G]* rather than Ind↦[M×G]* it is possible to convert a pair ⟨M,ρ⟩ (representing the meaning of an indication relative to an environment) into an 'absolute' value in M. This operation can be performed by $\mathscr{M}$:Mod→U→M; this must provide any T:Ind with the same meaning as the element of Mod linked to it by the environment, so $\mathscr{M}$⟦T⟧ must be λρ.$\mathscr{M}$⟦ρ⟦T⟧↓1↓1⟧(dip(ρ⟦T⟧↓1↓2)ρ). Needless to say $\mathscr{M}$⟦bool⟧ is λρ.$bool$ and $\mathscr{M}$⟦char⟧ is λρ.$char$ while the higher types are given meanings by extending $\mathscr{M}$ in an obvious way; thus $\mathscr{M}$⟦proc M⟧ is λρ.⟨$proc$,$\mathscr{M}$⟦M⟧⟩ and $\mathscr{M}$⟦struct (M_1 Σ_1,...,M_n Σ_n)⟧ is λρ.⟨$struct$,⟨⟨$\mathscr{M}$⟦M_1⟧ρ,Σ_1⟩,...,⟨$\mathscr{M}$⟦M_n⟧ρ,Σ_n⟩⟩⟩.

For any environments $\acute{\rho}$ and $\grave{\rho}$ drawn from the domains [Ide↦[M×D°]*]×[Ind↦[M×G]*] and [Ide↦[Mod×[Ind↦N]×D°]*]×[Ind↦[Mod×[Ind↦N]×G]*] respectively the mappings λM.$\mathscr{M}$⟦M⟧$\acute{\rho}$ and λM.$\mathscr{M}$⟦M⟧$\grave{\rho}$ are homomorphisms of Mod viewed as a word algebra into M regarded in the same light. If $\acute{\rho}$ and $\grave{\rho}$ are produced by the same program these homomorphisms might reasonably be expected to coincide. That this does indeed happen will be proved in 3.6.6; here the foundations for the proof will be laid by providing a means of building $\acute{\rho}$ from $\grave{\rho}$, namely

$turn$=λνρ.(λφ.⟨λI.φ(ρ⟦I⟧),λT.φ(ρ⟦T⟧)⟩)

(fix(λφω*.(λμ.#ω*=0→⟨⟩,⟨⟨μ,ω*↓1↓3⟩⟩§φ(ω*↑1))

$\mathscr{M}$⟦ω*↓1↓1⟧($turn$(ν-1)(dip(ω*↓1↓2)ρ)))))).

Until 3.6.6 any mention of an environment will be an allusion to a member of [Ide↦[Mod×[Ind↦N]×D°]*]×[Ind↦[Mod×[Ind↦N]×G]*], not to a member of [Ide↦[M×D°]*]×[Ind↦[M×G]*].

The environments set up by $\mathscr{X}$ during the execution of a declaration associate only finitely many members of Ide and Ind with lists in [Mod×[Ind→N]×G]* which are not empty. Furthermore the pointers provided by these lists must themselves refer to entries drawn from the narrow confines of the environment. These constraints can be summarized by giving ∞ its conventional interpretations and setting

$$slim=\lambda\rho.(\lambda\phi.((\sum\{\#\rho[\![I]\!]\,|\,I:Ide\}+\sum\{\#\rho[\![T]\!]\,|\,T:Ind\})<\infty)$$
$$\wedge\bigwedge\{\bigwedge\{1\le\nu\le\#\rho[\![I]\!]\rightarrow\phi(\rho[\![I]\!]\downarrow\nu),true\,|\,I:Ide\}$$
$$\wedge\bigwedge\{1\le\nu\le\#\rho[\![T]\!]\rightarrow\phi(\rho[\![T]\!]\downarrow\nu),true\,|\,T:Ind\}\,|\,\nu:N\})$$
$$(\lambda\langle M,\psi,\gamma\rangle.\bigwedge\{((\#\rho[\![T_0]\!]=0\rightarrow 0,1)\le\psi[\![T_0]\!]\le\#\rho[\![T_0]\!])$$
$$\wedge(1\le\nu\le\psi[\![T_0]\!]\rightarrow((dip\psi\rho[\![T_0]\!]\downarrow\nu\downarrow 2)[\![T_1]\!]\le\psi[\![T_1]\!])),true)$$
$$\wedge(M:Ind\rightarrow(1\le\#\rho[\![M]\!]),true)\,|\,T_0:Ind\wedge T_1:Ind\}$$

For all Δ:Dec and all ρ:U if $slim\rho=true$ then patently $slim(\mathscr{X}[\![\Delta]\!]\rho)=true$. More significantly, if $slim\rho=true$ and $\#\rho[\![T]\!]>0$ for some T:Ind then $slim(dip(\rho[\![T]\!]\downarrow 1\downarrow 2)\rho)=true$.

Demanding that coercions take as parameters predicates in M→T leads to the difficulties encountered in 3.6.2. The predicates to be discussed below are therefore defined not on M but on Mod×U, the elements of which can be mapped into M by applying $\mathscr{M}$. Although the domains D° and G embedded in U ensure that Mod×U is not flat, those members of Mod×U which must be distinguished from one another differ in respects that do not depend on D° or G; consequently the predicate chosen to express the equality of two modes can be made continuous. A typical coercion o is in [[Mod×U]→T]→[Mod×U×G]→[Mod×U×G] while every predicate required by a context belongs to [Mod×U]→T. In particular, for any I:Ide $\mathscr{V}[\![I]\!]$ is

$$\lambda\rho o\pi.(\lambda\gamma.o\pi\langle\rho[\![I]\!]\downarrow 1\downarrow 1,dip(\rho[\![I]\!]\downarrow 1\downarrow 2)\rho,\gamma\rangle)$$
$$((\lambda\delta.\lambda\kappa.(\delta:G\rightarrow do(\delta\kappa),\delta:J\rightarrow do\delta,\delta:E\rightarrow do(\kappa\delta),\tau))(\rho[\![I]\!]\downarrow 1\downarrow 3)).$$

Though Mod×U gives rise to a simpler treatment of equality than does M, such predicates as that revealing whether a given pair $\langle M,\rho\rangle$ signifies a procedure are less easily defined than their counterparts on M. Only after starting to evaluate any $\langle T,\rho\rangle$:Ind×U is it possible to establish that T has been declared to be a procedure; moreover the first step in the evaluation of $\langle T,\rho\rangle$ may simply lead to another member of Ind×U, so underlying the predicate must be a recursive algorithm. This is prevented from returning the answer $\perp$ on encountering a pair $\langle p,\rho\rangle$ with $\rho[\![p]\!]\downarrow 1$ equal to $\langle \rho,\lambda T.\#\rho[\![T]\!],\perp\rangle$ (and with $\mathscr{M}[\![p]\!]\rho$ equal to $\perp$) by the existence of a list in [Mod×U]* which is compared with every member of Mod×U produced during the evaluation. The comparison is effected by a continuous function which checks the top element of the list to see whether it tallies with any of the succeeding elements of the list in those respects which can influence the evaluation of a mode. In fact it is convenient to be able to test two lists simultaneously, so this function, *kept*, is deemed to be a member of [Mod×U]*→[Mod×U]*→T. Throughout the remainder of this section ξ will signify a typical member of [Mod×U]*; under this convention *kept* may therefore be defined by

$$kept=\lambda\xi_0\xi_1.(\lambda\phi_0.(\lambda\phi_1.\bigvee\{2\le\nu\le\#\xi_0\rightarrow(\phi_0 1=\phi_0\nu)\wedge(\phi_1 1=\phi_1\nu),false\mid\nu:N\})$$
$$(\lambda\nu.\langle\xi_1\downarrow\nu\downarrow 1,dip(\lambda T.\#(\xi_1\downarrow\nu\downarrow 2)[\![T]\!])(\xi_1\downarrow\nu\downarrow 2)\rangle))$$
$$(\lambda\nu.\langle\xi_0\downarrow\nu\downarrow 1,dip(\lambda T.\#(\xi_0\downarrow\nu\downarrow 2)[\![T]\!])(\xi_0\downarrow\nu\downarrow 2)\rangle).$$

It is necessary to carry out the checks involved in *kept* when members of Ind×U are reduced to members of Mod×U which are not themselves in Ind×U. This can be achieved using *wend*:N→[Mod×U]*→[Mod×U]*; for every $\langle M,\rho\rangle$:Mod×U this gives rise to $\bigsqcup\{wend\nu(\langle M,\rho\rangle)\mid\nu:N\}$, which lists the members of Mod×U created during this reduction. As the integral parameter ν supplied to *wend* is intended simply to measure the depth of recursion required,

$wend$=λνξ.ν<1→⊥,

~(ξ↓1↓1:Ind)∨$kept$ξξ→ξ,

(λ⟨M,ψ,γ⟩.$wend$(ν-1)(⟨M,dipψ(ξ↓1↓2)⟩)§ξ))((ξ↓1↓2)‖ξ↓1↓1‖↓1);

$wind$=λξ.⨆{$wend$νξ|ν:N}.

The 'shielding' conditions of the Algol 68 report ensure that no pair ⟨M,ρ⟩ which arises in a program can have $wind$⟨⟨M,ρ⟩⟩↓1↓1:Ind; were this not the case calamities could arise with the valuation $\mathscr{M}$, as 3.6.4 will indicate.

Any mode M_0 having the form proc M_1 for some M_1 actually belongs to {proc}°×Mod, so M_0↓2 coincides with M_1; likewise when M_0 is struct (M_1 Σ_1,...,M_n Σ_n) its second component, M_0↓2, can be taken to be ⟨⟨M_1,Σ_1⟩,...,⟨M_n,Σ_n⟩⟩. Accordingly $soften$, for instance, now satisfies

$soften$=λπ⟨M,ρ,γ⟩.(λ⟨M',ρ'⟩.~(M':Ind)∧π⟨M',ρ'⟩→⟨M',ρ',γ⟩,

~(M':{proc}°×Mod)→⊥,

⟨M'↓2,ρ',λκ.γ(λε.(ε|G)κ)⟩)

($wind$⟨⟨M,ρ⟩⟩↓1).

In 3.6.5 it will be necessary to have some knowledge of the textual composition of syntactic modes. This can be obtained by listing all the shorter modes out of which they are built; thus it is helpful to have available lug, which is given by

lug=λM.M:Ind→⟨M⟩,

(M=bool)→⟨M⟩,

(M=char)→⟨M⟩,

....

M:{proc}°×Mod→⟨M⟩§lug(M↓2),

M:{struct}°×[Mod×Sel]*→⟨M⟩§lug(M↓2↓1↓1)§...§lug(M↓2↓#(M↓2)↓1)

....

The test for equality between members of Mod×U, $bend$, is built up in much the same way as $wend$ but requires two lists in

[Mod×U]* (namely ξ_0 and ξ_1) and an integer ν_1 to mark off parts of the lists when they are examined. Unless the pairs $\xi_0\downarrow\nu_1$ and $\xi_1\downarrow\nu_1$ are found together in $\xi_0\dagger\nu_1$ and $\xi_1\dagger\nu_1$ their structure is investigated more closely. Should either of them be a member of Ind×U which is reduced by *wind* to another member of Ind×U, a parameter of *bend* becomes 2; as 3.6.5 will establish, in the same situation *equal* yields the answer $\perp$. Should $wind\langle\xi_0\downarrow\nu_1\rangle\downarrow1\downarrow1$ and $wind\langle\xi_1\downarrow\nu_1\rangle\downarrow1\downarrow1$ be blatantly dissimilar, the parameter becomes 1, but the final answer may nevertheless be 2. Thus the function *bend*:N→N→[Mod×U]*→[Mod×U]*→N→[[Mod×U]*×[Mod×U]*×N] is given by

$$
\begin{aligned}
bend=\lambda\nu_0\nu_1\xi_0\xi_1\varepsilon.(\lambda\phi.(\lambda\langle M_0,M_1\rangle.&\nu_0<1\rightarrow\perp,\\
&\nu_1<1\rightarrow\langle\xi_0,\xi_1,\varepsilon\rangle,\\
&kept(\xi_0\dagger(\nu_1-1))(\xi_1\dagger(\nu_1-1))\rightarrow\phi()()\varepsilon,\\
&M_0:\mathrm{Ind}\vee M_1:\mathrm{Ind}\rightarrow\phi()()2,\\
&(M_0=\mathrm{bool})\wedge(M_1=\mathrm{bool})\rightarrow\phi()()\varepsilon,\\
&(M_0=\mathrm{char})\wedge(M_1=\mathrm{char})\rightarrow\phi()()\varepsilon,\\
&\ldots,\\
&\bigwedge\{M_n:\{\mathrm{proc}\}^\circ\times\mathrm{Mod}\mid0\le n\le1\}\rightarrow\phi\langle M_0\downarrow2\rangle\langle M_1\downarrow2\rangle\varepsilon,\\
&\bigwedge\{M_n:\{\mathrm{struct}\}^\circ\times[\mathrm{Mod}\times\mathrm{Sel}]^*\mid0\le n\le1\}\rightarrow\\
&\qquad(\sim(\#(M_0\downarrow2)=\#(M_1\downarrow2))\rightarrow\phi()()(1\vee\varepsilon),\\
&\qquad\sim\bigwedge\{M_0\downarrow2\downarrow\nu\downarrow2=M_1\downarrow2\downarrow\nu\downarrow2\mid1\le\nu\le\#(M_0\downarrow2)\}\rightarrow\\
&\qquad\qquad\phi()()(1\vee\varepsilon),\\
&\qquad((\lambda\psi.\phi(\psi0)(\psi1))\\
&\qquad\quad(\lambda n.\langle M_n\downarrow2\downarrow1\downarrow1,\ldots,M_n\downarrow2\downarrow\#(M_n\downarrow2)\downarrow1\rangle))\\
&\ldots,\\
&\phi()()(1\vee\varepsilon))\\
&\langle wind\langle\xi_0\downarrow\nu_1\rangle\downarrow1\downarrow1,wind\langle\xi_1\downarrow\nu_1\rangle\downarrow1\downarrow1\rangle\\
&(\lambda\langle M^*_0,M^*_1\rangle.(\lambda\psi.bend(\nu_0-1)(\nu_1-1+\#M^*_0)(\psi0\S\xi_0)(\psi1\S\xi_1)\varepsilon)\\
&(\lambda n.fix(\lambda\psi\nu.\nu>\#M^*_n\rightarrow(),\\
&\qquad\langle\langle M^*_n\downarrow\nu,wind\langle\xi_n\downarrow\nu\rangle\downarrow1\downarrow2\rangle\rangle\S\psi(\nu+1)\rangle1)
\end{aligned}
$$

$$bind=\lambda\nu\xi_0\xi_1.\bigsqcup\{bend\nu_0\nu\xi_0\xi_1\mid\nu_0:\mathrm{N}\}.$$

3.6.4. Proposition.

Let ξ_0 and ξ_1 be members of [Mod×U]* having $\xi_1 = wind\xi_0$; define $\langle M_0,\rho_0\rangle = \xi_0\downarrow 1$ and $\langle M_1,\rho_1\rangle = \xi_1\downarrow 1$, and suppose that $slim\rho_0 = true$ and that $\#\rho_0[\![M_0]\!] > 0$ if M_0:Ind. Inevitably M_1 is proper and $\mathscr{M}[\![M_0]\!]\rho_0 = \mathscr{M}[\![M_1]\!]\rho_1$. Moreover if M_1 is a member of Ind either $\mathscr{M}[\![M_0]\!]\rho_0 = \perp$ or $kept(\langle\langle M_1,\rho_1\rangle\rangle \S\xi_0\dagger 1)(\langle\langle M_1,\rho_1\rangle\rangle \S\xi_0\dagger 1) = true$, while if M_1 is not a member of Ind $\mathscr{M}[\![M_0]\!]\rho_0$ is proper.

⊲Throughout the proof attention will be fixed on only one suitable list ξ_0 (and the corresponding list ξ_1). Since $slim\rho_0 = true$ any ν_0:N, ν_1:N, T_0:Ind and T_1:Ind having $1\le\nu_0\le\#\rho_0[\![T_0]\!]$ and $1\le\nu_1\le(\rho_0[\![T_0]\!]\downarrow\nu_0\downarrow 2)[\![T_1]\!]$ satisfy

$dip(dip\psi\rho_0[\![T_1]\!]\downarrow\nu_1\downarrow 2)\rho_0 = dip(\rho_0[\![T_1]\!]\downarrow(\nu_1+\#\rho_0[\![T_1]\!]-\psi[\![T_1]\!])\downarrow 2)\rho_0$

when $\psi = \rho_0[\![T_0]\!]\downarrow\nu_0\downarrow 2$. Hence if $i\xi$ is defined to be the difference between $\sum\{\#\rho_0[\![T]\!] \mid T\text{:Ind}\}$ and the number of pairs of the form $\langle T_0,\nu_0\rangle$ having

$(\lambda\xi'.kept\xi'\xi')(\langle\langle \rho_0[\![T_0]\!]\downarrow\nu_0\downarrow 1, dip(\rho_0[\![T_0]\!]\downarrow\nu_0\downarrow 2)\rho_0\rangle\rangle \S\xi) = true$ it can readily be proved by induction on $i\xi$ that for all ξ:[Mod×U]*, T_1:Ind and ν_1:N $wind(\langle\langle \rho_0[\![T_1]\!]\downarrow\nu_1\downarrow 1, dip(\rho_0[\![T_1]\!]\downarrow\nu_1\downarrow 2)\rho_0\rangle\rangle \S\xi)$ coincides with $wend(1+i\xi)(\langle\langle \rho_0[\![T_1]\!]\downarrow\nu_1\downarrow 1, dip(\rho_0[\![T_1]\!]\downarrow\nu_1\downarrow 2)\rho_0\rangle\rangle \S\xi)$. In particular ξ_1 is proper and equal to $wend(1+\sum\{\#\rho_0[\![T]\!] \mid T\text{:Ind}\})\xi_0$; moreover $\#\xi_1 \le \sum\{\#\rho_0[\![T]\!] \mid T\text{:Ind}\} + \#\xi_0$. By similar reasoning ξ_0 is $\xi_1\dagger(\#\xi_1 - \#\xi_0)$.

To proceed further it is necessary to truncate $\mathscr{M}$ so that it follows the algorithm for *wend* step by step. Thus for all M:Mod $\mathscr{M}_0[\![M]\!]$ is taken to be $\lambda\rho.\perp$, while when $m\ge 0$ $\mathscr{M}_{m+1}[\![T]\!]$ is $\lambda\rho.\mathscr{M}_m[\![\rho[\![T]\!]\downarrow 1\downarrow 1]\!](dip(\rho[\![T]\!]\downarrow 1\downarrow 2)\rho)$ for every T:Ind, $\mathscr{M}_{m+1}[\![bool]\!]$ is $\lambda\rho.bool$, $\mathscr{M}_{m+1}[\![char]\!]$ is $\lambda\rho.char$, $\mathscr{M}_{m+1}[\![proc\ M]\!]$ is $\lambda\rho.\langle proc, \mathscr{M}_m[\![M]\!]\rangle$ for every M:Mod and $\mathscr{M}_{m+1}[\![struct\ (M_1\ \Sigma_1,\ldots,M_n\ \Sigma_n)]\!]$ is $\lambda\rho.\langle struct, \langle\langle \mathscr{M}_m[\![M_1]\!],\Sigma_1\rangle,\ldots,\langle \mathscr{M}_m[\![M_m]\!],\Sigma_m\rangle\rangle\rangle$. Henceforth it will be assumed that $\mathscr{M}$ is the minimal fixed point of the mutually re-

cursive semantic equations which it satisfies; this assumption can be embodied in the assertion that $\mathscr{M}=\bigsqcup\mathscr{M}_n$.

Plainly $\mathscr{M}_n[\![\xi_1\downarrow 1\downarrow 1]\!](\xi_1\downarrow 1\downarrow 2)=\mathscr{M}_n[\![M_1]\!]\rho_1$ if $n\geq 0$. Assume that for some m having $0\leq m\leq \#\xi_1-\#\xi_0-1$ $\mathscr{M}_{n+m}[\![\xi_1\downarrow(m+1)\downarrow 1]\!](\xi_1\downarrow(m+1)\downarrow 2)=\mathscr{M}_n[\![M_1]\!]\rho_1$ for all $n\geq 0$. As $\xi_1\downarrow(m+1)\downarrow 1=(\xi_1\downarrow(m+2)\downarrow 2)[\![\xi_1\downarrow(m+2)\downarrow 1]\!]\downarrow 1\downarrow 1$ and $\xi_1\downarrow(m+1)\downarrow 2=dip((\xi_1\downarrow(m+2)\downarrow 2)[\![\xi_1\downarrow(m+2)\downarrow 1]\!]\downarrow 1\downarrow 2)(\xi_1\downarrow(m+2)\downarrow 2)$, for every $n\geq 0$ $\mathscr{M}_{n+m+1}[\![\xi_1\downarrow(m+2)\downarrow 1]\!](\xi_1\downarrow(m+2)\downarrow 2)=\mathscr{M}_{n+m}[\![\xi_1\downarrow(m+1)\downarrow 1]\!](\xi_1\downarrow(m+1)\downarrow 2)$ Hence $\mathscr{M}_{n+m}[\![\xi_1\downarrow(m+1)\downarrow 1]\!](\xi_1\downarrow(m+1)\downarrow 2)=\mathscr{M}_n[\![M_1]\!]\rho_1$ whenever $n\geq 0$ and $0\leq m\leq \#\xi_1-\#\xi_0$; because $\xi_1\downarrow(\#\xi_1-\#\xi_0+1)=\xi_0\downarrow 1$ it is now possible to infer that $\mathscr{M}[\![M_0]\!]\rho_0=\mathscr{M}[\![M_1]\!]\rho_1$.

Should M_1 not be a member of Mod the nature of ensures that $\mathscr{M}[\![M_1]\!]\rho_1$ is proper. Should M_1 belong to Ind, however, either $kept(\langle\xi_1\downarrow 1\rangle\S\xi_0\dagger 1)(\langle\xi_1\downarrow 1\rangle\S\xi_0\dagger 1)=true$ or for some m having $1\leq m\leq \#\xi_1-\#\xi_0$ $kept\langle\xi_1\downarrow 1,\xi_1\downarrow(m+1)\rangle\langle\xi_1\downarrow 1,\xi_1\downarrow(m+1)\rangle=true$. A simple structural induction is enough to establish that any pairs $\langle M_2,\rho_2\rangle$ and $\langle M_3,\rho_3\rangle$ subject to $kept\langle\langle M_2,\rho_2\rangle,\langle M_3,\rho_3\rangle\rangle\langle\langle M_2,\rho_2\rangle,\langle M_3,\rho_3\rangle\rangle=true$ are such that $\mathscr{M}_n[\![M_2]\!]\rho_2=\mathscr{M}_n[\![M_3]\!]\rho_2$ for all $n\geq 0$. Consequently if $kept\langle\xi_1\downarrow 1,\xi_1\downarrow(m+1)\rangle\langle\xi_1\downarrow 1,\xi_1\downarrow(m+1)\rangle=true$ for a certain m satisfying $1\leq m\leq \#\xi_1-\#\xi_0$ then $\mathscr{M}_{n+m}[\![M_1]\!]\rho_1=\mathscr{M}_n[\![M_1]\!]\rho_1$ for all $n\geq 0$; in this situation $\mathscr{M}_n[\![M_1]\!]\rho_1=\perp$ for all $n\geq 0$ and so $\mathscr{M}[\![M_0]\!]\rho_0=\perp$.✠

This proposition enables us to connect the soft contexts described in 3.6.2 and 3.6.3. Suppose that $\mathfrak{f}:M\to T$ and $\mathfrak{F}:[Mod\times U]\to T$ are related in such a way that for all pairs $\langle M,\rho\rangle$ $\mathfrak{f}\perp=\perp$ and $\mathfrak{f}(\mathscr{M}[\![M]\!]\rho)=\mathfrak{F}\langle M,\rho\rangle$ unless M is a member of Ind or $slimp\rho=false$. For each pair $\langle T,\rho\rangle$ having $slimp\rho=true$ and $\#\rho[\![T]\!]>0$ either $wind\langle\langle T,\rho\rangle\rangle\downarrow 1\downarrow 1$ is a member of Ind and $\mathscr{M}[\![T]\!]\rho=\perp$ or $wind\langle\langle T,\rho\rangle\rangle\downarrow 1\downarrow 1$ is not a member of Ind. As a result every M:Mod and ρ:U for which $slimp\rho=true$ and $\#\rho[\![T]\!]>0$ when T:Ind give rise to equivalent coercions: if $\hat{\gamma}$ is an element of the domain G appropriate to M and if $\check{\gamma}$ is

drawn from the domain appropriate to Mod×U in such a way that $g\langle \acute{\gamma},\grave{\gamma}\rangle = true$ for some relation g then $g\langle soften\grave{\pi}(\mathbb{M}\rrbracket M\rrbracket\rho,\acute{\gamma}),soften\grave{\pi}(M,\rho,\grave{\gamma})\rangle = true$ provided that $g\langle \perp,\perp\rangle = true$.

Unfortunately a knowledge of the properties of *wind* is not sufficient to allow us to handle the predicates required in strong contexts. Accordingly we must now give a similar account of the properties of *bind*. For the purposes of this account an indication T will be said to 'occur' in a mode M if $T = lug M\downarrow\nu$ for some ν having $1\leq\nu\leq\#lug M$.

3.6.5. Proposition.

Let ξ_0 and ξ_1 be members of [Mod×U]*; set $\langle M_0,\rho_0\rangle = \xi_0\downarrow 1$, $\mu_0 = \llbracket M_0\rrbracket\rho_0$, $\langle M_1,\rho_1\rangle = \xi_1\downarrow 1$ and $\mu_1 = \llbracket M_1\rrbracket\rho_1$, and suppose that $\#\xi_0 = \#\xi_1$ and that $slim\rho_0 \wedge slim\rho_1 = true$. Assume also that if n is 0 or 1 then $\#\rho_n\llbracket T_1\rrbracket > 0$ whenever T_1:Ind is an indication which occurs in M_n or in $\rho_n\llbracket T_0\rrbracket\downarrow\nu_0\downarrow 1$ for any T_0:Ind and ν_0:N. There is some ν_0:N depending only on $\langle M_0,\rho_0\rangle$ and $\langle M_1,\rho_1\rangle$ such that $equal\nu_0\mu_0\mu_1 = \bigwedge\{equal\nu\mu_0\mu_1 \mid \nu:N\}$; moreover $bind 0(\langle M_0,\rho_0\rangle)(\langle M_1,\rho_1\rangle)0\downarrow 3$ is 0, 1 or 2 precisely when $equal\nu_0\mu_0\mu_1$ is *true*, *false* or $\perp$.

⊲During the course of this proof $\langle M_0,\rho_0\rangle$ and $\langle M_1,\rho_1\rangle$ will be taken to be fixed pairs satisfying the conditions above. For any lists ξ_0 and ξ_1 in [Mod×U]* the sequence μ^*_n will be taken to be $fix(\lambda\phi\nu.\nu>\#\xi_0\rightarrow\langle\rangle,\langle\mathbb{M}\rrbracket\xi_0\downarrow\nu\downarrow 1\rrbracket(\xi_0\downarrow\nu\downarrow 2)\rangle \S\phi(\nu+1))1$ if n is 0 or 1. Given any ν_1:N ξ_0 and ξ_1 will be said to be 'closed above' ν_1 when they are constrained thus: $\xi_0\downarrow\#\xi_0 = \langle M_0,\rho_0\rangle$, $\xi_1\downarrow\#\xi_1 = \langle M_1,\rho_1\rangle$, $\#\xi_0 = \#\xi_1$,
$\bigwedge\{equal\nu\mu_0\mu_1 \mid \nu:N\} = \bigwedge\{\bigwedge\{equal\nu(\mu^*_0\downarrow\nu_0)(\mu^*_1\downarrow\nu_0) \mid \nu:N\} \mid 1\leq\nu_0\leq\#\xi_0\}$,
for all ν_0:N with $\nu_1+1\leq\nu_0\leq\#\xi_0$ there are $\nu_1,\ldots,\nu_m$ such that
$equal\nu(\mu^*_0\downarrow\nu_0)(\mu^*_1\downarrow\nu_0) = \bigwedge\{equal(n=0\rightarrow 1,\nu-1)(\mu^*_0\downarrow\nu_n)(\mu^*_1\downarrow\nu_n) \mid 0\leq n\leq m\}$
for every $\nu\geq 1$, and for all ν_2:N with $1\leq\nu_2\leq\#\xi_n$ (if n is 0 or 1)

either there is some ν_3:N having $\xi_n\downarrow\nu_2=\langle lugM_n\downarrow\nu_3,\rho_n\rangle$ or there are ν_4:N, ν_5:N and T:Ind having $\xi_n\downarrow\nu_2=\langle lug(\rho_n[\![T]\!]\downarrow\nu_4\downarrow 1)\downarrow\nu_5, dip(\rho_n[\![T]\!]\downarrow\nu_4\downarrow 2)\rho_n\rangle$. For all such ξ_0 and ξ_1 the integer $j\nu_1\xi_0\xi_1$ will be taken to be the difference between $\prod\{\#(lugM_n)+\sum\{\sum\{1\le\nu\le\#\rho_n[\![T]\!]\to\#(lug(\rho_n[\![T]\!]\downarrow\nu\downarrow 1)),0\,|\,T:Ind\}\,|\,\nu:N\}\,|\,0\le n\le 1\}$ and the number of 'essentially different' members of $\langle\xi_0\dagger\nu_1,\xi_1\dagger\nu_1\rangle$, which is $(fix(\lambda\phi\nu.\nu\le\#\xi_0\to((kent(\xi_0\dagger\nu)(\xi_1\dagger\nu)\to 0,1)+\phi(\nu+1)),0)\nu_1)$. It will also be convenient to let $k\nu_1\xi_0\xi_1$ be 0, 1 or 2 corresponding to whether $\wedge\{\nu_1+1\le\nu\le\#\xi_0\to equal1(\ll[\![\xi_0\downarrow\nu\downarrow 1]\!](\xi_0\downarrow\nu\downarrow 2))(\ll[\![\xi_1\downarrow\nu\downarrow 1]\!](\xi_1\downarrow\nu\downarrow 2)),true\,|\,\nu:N\}$ is *true*, *false* or $\perp$; by induction on the structure of modes it can be seen that for no ν_0:N can $equal\nu_0(\ll[\![M_2]\!]\rho_2)(\ll[\![M_3]\!]\rho_3)$ be T when $\langle M_2,\rho_2\rangle$ and $\langle M_3,\rho_3\rangle$ satisfy the conditions imposed on $\langle M_0,\rho_0\rangle$ and $\langle M_1,\rho_1\rangle$, so $k\nu_1\xi_0\xi_1$ is well-defined.

Consider the following hypothesis: if ν_0, ν_1, ξ_0 and ξ_1 are such that ξ_0 and ξ_1 are closed above ν_1 and $j\nu_1\xi_0\xi_1<\nu_0$ then, writing $\langle\xi_2,\xi_3,\varepsilon_1\rangle$ for $bind\nu_1\xi_0\xi_1(k\nu_1\xi_0\xi_1)$, ξ_2 and ξ_3 are closed above 0 and $\varepsilon_1=k0\xi_2\xi_3$. On the assumption that this hypothesis holds for one particular ν_0:N it will be shown to hold for ν_0+1 by analysing the possible cases arising from the definition of *bend*. Let ξ_0 and ξ_1 be any lists in [Mod×U]* which are closed above ν_1 for some ν_1:N having $j\nu_1\xi_0\xi_1\le\nu_0+1$.

If $\nu_1<1$, $bind\nu_1\xi_0\xi_1\varepsilon=\langle\xi_0,\xi_1,\varepsilon\rangle$ for all ε, so $bind\nu_1\xi_0\xi_1(k\nu_1\xi_0\xi_1)\downarrow 3=k0(bind\nu_1\xi_0\xi_1(k\nu_1\xi_0\xi_1)\downarrow 1)(bind\nu_1\xi_0\xi_1(k\nu_1\xi_0\xi_1)\downarrow 2)$

If $kept(\xi_0\dagger(\nu_1-1))(\xi_1\dagger(\nu_1-1))=true$, by induction on ν_1 it can be seen that there is some ν_2 with $0\le\nu_2\le\nu_1-1$ for which $kept(\langle\xi_0\downarrow\nu_3\rangle\S\xi_0\dagger\nu_1)(\langle\xi_1\downarrow\nu_3\rangle\S\xi_1\dagger\nu_1)=true$ whenever $\nu_2+1\le\nu_3\le\nu_1$ and for which $kept(\langle\xi_0\downarrow\nu_2\rangle\S\xi_0\dagger\nu_1)(\langle\xi_0\downarrow\nu_2\rangle\S\xi_1\dagger\nu_1)=false$ unless $\nu_2=0$; moreover $bind\nu_1\xi_0\xi_1\varepsilon=bind\nu_2\xi_0\xi_1\varepsilon$ for every ε:T. Should $kept(\langle\xi_0\downarrow\nu_3\rangle\S\xi_0\dagger\nu_1)(\langle\xi_1\downarrow\nu_3\rangle\S\xi_1\dagger\nu_1)$ be *true* there must be some

ν_4:N having $\nu_1+1\leq\nu_4\leq\#\xi_0$ and
$kept(\xi_0\downarrow\nu_3,\xi_1\downarrow\nu_4)(\xi_1\downarrow\nu_3,\xi_1\downarrow\nu_4)=true$; an induction on the structure of $\xi_0\downarrow\nu_3\downarrow1$ and $\xi_1\downarrow\nu_3\downarrow1$ will establish that $\mathcal{M}[\![\xi_n\downarrow\nu_3\downarrow1]\!](\xi_n\downarrow\nu_3\downarrow2)=\mathcal{M}[\![\xi_n\downarrow\nu_4\downarrow1]\!](\xi_n\downarrow\nu_4\downarrow2)$ (if n is 0 or 1). The definition of *equal* in 3.6.2 therefore ensures that $k\nu_1\xi_0\xi_1=k\nu_2\xi_0\xi_1$. If $\nu_2=0$, $bind\nu_1\xi_0\xi_1(k\nu_1\xi_0\xi_1)\downarrow3=k0\xi_0\xi_1$ in accordance with the remarks above; if $\nu_2>0$, $bind\nu_1\xi_0\xi_1(k\nu_1\xi_0\xi_1)$ coincides with $bind\nu_2\xi_0\xi_1(k\nu_2\xi_0\xi_1)$ where $j\nu_2\xi_0\xi_1=\nu_0+1$, $\nu_2<\nu_1$ and $kept(\xi_0\dagger(\nu_2-1))(\xi_1\dagger(\nu_2-1))=false$. Thus this case can be subsumed under those dealt with below, in which it will be taken for granted that $kept(\xi_0\dagger(\nu_1-1))(\xi_1\dagger(\nu_1-1))=false$. Henceforth $\langle M_{n+2},\rho_{n+2}\rangle$ and μ_{n+2} will be written for $wind(\xi_n\downarrow\nu_1)\downarrow1$ and $\mathcal{M}[\![\xi_n\downarrow\nu_1\downarrow1]\!](\xi_n\downarrow\nu_1\downarrow2)$ respectively (if n is 0 or 1); 3.6.4 implies that $\mu_2=\mathcal{M}[\![M_2]\!]\rho_2$ and that $\mu_3=\mathcal{M}[\![M_3]\!]\rho_3$. In addition $\langle\xi_2,\xi_3,\varepsilon_1\rangle$ will be defined to be $bind\nu_1\xi_0\xi_1$.

If either M_2 or M_3 belongs to Ide, $bind\nu_1\xi_0\xi_1\varepsilon$ is $bind(\nu_1-1)\xi_0\xi_1 2$ for all ε. By 3.6.4 neither μ_2 nor μ_3 is τ but at least one is $\perp$, so $equal1\mu_2\mu_3=\perp$; in consequence $k(\nu_1-1)\xi_0\xi_1=2$ and $\langle\xi_2,\xi_3,\varepsilon_1\rangle=bind(\nu_1-1)\xi_0\xi_1(k(\nu_1-1)\xi_0\xi_1)$. As $j(\nu_1-1)\xi_0\xi_1=\nu_0$ the induction hypothesis ensures that the lists $\xi_2$and ξ_3 are closed above 0 and have $\varepsilon_1=k0\xi_2\xi_3$.

If M_2 and M_3 are both bool or both char, (and if $kept(\xi_0\dagger(\nu_1-1))(\xi_1\dagger(\nu_1-1))=false$) then $k\nu_1\xi_0\xi_1=k(\nu_1-1)\xi_0\xi_1$ and, from the definition of *bind* in 3.6.3, $bind\nu_1\xi_0\xi_1\varepsilon=bind(\nu_1-1)\xi_0\xi_1\varepsilon$ for all ε. Hence $\langle\xi_2,\xi_3,\varepsilon_1\rangle=bind(\nu_1-1)\xi_0\xi_1(k(\nu_1-1)\xi_0\xi_1)$; the induction hypothesis is applicable to $bind(\nu_1-1)\xi_0\xi_1(k(\nu_1-1)\xi_0\xi_1)$ since $j(\nu_1-1)\xi_0\xi_1=\nu_0$, so ξ_2 and ξ_3 are closed above 0 and have $\varepsilon_1=k0\xi_2\xi_3$.

The proof needed when M_2 and M_3 are in {proc}°×Mod will be omitted owing to its affinity with that required when M_2 and M_3

are members of {struct}°×[Mod×Sel]*.

If M_2 and M_3 belong to {struct}°×[Mod×Sel]*, suppose first that $\#(M_2\downarrow 2)$ and $\#(M_3\downarrow 3)$ are different. In this situation $bind\nu_1\xi_0\xi_1(k\nu_1\xi_0\xi_1)=bind(\nu_1\text{-}1)\xi_0\xi_1((k\nu_1\xi_0\xi_1)\vee 1)$ and $(k\nu_1\xi_0\xi_1)\vee 1=k(\nu_1\text{-}1)\xi_0\xi_1$ as $equal 1\mu_2\mu_3=false$; consequently $\langle\xi_2,\xi_3,\varepsilon_1\rangle=bind(\nu_1\text{-}1)\xi_0\xi_1(k(\nu_1\text{-}1)\xi_0\xi_1)$ and the argument of the preceding paragraphs can be applied. Similar remarks are pertinent if $\#(M_2\downarrow 2)$ equals $\#(M_3\downarrow 2)$ but the selectors of the modes do not correspond. Finally suppose that neither of these faults arises, so that $bind\nu_1\xi_0\xi_1\varepsilon$ is $bind\nu_2\xi_4\xi_5\varepsilon$ where ν_2 is $\nu_1\text{-}1+\#(M_n\downarrow 2)$ and where $\xi_n=(\langle M_n\downarrow 2\downarrow 1\downarrow 1,\rho_n\rangle,\ldots,\langle M_n\downarrow 2\downarrow\#(M_n\downarrow 2)\downarrow 1,\rho_n\rangle)\S\xi_{n-4}$ if n is 4 or 5. The definition of *equal* given in 3.6.2 ensures that ξ_4 and ξ_5 are closed above ν_2 while $j\nu_2\xi_4\xi_3=\nu_0$, so that the induction hypothesis is relevant to $bind\nu_2\xi_4\xi_3(k\nu_2\xi_4\xi_3)$; as $k\nu_2\xi_4\xi_3=k\nu_1\xi_0\xi_1$, ξ_2 and ξ_3 are closed above 0 and $\varepsilon_1=k0\xi_0\xi_1$.

This completes as much of the induction as need be given; it remains only to note that the result holds when $j\nu_1\xi_0\xi_1=0$ (because then $bind\nu_1\xi_0\xi_1\varepsilon=bind0\xi_0\xi_1\varepsilon=\varepsilon$ for all ε) and so it holds whatever value $j\nu_1\xi_0\xi_1$ may take.

Let $\langle\xi_2,\xi_3,\varepsilon_1\rangle$ be $bind1(\langle M_0,\rho_0\rangle)(\langle M_1,\rho_1\rangle)0$, so that ξ_2 and ξ_3 are closed above 0 and $\varepsilon_1=k0\xi_2\xi_3$. When n is 2 or 3 write μ^*_n for $fix(\lambda\phi\nu.\nu>\#\xi_n\rightarrow(),(\mathbb{M}[\![\xi_n\downarrow\nu\downarrow 1]\!](\xi_n\downarrow\nu\downarrow 2))\S\phi(\nu+1))1$; as $\mu^*_n\downarrow\nu_3=\mu^*_n\downarrow\nu_4$ when $kept\langle\xi_2\downarrow\nu_3,\xi_2\downarrow\nu_4\rangle\langle\xi_3\downarrow\nu_3,\xi_3\downarrow\nu_4\rangle=true$, there are at most $j0()()$ distinct pairs of the form $\langle\mu^*_2\downarrow\nu_0,\mu^*_3\downarrow\nu_0\rangle$. For every ν_0 having $1\le\nu_0\le\#\xi_2$, there is some $m\ge 0$ for which certain $\nu_1,\ldots,\nu_m$ are such that for all $\nu\ge 1$ $equal\nu(\mu^*_2\downarrow\nu_0)(\mu^*_3\downarrow\nu_0)$ coincides with $\wedge\{equal(n=0\rightarrow 1,\nu\text{-}1)(\mu^*_2\downarrow\nu_n)(\mu^*_3\downarrow\nu_n)\mid 0\le n\le m\}$; indeed induction on ν even demonstrates that $\nu_1,\ldots,\nu_m$ can be made to satisfy $kept\langle\xi_2\downarrow\nu_r,\xi_2\downarrow\nu_s\rangle\langle\xi_3\downarrow\nu_r,\xi_3\downarrow\nu_s\rangle=false$ when $0\le r\le s\le m$. Hence, by induction on k, for every ν_0 with $1\le\nu_0\le\#\xi_2$ there are $\nu_1,\ldots,\nu_m$

and an integer k having $k-1\le l$ if $l<m$ such that $equal\nu(\mu^*_2\downarrow\nu_0)(\mu^*_3\downarrow\nu_0)$ is $\bigwedge\{equal(0\le n\le l\to 1,\nu-k)(\mu^*_2\downarrow\nu_n)(\mu^*_3\downarrow\nu_n)\mid 0\le n\le m\}$ whenever $\nu\ge k$; furthermore $kept\langle\xi_2\downarrow\nu_r,\xi_2\downarrow\nu_s\rangle\langle\xi_3\downarrow\nu_r,\xi_3\downarrow\nu_s\rangle=false$ if $0\le r\le s\le m$. In particular, setting $k=j0\langle\rangle\langle\rangle$ shows that $equal\nu(\mu^*_2\downarrow\nu_0)(\mu^*_3\downarrow\nu_0)=\bigwedge\{equal1(\mu^*_2\downarrow\nu_n)(\mu^*_3\downarrow\nu_n)\mid 0\le n\le m\}$ for certain $\nu_1,\ldots,\nu_m$ and for all $\nu\ge k$. Moreover, since ξ_2 and ξ_3 are closed above 0,

$\bigwedge\{equal\nu\mu_0\mu_1\mid\nu:N\}=\bigwedge\{\bigwedge\{equal\nu(\mu^*_2\downarrow\nu_0)(\mu^*_3\downarrow\nu_0)\mid\nu:N\}\mid 1\le\nu_0\le\#\xi_2\}$.

Hence $equal(1+(j0\langle\rangle\langle\rangle))\mu_0\mu_1$, $\bigwedge\{equal\nu\mu_0\mu_1\mid\nu:N\}$ and $\bigwedge\{equal1(\mu^*_2\downarrow\nu_0)(\mu^*_3\downarrow\nu_0)\mid 1\le\nu_0\le\#\xi_2\}$ coincide. In conjunction with the fact that $\varepsilon_1=k0\xi_2\xi_3$ this ensures that $bind1\langle\langle M_0,\rho_0\rangle\rangle\langle\langle M_1,\rho_1\rangle\rangle 0\downarrow 3$ is 0, 1 or 2 precisely when $\bigwedge\{equal\nu(\mathscr{M}[\![M_0]\!]\rho_0)(\mathscr{M}[\![M_1]\!]\rho_1)\mid\nu:N\}$ is $true$, $false$ or $\perp$.▷

There are two forms of the shielding conditions of the Algol 68 report which correspond in the same way as *equal* and *bind*. The proof of this will be ignored because it does not differ in important respects from that above. Thus now it will be shown that $\mathscr{M}$ mimics $\mathscr{N}$ in the appropriate manner.

3.6.6. Proposition.

Let ρ be an environment having $slim\rho=true$, and suppose that for every T:Ind and ν:N such that $1\le\nu\le\#\rho[\![T]\!]$ $wind\langle\rho[\![T]\!]\downarrow\nu\downarrow 1,dip(\rho[\![T]\!]\downarrow\nu\downarrow 2)\rho\rangle\downarrow 1\downarrow 1$ is not a member of Ind. For all M:Mod $\mathscr{N}[\![M]\!](\bigsqcup\{turn\nu\rho\mid\nu:N\})=\mathscr{M}[\![M]\!]\rho$.

◁The restrictions imposed on ρ are such that induction on the size of ρ ensures that $turn(1+\sum\{\#\rho[\![T]\!]\mid T:Ind\})\rho$ is proper, so that, more significantly, the environment $\bigsqcup\{turn\nu\rho\mid\nu:N\}$ is a proper element of $[Ide\mapsto[M\times D^\circ]^*]\times[Ind\mapsto[M\times G]^*]$. By virtue of the 'shielding' conditions and the fact that for all Δ:Dec $slim(\mathscr{X}[\![\Delta]\!]\rho)=true$ whenever $slim\rho=true$ any valid Algol 68 program obeys these restrictions; were this not so the proof of the present

result would require the domain of environments appropriate to M to be $[Ide \to [M^\circ \times D^\circ]^*] \times [Ind \to [M^\circ \times D^\circ]^*]$.

The truncated versions of $\mathscr{M}$ set up in 3.6.4 are manifestly such that for all M:Mod $\mathscr{N}[\![M]\!](turn\rho) \sqsupseteq \mathscr{M}_0[\![M]\!]\rho$. Assume that for some $n \geq 0$ every M:Mod satisfies $\mathscr{N}[\![M]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\}) \sqsupseteq \mathscr{M}_n[\![M]\!]\rho$. For any T:Ind it is plain that $\mathscr{N}[\![T]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\}) \sqsupseteq \mathscr{M}_{n+1}[\![T]\!]\rho$ when $\#\rho[\![T]\!]=0$; moreover, when $\#\rho[\![T]\!]>0$

$$\begin{aligned}\mathscr{N}[\![T]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\}) &= (\bigsqcup\{turn\nu\rho \mid \nu:N\})[\![T]\!]\dagger 1\dagger 1\\ &= \mathscr{M}[\![\rho[\![T]\!]\dagger 1\dagger 1]\!](\bigsqcup\{turn\nu(dip(\rho[\![T]\!]\dagger 1\dagger 2)\rho) \mid \nu:N\})\\ &\sqsupseteq \mathscr{M}_n[\![\rho[\![T]\!]\dagger 1\dagger 1]\!](dip(\rho[\![T]\!]\dagger 1\dagger 2)\rho)\\ &= \mathscr{M}_{n+1}[\![T]\!]\rho.\end{aligned}$$

Similar inequalities can be established for the other modes so by induction $\mathscr{N}[\![M]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\}) \sqsupseteq \mathscr{M}_{n+1}[\![M]\!]\rho$ for every M:Mod and for every $n \geq 0$. Hence by the definition of $\mathscr{M}_n$ the valuations satisfy $\mathscr{N}[\![M]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\}) \sqsupseteq \mathscr{M}[\![M]\!]\rho$.

When showing that $\mathscr{N}[\![M]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\}) \sqsubseteq \mathscr{M}[\![M]\!]\rho$ use can be made of inclusive predicate u defined on environments $\acute{\rho}$ and $\grave{\rho}$ with $\acute{\rho}$ in from $[Ide \to [M \times D^\circ]^*] \times [Ind \to [M \times G]^*]$ and with $\grave{\rho}$ in $[Ide \to [Mod \times [Ind \to N] \times D^\circ]^*] \times [Ind \to [Mod \times [Ind \to N] \times G]^*]$. It is natural to demand that

$$u = \lambda(\acute{\rho},\grave{\rho}).\bigwedge\{(1 \leq \nu \leq \#\acute{\rho}[\![T]\!] \to (\acute{\rho}[\![T]\!]\dagger\nu\dagger 1 \sqsubseteq \mathscr{M}[\![\grave{\rho}[\![T]\!]\dagger\nu\dagger 1]\!](dip(\grave{\rho}[\![T]\!]\dagger\nu\dagger 2)\grave{\rho})), true)$$
$$\wedge (\#\acute{\rho}[\![T]\!] = \#\grave{\rho}[\![T]\!]) \mid T:Ind\}$$

so that by an obvious structural induction whenever $u(\acute{\rho},\grave{\rho}) = true$ $\mathscr{N}[\![M]\!]\acute{\rho} \sqsubseteq \mathscr{M}[\![M]\!]\grave{\rho}$ for every M:Mod. For the environment ρ mentioned in the statement of the proposition $u(turn0\rho,\rho) = true$. Moreover, for any ν_0:N, ν_1:N and T:Ind such that $u(turn\nu_0\rho,\rho) = true$ and $1 \leq \nu_1 \leq \#\rho[\![T]\!]$, $u(turn\nu_0(dip(\rho[\![T]\!]\dagger\nu_1\dagger 2)\rho), dip(\rho[\![T]\!]\dagger\nu_1\dagger 2)\rho) = true$ and

$$\begin{aligned}turn(\nu_0+1)\rho[\![T]\!]\dagger\nu_1\dagger 1 &= \mathscr{M}[\![\rho[\![T]\!]\dagger\nu_1\dagger 1]\!](turn\nu_0(dip(\rho[\![T]\!]\dagger\nu_1\dagger 2)\rho))\\ &\sqsubseteq \mathscr{M}[\![\rho[\![T]\!]\dagger\nu_1\dagger 1]\!](dip(\rho[\![T]\!]\dagger\nu_1\dagger 2)\rho),\end{aligned}$$

so that $u(turn(\nu_0+1)\rho,\rho) = true$. Proceeding to the limit, $u(\bigsqcup\{turn\nu\rho \mid \nu:N\},\rho) = true$ and $\mathscr{N}[\![M]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\}) \sqsubseteq \mathscr{M}[\![M]\!]\rho$ for all

modes M belonging to the domain Mod. Together with the result established in the preceding paragraph this ensures that for every M:Mod the elements $\mathcal{M}[\![M]\!](\bigsqcup\{turn\nu\rho \mid \nu:N\})$ and $\mathcal{M}[\![M]\!]\rho$ of M must coincide.⊳

The procedures adopted in the proposition above have ramifications far beyond the realm of Algol 68. In particular, the fascinating theorems due to Gordon [6] can be given short proofs by introducing valuations akin to $\mathcal{M}_n$ and inclusive predicates like u. The major difference between these proofs and that above arises from the requirement that the substitutes for the domains labelled U contain components more closely analogous to Ind↦[[U→M]×G]* and Ind↦[Mod×G]* than they are to Ind↦[M×G]* and Ind↦[Mod×[Ind↦N]×G]*. This necessitates the provision of a self-referential version of u which can loosely be said to satisfy the equation

$$u=\lambda(\vec{\rho},\vec{\rho}).\bigwedge\{(1\le\nu\le\#\vec{\rho}[\![T]\!]\wedge u(\vec{\rho}_0,\vec{\rho}_0)\to((\vec{\rho}[\![T]\!]\downarrow\nu\downarrow 1)\,\vec{\rho}_0\sqsubseteq\mathcal{M}[\![\vec{\rho}[\![T]\!]\downarrow\nu\downarrow 1]\!]\vec{\rho}_0),true)$$
$$\wedge(\#\vec{\rho}[\![T]\!]=\#\vec{\rho}[\![T]\!])\mid T:Ind\};$$

the use of such a predicate can be justified by an appeal to the general method of 2.2.8. As compensation for the extra labour involved in making this appeal the inductive arguments about *turn* may be deleted, since in effect their role is taken by the limiting process to which the appropriate predictor has to be subjected when u is constructed.

The techniques of the foregoing pages have long since been applied in a formal definition of Algol 68. Of greater interest than this is the application of propositions like that above to verify the equivalence of standard semantics and a variety of semantics which models stored-program machines by representing functions as finitary objects belonging to flat lattices. It may not be unfortunate that lack of space precludes the discussion of either of these topics.

INDEX

BIBLIOGRAPHY

[1] de Bakker, J.W.: *Recursive procedures*, Mathematical Centre Tract 24, Mathematisch Centrum, Amsterdam (1971).

[2] Burstall, R.M.: *Some techniques for proving the correctness of programs which alter data structures*, pages 23-50 of Machine Intelligence 7 (edited by Meltzer, B., and Michie, D.), Edinburgh University Press, Edinburgh (1972).

[3] Dahl, O.J.: *Discrete event simulation languages*, pages 249-295 of Programming Languages (edited by Genuys, F.), Academic Press, London (1968).

[4] Dijkstra, E.W.: *Recursive Programming*, pages 312-318 of Numerische Mathematik 2 (1960).

[5] Evans, A.: *A reference manual and primer for Pal*, Department of Electrical Engineering, Massachusetts Institute of Technology (1969).

[6] Gordon, M.J.C.: *Models of pure Lisp*, Experimental Programming Report 31, Theory of Computation Group, Edinburgh University (1973).

[7] Henhapl, W., and Jones, C.B.: *The block concept and some possible implementations with proofs of equivalence*, Technical Report 104, International Business Machines, Wien (1970).

[8] Hoare, C.A.R., and Wirth, N.: *An axiomatic definition of the programming language Pascal*, Bericht der Fachgruppe Computer-Wissenschaften 6, Eidgenössische Technische Hochschule, Zürich (1972).

[9] Knaster, B.: *Un théorème sur les fonctions d'ensembles*, pages 133-134 of Annales de la Societé Polonaise de Mathematique 6 (1928).

[10] Landin, P.J.: *The mechanical evaluation of expressions*, pages 308-320 of Computer Journal 6 (1964).

[11] Milner, A.J.R.G.: *Processes: a mathematical model of computing agents*, Colloquium in Mathematical Logic, Bristol (1973).

[12] Morris, F.L.: *Correctness of translations of programming languages*, Computer Science Memorandum 303, Department of Computer Science, Stanford University (1972).

[13] Naur, P., editor: *Report on the algorithmic language Algol 60*, pages 349-367 of Computer Journal 5 (1963).

[14] Park, D.M.R.: *The Y combinator in Scott's lambda-calculus models*, Symposium on Programming Theory, Warwick (1970).

[15] Plotkin, G.D.: *Lambda-definability and logical relations*, Artificial Intelligence Memorandum 4, School of Artificial Intelligence, Edinburgh University (1973).

[16] Reynolds, J.C.: *Definitional interpreters for higher-order programming languages*, pages 717-740 of Proceedings of the Twenty-Fifth Association for Computing Machinery National Conference, New York (1972).

[17] Scott, D.S.: *Outline of a mathematical theory of computation*, pages 169-176 of Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems, Princeton (1970).

[18] Scott, D.S.: *The lattice of flow diagrams*, pages 311-366 of Springer Lecture Notes in Mathematics 188 (1971).

[19] Scott, D.S.: *Continuous lattices*, pages 97-136 of Springer Lecture Notes in Mathematics 274 (1972).

[20] Scott, D.S., and Strachey, C.: *Towards a mathematical semantics for computer languages*, Microwave Research Institute Symposia Series Volume 21, Polytechnic Institute of Brooklyn (1971).

[21] Strachey, C.: *Towards a formal semantics*, pages 198-220 of Formal Language Description Languages for Computer Programming (edited by Steel, T.B.), North Holland Publishing Company, Amsterdam (1966).

[22] Strachey, C.: *Fundamental concepts in programming languages*, International Summer School in Computer Programming, København (1967).

[23] Strachey, C.: *The varieties of programming language*, pages 222-233 of Proceedings of the International Computing Symposium, Venezia (1972).

[24] Tarski, A.: *A lattice-theoretical fixpoint theorem and its applications*, pages 285-309 of Pacific Journal of Mathematics 5 (1955).

[25] Wadsworth, C.P.: *Another approach to jumps*, Programming Research Group, Oxford University (1970).

[26] van Wijngaarden, A., editor: *Report on the algorithmic language Algol 68*, pages 79-218 of Numerische Mathematik 14 (1969).

APPENDIX ONE

STANDARD SEMANTICS

β:V=B+L*+J+F	stored values
ε:E=L+B+L*+J+F	expressed values
δ:D=L+B+L*+J+F+G	denoted values
ω:W=L+B+L*+J+F+G+K°	witnessed values
β:B={*dummy*}°+T+N+R+H*	basic values
ε:T={*true*}°+{*false*}°	truth values
θ:J=C°	label closures
ϕ:F=[E→K→C]°	function closures
γ:G=[K→C]°	recursion closures
θ:C=S→A	command continuations
κ:K=E→C	expression continuations
χ:X=U→C	declaration continuations
ρ:U=[Ide↦D°*]×K°*	environments
σ:S=[L↦[T×V]]×V*×V*	stores
o:A	answers
α:L	locations
ν:N	integers
R	reals
H	characters

O:Mon	monadic operators
Ω:Dya	dyadic operators
I:Ide	identifiers
B:Bas	bases
Φ:Abs	abstractions
E:Exp	expressions
Δ:Dec	declarations

$\mathcal{O}$: Mon→E→B

$\mathcal{W}$: Dya→[E×E]→B

$\mathcal{B}$: Bas→E

$\mathcal{F}$: Abs→U→F

$\mathcal{E}$: Exp→U→K→C

$\mathcal{L}$: Exp→U→K→C

$\mathcal{R}$: Exp→U→K→C

$\mathcal{G}$: Exp→U→K→C

$\mathcal{P}$: Exp→U→K→J*

$\mathcal{Q}$: Exp→U→K→J*

$\mathcal{J}$: Exp→Ide*

$\mathcal{K}$: Exp→Ide*

$\mathcal{S}$: Dec→U→S→G*

$\mathcal{D}$: Dec→U→X→C

$\mathcal{T}$: Dec→U→X→C

$\mathcal{I}$: Dec→Ide*

$\mathcal{H}$: Dec→Ide*

Φ::=fn()E|fnI.E|fn$I_1,\ldots,I_n$.E|fnI..E|fn$I_1,\ldots,I_n$..E.

E::=I|B|Φ|OE|$E_0 \Omega E_1$|E_0:=E_1|$E_1,\ldots,E_n$:=E_0|get E|put E|E_0 aug E_1|
$E_1,\ldots,E_n$|\$E|E\$|£E|E£|$E_0 E_1$|val E|res E|goto E|Δ inside E|
E_0; E_1|if E_0 then E_1 else E_2|while E_0 do E_1|I:E|I::E|(E).

Δ::=I=E|$I_1,\ldots,I_n$=E|I==E|$I_1,\ldots,I_n$==E|Δ_0 within Δ_1|Δ_1 and...and Δ_n|
rec Δ|(Δ).

$$\mathscr{F}[\![\ \mathsf{fn}()E]\!] =$$

$$\lambda\rho.\lambda\varepsilon\kappa.rv(\lambda\beta.\#\beta\,|\,L^*=0\rightarrow\mathscr{L}[\![\ E]\!]\rho\kappa,\tau)\varepsilon.$$

$$\mathscr{F}[\![\ \mathsf{fn}I.E]\!] =$$

$$\lambda\rho.\lambda\varepsilon\kappa.\mathscr{L}[\![\ E]\!]\rho[\varepsilon/I]\kappa.$$

$$\mathscr{F}[\![\ \mathsf{fn}I_1,\ldots,I_n.E]\!] =$$

$$\lambda\rho.\lambda\varepsilon\kappa.rv(\lambda\beta.\#\beta\,|\,L^*=n\rightarrow\mathscr{L}[\![\ E]\!]\rho[\beta/\langle I_1,\ldots,I_n\rangle]\kappa,\tau)\varepsilon.$$

$$\mathscr{F}[\![\ \mathsf{fn}I..E]\!] =$$

$$\lambda\rho.\lambda\varepsilon\kappa.rv(\lambda\beta.\mathscr{L}[\![\ E]\!]\rho[\beta/I]\kappa)\varepsilon.$$

$$\mathscr{F}[\![\ \mathsf{fn}I_1,\ldots,I_n..E]\!] =$$

$$\lambda\rho.\lambda\varepsilon\kappa.rv(\lambda\beta\sigma.\#\beta\,|\,L^*=n\rightarrow\mathscr{L}[\![\ E]\!]\rho[holds\,\beta\sigma/\langle I_1,\ldots,I_n\rangle]\kappa\sigma,\tau)\varepsilon.$$

$$\mathscr{E}[\![\ E]\!] =$$

$$\lambda\rho\kappa\sigma.(\lambda\alpha^*.(\lambda\rho'.\alpha^*:L^*\rightarrow\mathscr{G}[\![\ E]\!]\rho'\kappa(updates\alpha^*(\mathscr{P}[\![\ E]\!]\rho'\kappa)\sigma),\tau)$$

$$(fix(\lambda\rho''.\rho[\alpha^*/\mathscr{J}[\![\ E]\!]][\mathscr{Q}[\![\ E]\!]\rho''\kappa/\mathscr{K}[\![\ E]\!]])))$$

$$(news(\#\mathscr{J}[\![\ E]\!])\sigma).$$

$$\mathscr{L}[\![\ E]\!] =$$

$$\lambda\rho\kappa.\mathscr{E}[\![\ E]\!]\rho(lv\kappa).$$

$$\mathscr{R}[\![\ E]\!] =$$

$$\lambda\rho\kappa.\mathscr{E}[\![\ E]\!]\rho(rv\kappa).$$

$$\mathscr{G}[\![\ I]\!] =$$

$$\lambda\rho\kappa.(\lambda\delta.\delta:G\rightarrow\delta\kappa,\kappa\delta)(\rho[\![\ I]\!]\downarrow 1).$$

$$\mathscr{G}[\![\ B]\!] =$$

$$\lambda\rho\kappa.\kappa(\mathscr{B}[\![\ B]\!]).$$

$$\mathscr{G}[\![\ \Phi]\!] =$$

$$\lambda\rho\kappa.\kappa(\mathscr{F}[\![\ \Phi]\!]\rho).$$

$$\mathscr{G}[\![\ OE]\!] =$$

$$\lambda\rho\kappa.\mathscr{R}[\![\ E]\!]\rho(\lambda\varepsilon.rv\kappa(\mathscr{O}[\![\ O]\!]\varepsilon)).$$

$$\mathscr{G}[\![E_0 \Omega E_1]\!] =$$

$$\lambda\rho\kappa.\mathit{run}\langle \mathscr{R}[\![E_0]\!]\rho, \mathscr{R}[\![E_1]\!]\rho\rangle\ (\lambda\varepsilon^*.\mathit{rv}\kappa(\mathscr{W}[\![\Omega]\!]\langle \varepsilon^*\downarrow 1, \varepsilon^*\downarrow 2\rangle)).$$

$$\mathscr{G}[\![E_0 := E_1]\!] =$$

$$\lambda\rho\kappa.\mathit{run}\langle \mathscr{L}[\![E_0]\!]\rho, \mathscr{R}[\![E_1]\!]\rho\rangle\ (\lambda\varepsilon^*\sigma.\kappa(\mathit{dummy})(\mathit{update}\,(\varepsilon^*\downarrow 1)\,(\varepsilon^*\downarrow 2)\sigma)).$$

$$\mathscr{G}[\![E_1, \ldots, E_n := E_0]\!] =$$

$$\lambda\rho\kappa.\,(\lambda\psi.\mathit{run}\langle \mathscr{R}[\![E_0]\!]\rho, \mathscr{L}[\![E_1]\!]\rho, \ldots, \mathscr{L}[\![E_n]\!]\rho\rangle\ \psi)$$

$$(\lambda\varepsilon^*\sigma.\#\varepsilon^*\downarrow 1 \mid L^* = n \to \kappa(\mathit{dummy})(\mathit{updates}\,(\varepsilon^*\dagger 1)(\mathit{holds}(\varepsilon^*\downarrow 1)\sigma)\sigma), \tau).$$

$$\mathscr{G}[\![\mathsf{get}\ E]\!] =$$

$$\lambda\rho\kappa.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon\sigma.\#(\sigma\downarrow 2) > 0 \to \kappa\varepsilon(\mathit{update}\varepsilon(\sigma\downarrow 2\downarrow 1)\langle \sigma\downarrow 1, (\sigma\downarrow 2)\dagger 1, \sigma\downarrow 3\rangle), \tau).$$

$$\mathscr{G}[\![\mathsf{put}\ E]\!] =$$

$$\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\sigma.\kappa\varepsilon\langle \sigma\downarrow 1, \sigma\downarrow 2, \langle\varepsilon\rangle \S (\sigma\downarrow 3)\rangle).$$

$$\mathscr{G}[\![E_0\ \mathsf{aug}\ E_1]\!] =$$

$$\lambda\rho\kappa.\mathit{run}\langle \mathscr{R}[\![E_0]\!]\rho, \mathscr{L}[\![E_1]\!]\rho\rangle\ (\lambda\varepsilon^*.\varepsilon^*\downarrow 1 : L^* \to \kappa((\varepsilon^*\downarrow 1) \S \langle \varepsilon^*\downarrow 2\rangle), \tau).$$

$$\mathscr{G}[\![E_1, \ldots, E_n]\!] =$$

$$\lambda\rho\kappa.\mathit{run}\langle \mathscr{L}[\![E_1]\!]\rho, \ldots, \mathscr{L}[\![E_n]\!]\rho\rangle\ (\lambda\varepsilon^*.\kappa(\varepsilon^* \mid L^*)).$$

$$\mathscr{G}[\![\$E]\!] =$$

$$\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(l\mathit{v}\kappa).$$

$$\mathscr{G}[\![E\$]\!] =$$

$$\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon.\varepsilon : F \to \kappa(\lambda\varepsilon'\kappa'.\mathit{rv}(\lambda\beta.(\varepsilon \mid F)\beta\kappa')\varepsilon'), \tau).$$

$$\mathscr{G}[\![£E]\!] =$$

$$\lambda\rho\kappa.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\sigma.(\lambda\alpha^*.\alpha^* : L^* \to \kappa\alpha^*(\mathit{updates}\,\alpha^*(\mathit{holds}\varepsilon\sigma)\sigma), \tau)(\mathit{news}(\#\varepsilon \mid L^*)\sigma)).$$

$$\mathscr{G}[\![E£]\!] =$$

$$\lambda\rho\kappa.(\lambda\kappa'.\mathscr{R}[\![E]\!]\rho\kappa')$$

$$(\lambda\varepsilon.(\lambda\phi.\varepsilon : F \to \kappa(\lambda\varepsilon''\kappa''.\mathit{rv}(\lambda\beta.\phi\beta\kappa'')\varepsilon''), \tau)$$

$$(\lambda\varepsilon''\kappa''\sigma''.(\lambda\alpha^*.\alpha^* : L^* \to (\varepsilon \mid F)\alpha^*\kappa''(\mathit{updates}\,\alpha^*(\mathit{holds}\varepsilon''\sigma'')\sigma''), \tau)$$

$$(\mathit{news}(\#\varepsilon'' \mid L^*)\sigma''))).$$

$$\mathscr{G}[\![E_0 E_1]\!] =$$

$$\lambda\rho\kappa.\,(\lambda\psi.\mathit{run}\langle \mathscr{R}[\![E_0]\!]\rho, \mathscr{L}[\![E_1]\!]\rho\rangle\ \psi)$$

$$(\lambda\varepsilon^*.(\lambda\kappa'.\varepsilon^*\downarrow 1 : F \to (\varepsilon^*\downarrow 1 \mid F)\,(\varepsilon^*\downarrow 2)\kappa, \mathit{rv}\kappa'(\varepsilon^*\downarrow 2))$$

$$(\lambda\varepsilon'.1 \le \varepsilon' \mid N \le \#\varepsilon^*\downarrow 1 \mid L^* \to \kappa(\varepsilon^*\downarrow 1 \downarrow \varepsilon'), \tau)).$$

$$\mathscr{G}[\![\,\texttt{val}\ E\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{L}[\![\,E\,]\!]\rho[\kappa/\texttt{res}\,]\kappa.$$

$$\mathscr{G}[\![\,\texttt{res}\ E\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{L}[\![\,E\,]\!]\rho(\rho[\![\,\texttt{res}\,]\!]\downarrow 1\,|\,K).$$

$$\mathscr{G}[\![\,\texttt{goto}\ E\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{R}[\![\,E\,]\!]\rho(\lambda\varepsilon.\varepsilon\,|\,C).$$

$$\mathscr{G}[\![\,\Delta\ \texttt{inside}\ E\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{D}[\![\,\Delta\,]\!]\rho(\lambda\rho'.\mathscr{L}[\![\,E\,]\!](\mathit{divert}\rho\rho')\kappa).$$

$$\mathscr{G}[\![\,E_0;\ E_1\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{G}[\![\,E_0\,]\!]\rho(\lambda\varepsilon.\mathscr{G}[\![\,E_1\,]\!]\rho\kappa).$$

$$\mathscr{G}[\![\,\texttt{if}\ E_0\ \texttt{then}\ E_1\ \texttt{else}\ E_2\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{R}[\![\,E_0\,]\!]\rho(\lambda\varepsilon.\varepsilon\,|\,\mathsf{T}\rightarrow\mathscr{G}[\![\,E_1\,]\!]\rho\kappa,\mathscr{G}[\![\,E_2\,]\!]\rho\kappa).$$

$$\mathscr{G}[\![\,\texttt{while}\ E_0\ \texttt{do}\ E_1\,]\!] =$$

$$\lambda\rho\kappa.\mathit{fix}(\lambda\theta.\mathscr{R}[\![\,E_0\,]\!]\rho(\lambda\varepsilon.\varepsilon\,|\,\mathsf{T}\rightarrow\mathscr{G}[\![\,E_1\,]\!]\rho(\lambda\varepsilon.\theta),\kappa(\mathit{dummy}))).$$

$$\mathscr{G}[\![\,I{:}E\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{G}[\![\,E\,]\!]\rho\kappa.$$

$$\mathscr{G}[\![\,I{::}E\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{G}[\![\,E\,]\!]\rho\kappa.$$

$$\mathscr{G}[\![\,(E)\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{G}[\![\,E\,]\!]\rho\kappa.$$

$$\mathscr{P}[\![\,E_0;\ E_1\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{P}[\![\,E_0\,]\!]\rho(\lambda\varepsilon.\mathscr{G}[\![\,E_1\,]\!]\rho\kappa)\,\S\,\mathscr{P}[\![\,E_1\,]\!]\rho\kappa.$$

$$\mathscr{P}[\![\,\texttt{if}\ E_0\ \texttt{then}\ E_1\ \texttt{else}\ E_2\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{P}[\![\,E_1\,]\!]\rho\kappa\,\S\,\mathscr{P}[\![\,E_2\,]\!]\rho\kappa.$$

$$\mathscr{P}[\![\,\texttt{while}\ E_0\ \texttt{do}\ E_1\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{P}[\![\,E_1\,]\!]\rho(\lambda\varepsilon.\mathscr{G}[\![\,\texttt{while}\ E_0\ \texttt{do}\ E_1\,]\!]\rho\kappa).$$

$$\mathscr{P}[\![\,I{:}E\,]\!] =$$

$$\lambda\rho\kappa.\langle\mathscr{G}[\![\,E\,]\!]\rho\kappa\rangle\,\S\,\mathscr{P}[\![\,E\,]\!]\rho\kappa.$$

$$\mathscr{P}[\![\,I{::}E\,]\!] =$$

$$\lambda\rho\kappa.\mathscr{P}[\![\,E\,]\!]\rho\kappa.$$

$$\mathscr{P}[\![(E)]\!] =$$

$$\lambda\rho\kappa.\mathscr{P}[\![E]\!]\rho\kappa.$$

$$\mathscr{Q}[\![E_0 ; E_1]\!] =$$

$$\lambda\rho\kappa.\mathscr{Q}[\![E_0]\!]\rho(\lambda\varepsilon.\mathscr{G}[\![E_1]\!]\rho\kappa)\S\mathscr{Q}[\![E_1]\!]\rho\kappa.$$

$$\mathscr{Q}[\![\texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2]\!] =$$

$$\lambda\rho\kappa.\mathscr{Q}[\![E_1]\!]\rho\kappa\S\mathscr{Q}[\![E_2]\!]\rho\kappa.$$

$$\mathscr{Q}[\![\texttt{while } E_0 \texttt{ do } E_1]\!] =$$

$$\lambda\rho\kappa.\mathscr{Q}[\![E_1]\!]\rho(\lambda\varepsilon.\mathscr{G}[\![\texttt{while } E_0 \texttt{ do } E_1]\!]\rho\kappa).$$

$$\mathscr{Q}[\![I:L]\!] =$$

$$\lambda\rho\kappa.\mathscr{Q}[\![E]\!]\rho\kappa.$$

$$\mathscr{Q}[\![I::E]\!] =$$

$$\lambda\rho\kappa.\langle\mathscr{G}[\![E]\!]\rho\kappa\rangle\S\mathscr{Q}[\![E]\!]\rho\kappa.$$

$$\mathscr{Q}[\![(E)]\!] =$$

$$\lambda\rho\kappa.\mathscr{Q}[\![E]\!]\rho\kappa.$$

$$\mathscr{J}[\![E_0 ; E_1]\!] =$$

$$\mathscr{J}[\![E_0]\!]\S\mathscr{J}[\![E_1]\!].$$

$$\mathscr{J}[\![\texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2]\!] =$$

$$\mathscr{J}[\![E_1]\!]\S\mathscr{J}[\![E_2]\!].$$

$$\mathscr{J}[\![\texttt{while } E_0 \texttt{ do } E_1]\!] =$$

$$\mathscr{J}[\![E_1]\!].$$

$$\mathscr{J}[\![I:E]\!] =$$

$$\langle I\rangle\S\mathscr{J}[\![L]\!].$$

$$\mathscr{J}[\![I::E]\!] =$$

$$\mathscr{J}[\![E]\!].$$

$$\mathscr{J}[\![(E)]\!] =$$

$$\mathscr{J}[\![E]\!].$$

$$\mathscr{K}[\![E_0 ; E_1]\!] =$$

$$\mathscr{K}[\![E_0]\!]\S\mathscr{K}[\![E_1]\!].$$

$$\mathscr{E}[\![\text{if } E_0 \text{ then } E_1 \text{ else } E_2]\!] =$$
$$\quad \mathscr{E}[\![E_1]\!] \S \mathscr{E}[\![E_2]\!].$$

$$\mathscr{E}[\![\text{while } E_0 \text{ do } E_1]\!] =$$
$$\quad \mathscr{E}[\![E_1]\!].$$

$$\mathscr{E}[\![I{:}E]\!] =$$
$$\quad \mathscr{E}[\![E]\!].$$

$$\mathscr{E}[\![I{::}E]\!] =$$
$$\quad \langle I \rangle \S \mathscr{E}[\![E]\!].$$

$$\mathscr{E}[\![(E)]\!] =$$
$$\quad \mathscr{E}[\![E]\!].$$

$$\mathscr{S}[\![\Delta]\!] =$$
$$\lambda\rho\sigma.\mathit{fix}(\lambda\psi I^*.(\lambda\gamma.\#I^*=0 \to \langle\rangle, \langle\gamma\rangle \S \psi(I^* \dagger 1))$$
$$(\lambda\kappa'\sigma'.\mathscr{T}[\![\Delta]\!]\rho(\lambda\rho''\sigma''.\kappa'(\rho''[\![I^* \downarrow 1]\!] \downarrow 1 \,|\, E)\sigma')\sigma))$$
$$(\mathscr{H}[\![\Delta]\!]).$$

$$\mathscr{D}[\![I{=}E]\!] =$$
$$\lambda\rho\chi.\mathscr{L}[\![E]\!]\rho(\lambda\varepsilon.\chi(\mathit{arid}[\varepsilon/I])).$$

$$\mathscr{D}[\![I_1,\ldots,I_n{=}E]\!] =$$
$$\lambda\rho\chi.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon.\#\varepsilon \,|\, L^*=n \to \chi(\mathit{arid}[\varepsilon/\langle I_1,\ldots,I_n\rangle]), \tau).$$

$$\mathscr{D}[\![I{==}E]\!] =$$
$$\lambda\rho\chi.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon.\chi(\mathit{arid}[\varepsilon/I])).$$

$$\mathscr{D}[\![I_1,\ldots,I_n{==}E]\!] =$$
$$\lambda\rho\chi.\mathscr{R}[\![E]\!]\rho(\lambda\varepsilon\sigma.\#\varepsilon \,|\, L^*=n \to \chi(\mathit{arid}[\mathit{holds}\,\varepsilon\sigma/\langle I_1,\ldots,I_n\rangle])\sigma, \tau).$$

$$\mathscr{D}[\![\Delta_0 \text{ within } \Delta_1]\!] =$$
$$\lambda\rho\chi.\mathscr{D}[\![\Delta_0]\!]\rho(\lambda\rho'.\mathscr{D}[\![\Delta_1]\!](\mathit{divert}\rho\rho')\chi).$$

$$\mathscr{D}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$
$$\lambda\rho\chi.\mathit{run}\langle\mathscr{D}[\![\Delta_1]\!]\rho,\ldots,\mathscr{D}[\![\Delta_n]\!]\rho\rangle(\lambda\rho^*.\chi(\mathit{conserve}\rho^*)).$$

$$\mathscr{D}[\![\text{rec } \Delta]\!] =$$

$$\lambda\rho\chi\sigma.(\lambda\alpha^*.(\lambda\rho'.\alpha^*:\mathsf{L}^*\to\mathscr{T}[\![\Delta]\!]\rho'\chi(updates\alpha^*(dummy^*)\sigma),\tau)$$

$$(fix(\lambda\rho''.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]][\mathscr{S}[\![\Delta]\!]\rho''(updates\alpha^*(dummy^*)\sigma)/\mathscr{A}[\![\Delta]\!]]))$$

$$(news(\#\mathscr{I}[\![\Delta]\!])\sigma).$$

$$\mathscr{D}[\![(\Delta)]\!] =$$

$$\lambda\rho\chi.\mathscr{D}[\![\Delta]\!]\rho\chi.$$

$$\mathscr{T}[\![\mathrm{I}=\mathrm{E}]\!] =$$

$$\lambda\rho\chi.\mathscr{R}[\![\mathrm{E}]\!]\rho(\lambda\varepsilon\sigma.(\lambda\delta.\delta:\mathsf{L}\to\chi(arid[\delta/\mathrm{I}])(update\delta\varepsilon\sigma),\tau)(\rho[\![\mathrm{I}]\!]\downarrow 1)).$$

$$\mathscr{T}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}]\!] =$$

$$\lambda\rho\chi.(\lambda\kappa'.\mathscr{R}[\![\mathrm{E}]\!]\rho(\lambda\varepsilon\sigma.\#\varepsilon\,|\,\mathsf{L}^*=n\to\kappa'\varepsilon\sigma,\tau))$$

$$(\lambda\varepsilon\sigma.(\lambda\delta^*.\delta^*:\mathsf{L}^*\to\chi(arid[\delta^*/\langle\mathrm{I}_1,\ldots,\mathrm{I}_n\rangle])(updates\delta^*(holds\varepsilon\sigma)\sigma),\tau)$$

$$(\langle\rho[\![\mathrm{I}_1]\!]\downarrow 1,\ldots,\rho[\![\mathrm{I}_n]\!]\downarrow 1\rangle)).$$

$$\mathscr{T}[\![\mathrm{I}==\mathrm{E}]\!] =$$

$$\lambda\rho\chi.\mathscr{R}[\![\mathrm{E}]\!]\rho(\lambda\varepsilon.\chi(arid[\varepsilon/\mathrm{I}])).$$

$$\mathscr{T}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n==\mathrm{E}]\!] =$$

$$\lambda\rho\chi.\mathscr{R}[\![\mathrm{E}]\!]\rho(\lambda\varepsilon\sigma.\#\varepsilon\,|\,\mathsf{L}^*=n\to\chi(arid[holds\varepsilon\sigma/\langle\mathrm{I}_1,\ldots,\mathrm{I}_n\rangle])\sigma,\tau).$$

$$\mathscr{T}[\![\Delta_0 \text{ within } \Delta_1]\!] =$$

$$\lambda\rho\chi.\mathscr{D}[\![\Delta_0]\!]\rho(\lambda\rho'.\mathscr{T}[\![\Delta_1]\!](divert\rho\rho')\chi).$$

$$\mathscr{T}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$

$$\lambda\rho\chi.\mathscr{T}[\![\Delta_1]\!]\rho(\lambda\rho_1.\mathscr{T}[\![\Delta_2]\!](divert\rho\rho_1)$$

$$\ldots(\lambda\rho_n.\chi(conserve\langle\rho_1,\ldots,\rho_n\rangle))\ldots)).$$

$$\mathscr{T}[\![\text{rec } \Delta]\!] =$$

$$\lambda\rho\chi.\mathscr{T}[\![\Delta]\!]\rho\chi.$$

$$\mathscr{T}[\![(\Delta)]\!] =$$

$$\lambda\rho\chi.\mathscr{T}[\![\Delta]\!]\rho\chi.$$

$$\mathscr{I}[\![\mathrm{I}=\mathrm{E}]\!] =$$

$$\langle\mathrm{I}\rangle.$$

$$\mathscr{I}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}]\!] =$$

$$\langle\mathrm{I}_1,\ldots,\mathrm{I}_r\rangle.$$

$\mathscr{I}[\![I == E]\!] =$

$\langle \rangle .$

$\mathscr{I}[\![I_1, \ldots, I_n == E]\!] =$

$\langle \rangle .$

$\mathscr{I}[\![\Delta_0 \text{ within } \Delta_1]\!] =$

$\mathscr{I}[\![\Delta_1]\!] .$

$\mathscr{I}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$

$\mathscr{I}[\![\Delta_1]\!] \S \ldots \S \mathscr{I}[\![\Delta_n]\!] .$

$\mathscr{I}[\![\text{rec } \Delta]\!] =$

$\mathscr{I}[\![\Delta]\!] .$

$\mathscr{I}[\![(\Delta)]\!] =$

$\mathscr{I}[\![\Delta]\!] .$

$\mathscr{H}[\![I = E]\!] =$

$\langle \rangle .$

$\mathscr{H}[\![I_1, \ldots, I_n = E]\!] =$

$\langle \rangle .$

$\mathscr{H}[\![I == E]\!] =$

$\langle I \rangle .$

$\mathscr{H}[\![I_1, \ldots, I_n == E]\!] =$

$\langle I_1, \ldots, I_n \rangle .$

$\mathscr{H}[\![\Delta_0 \text{ within } \Delta_1]\!] =$

$\mathscr{H}[\![\Delta_1]\!] .$

$\mathscr{H}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$

$\mathscr{H}[\![\Delta_1]\!] \S \ldots \S \mathscr{H}[\![\Delta_n]\!] .$

$\mathscr{H}[\![\text{rec } \Delta]\!] =$

$\mathscr{H}[\![\Delta]\!] .$

$\mathscr{H}[\![(\Delta)]\!] =$

$\mathscr{H}[\![\Delta]\!] .$

APPENDIX TWO

STORE SEMANTICS

$\beta: V = B + L^* + J + F$	stored values
$\varepsilon: E = L + B + L^* + J + F$	expressed values
$\delta: D = L + B + L^* + J + F + G$	denoted values
$\omega: W = L + B + L^* + J + F + G + J + P$	witnessed values
$\beta: B = \{dummy\}^\circ + T + N + R + H^*$	basic values
$\varepsilon: T = \{true\}^\circ + \{false\}^\circ$	truth values
$\theta: J = Z^\circ \times U \times Y$	label closures
$\phi: F = O^\circ \times U$	function closures
$\gamma: G = O^\circ \times U \times S$	recursion closures
$\zeta: Z = U \to Y \to S \to A$	consecutions
$\xi: O = Z \to Z$	controls
$\pi: P = U \times Y \times S$	states
$\rho: U = [Ide \to D^\circ{}^*] \times J^* \times P^*$	environments
$\upsilon: Y = E^*$	stacks
$\sigma: S = [L \to [T \times V]] \times V^* \times V^*$	stores
$o: A$	answers
$\alpha: L$	locations
$\nu: N$	integers
R	reals
H	characters

$O: Mon$	monadic operators
$\Omega: Dya$	dyadic operators
$I: Ide$	identifiers
$B: Bas$	bases
$\Phi: Abs$	abstractions
$E: Exp$	expressions
$\Delta: Dec$	declarations

$\mathcal{O}$: Mon→E→B

$\mathcal{W}$: Dya→[E×E]→B

$\mathcal{B}$: Bas→B

$\mathcal{F}$: Abs→U→F

$\mathcal{C}$: Exp→O

$\mathcal{L}$: Exp→O

$\mathcal{R}$: Exp→O

$\mathcal{G}$: Exp→O

$\mathcal{P}$: Exp→Z→U→Y→J*

$\mathcal{Q}$: Exp→Z→U→Y→J*

$\mathcal{J}$: Exp→Ide*

$\mathcal{K}$: Exp→Ide*

$\mathcal{S}$: Dec→U→S→G*

$\mathcal{D}$: Dec→O

$\mathcal{T}$: Dec→O

$\mathcal{I}$: Dec→Ide*

$\mathcal{H}$: Dec→Ide*

Φ::=fn()E|fnI.E|fnI$_1$,...,I$_n$.E|fnI..E|fnI$_1$,...,I$_n$..E.

E::=I|B|Φ|OE|E$_0$$\Omega$E$_1$|E$_0$:=E$_1$|E$_1$,...,E$_n$:=E$_0$|get E|put E|E$_0$ aug E$_1$|
E$_1$,...,E$_n$|$E|E$|£E|E£|E$_0$E$_1$|val E|res E|goto E|Δ inside E|
E$_0$; E$_1$|if E$_0$ then E$_1$ else E$_2$|while E$_0$ do E$_1$|I:E|I::E|(E).

Δ::=I=E|I$_1$,...,I$_n$=E|I==E|I$_1$,...,I$_n$==E|Δ_0 within Δ_1|Δ_1 and...and Δ_n|
rec Δ|(Δ).

$$\mathscr{F}[\![\,\mathsf{fn}\,()E]\!] =$$

$$\lambda\rho.\langle\,\lambda\zeta.sv(\lambda\rho'\upsilon'\sigma'.\#\upsilon'\downarrow 1 \mid L^*=0 \to \mathscr{L}[\![E]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma',\tau), rend[\![\,\mathsf{fn}\,()E]\!]\rho\rangle\ .$$

$$\mathscr{F}[\![\,\mathsf{fn}I.E]\!] =$$

$$\lambda\rho.\langle\,\lambda\zeta\rho'\upsilon'\sigma'.\mathscr{L}[\![E]\!]\zeta\rho'[\upsilon'\downarrow 1/I](\upsilon'\dagger 1)\sigma', rend[\![\,\mathsf{fn}I.E]\!]\rho\rangle\ .$$

$$\mathscr{F}[\![\,\mathsf{fn}I_1,\ldots,I_n.E]\!] =$$

$$\lambda\rho.(\lambda\xi.\langle sv\xi, rend[\![\,\mathsf{fn}I_1,\ldots,I_n.E]\!]\rho\rangle)$$

$$(\lambda\zeta\rho'\upsilon'\sigma'.\#\upsilon'\downarrow 1 \mid L^*=n \to \mathscr{S}[\![E]\!]\zeta\rho'[\upsilon'\downarrow 1/\langle I_1,\ldots,I_n\rangle](\upsilon'\dagger 1)\sigma',\tau).$$

$$\mathscr{F}[\![\,\mathsf{fn}I..E]\!] =$$

$$\lambda\rho.\langle\,\lambda\zeta.sv(\lambda\rho'\upsilon'\sigma'.\mathscr{L}[\![E]\!]\zeta\rho'[\upsilon'\downarrow 1/I](\upsilon'\dagger 1)\sigma'), rend[\![\,\mathsf{fn}I..E]\!]\rho\rangle\ .$$

$$\mathscr{F}[\![\,\mathsf{fn}I_1,\ldots,I_n..E]\!] =$$

$$\lambda\rho.(\lambda\xi.\langle sv\xi, rend[\![\,\mathsf{fn}I_1,\ldots,I_n..E]\!]\rho\rangle)$$

$$(\lambda\zeta\rho'\upsilon'\sigma'.\#\upsilon'\downarrow 1 \mid L^*=n \to \mathscr{L}[\![E]\!]\zeta\rho'[holds(\upsilon'\downarrow 1)\sigma'/\langle I_1,\ldots,I_n\rangle](\upsilon'\dagger 1)\sigma',\tau)$$

$$\mathscr{E}[\![E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\alpha^*.(\lambda\rho'.(\lambda\sigma'.\alpha^*:L^* \to \mathscr{G}[\![E]\!](\lambda\rho''.\zeta(revert\rho\rho''))\rho'\upsilon\sigma',\tau)$$

$$(updates\alpha^*(\mathscr{F}[\![E]\!](\lambda\rho''.\zeta(revert\rho\rho''))\rho'\upsilon)\sigma'))$$

$$(fix(\lambda\rho'.\rho[\alpha^*/\mathscr{J}[\![E]\!]][\mathscr{Q}[\![E]\!](\lambda\rho''.\zeta(revert\rho\rho''))\rho'\upsilon/\mathscr{K}[\![E]\!]]))$$

$$(novels(\#\mathscr{J}[\![E]\!])\rho\upsilon\sigma).$$

$$\mathscr{L}[\![E]\!] =$$

$$\lambda\zeta.\mathscr{E}[\![E]\!](mv\zeta).$$

$$\mathscr{R}[\![E]\!] =$$

$$\lambda\zeta.\mathscr{E}[\![E]\!](sv\zeta).$$

$$\mathscr{G}[\![I]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\delta.\delta:G \to (\delta\downarrow 1)\zeta((\delta\downarrow 2)[\langle\rho,\upsilon,\sigma\rangle/\mathsf{rec}])()(\delta\downarrow 3),\zeta\rho(\langle\delta\rangle\,\S\upsilon)\sigma)(\rho[\![I]\!]\downarrow 1))$$

$$\mathscr{G}[\![B]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\zeta\rho(\langle\mathscr{B}[\![B]\!]\rangle\,\S\upsilon)\sigma.$$

$$\mathscr{G}[\![\Phi]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\zeta\rho(\langle\mathscr{F}[\![\Phi]\!]\rho\rangle\,\S\upsilon)\sigma.$$

$$\mathscr{G}[\![\, \mathrm{O}E \,]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![E]\!](\lambda\rho\upsilon\sigma.sv\zeta\rho(\langle \mathscr{O}[\![O]\!](\upsilon\downarrow 1)\rangle \S\upsilon\dagger 1)\sigma).$$

$$\mathscr{G}[\![\, E_0 \Omega E_1 \,]\!] =$$

$$\lambda\zeta.mete\langle \mathscr{R}[\![E_0]\!], \mathscr{R}[\![E_1]\!]\rangle(\lambda\rho\upsilon\sigma.sv\zeta\rho(\langle \mathscr{W}[\![\Omega]\!]\langle \upsilon\downarrow 2, \upsilon\downarrow 1\rangle\rangle \S\upsilon\dagger 2)\sigma).$$

$$\mathscr{G}[\![\, E_0 := E_1 \,]\!] =$$

$$\lambda\zeta.mete\langle \mathscr{L}[\![E_0]\!], \mathscr{R}[\![E_1]\!]\rangle(\lambda\rho\upsilon\sigma.\zeta\rho((dummy)\S\upsilon\dagger 2)(update(\upsilon\downarrow 2)(\upsilon\downarrow 1)\sigma)).$$

$$\mathscr{G}[\![\, E_1, \ldots, E_n := E_0 \,]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.mete\langle \mathscr{R}[\![E_0]\!], \mathscr{L}[\![E_1]\!], \ldots, \mathscr{L}[\![E_n]\!]\rangle(\lambda\rho\upsilon\sigma.\#\upsilon\downarrow(n+1) \mid L^* = n \rightarrow \zeta'\rho\upsilon\sigma, \tau))$$

$$(\lambda\rho\upsilon\sigma.\zeta\rho((dummy)\S\upsilon\dagger(n+1)(updates\langle \upsilon\downarrow n, \ldots, \upsilon\downarrow 1\rangle(holds(\upsilon\downarrow(n+1))\sigma)\sigma).$$

$$\mathscr{G}[\![\, \mathbf{get}\ E \,]\!] =$$

$$\lambda\zeta.\mathscr{L}[\![E]\!](\lambda\rho\upsilon\sigma.\#(\sigma\downarrow 2) > 0 \rightarrow \zeta\rho\upsilon(update(\upsilon\downarrow 1)(\sigma\downarrow 2\downarrow 1)(\sigma\downarrow 1, (\sigma\downarrow 2)\dagger 1, \sigma\downarrow 3)), \tau)$$

$$\mathscr{G}[\![\, \mathbf{put}\ E \,]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![E]\!](\lambda\rho\upsilon\sigma.\zeta\rho\upsilon\langle \sigma\downarrow 1, \sigma\downarrow 2, (\upsilon\downarrow 1) \S (\sigma\downarrow 3)\rangle).$$

$$\mathscr{G}[\![\, E_0\ \mathbf{aug}\ E_1 \,]\!] =$$

$$\lambda\zeta.mete\langle \mathscr{R}[\![E_0]\!], \mathscr{L}[\![E_1]\!]\rangle(\lambda\rho\upsilon\sigma.\upsilon\downarrow 2 : L^* \rightarrow \zeta\rho((\upsilon\downarrow 2)\S(\upsilon\downarrow 1))\S\upsilon\dagger 2)\sigma, \tau).$$

$$\mathscr{G}[\![\, E_1, \ldots, E_n \,]\!] =$$

$$\lambda\zeta.mete\langle \mathscr{L}[\![E_1]\!], \ldots, \mathscr{L}[\![E_n]\!]\rangle(\lambda\zeta\rho\upsilon\sigma.\zeta\rho(\langle \upsilon\downarrow n, \ldots, \upsilon\downarrow 1\rangle \mid L^*)\S\upsilon\dagger n, \sigma).$$

$$\mathscr{G}[\![\, \$E \,]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![E]\!](mv\zeta).$$

$$\mathscr{G}[\![\, E\$ \,]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![E]\!](\lambda\rho\upsilon\sigma.\upsilon\downarrow 1 : F \rightarrow \zeta\rho(\langle \lambda\zeta'.sv((\upsilon\downarrow 1\downarrow 1 \mid 0)\zeta'), \upsilon\downarrow 1\downarrow 2\rangle \S\upsilon\dagger 1)\sigma, \tau).$$

$$\mathscr{G}[\![\, £E \,]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.\mathscr{R}[\![E]\!]\zeta')$$

$$(\lambda\rho\upsilon\sigma.(\lambda\alpha^*.\alpha^* : L^* \rightarrow \zeta\rho(\langle \alpha^*\rangle \S\upsilon\dagger 1)(updates\alpha^*(holds(\upsilon\downarrow 1)\sigma)\sigma), \tau)$$

$$(novels(\#\upsilon\downarrow 1 \mid L^*)\rho\upsilon\sigma)).$$

$$\mathscr{G}[\![\, E£ \,]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.\mathscr{R}[\![E]\!]\zeta')$$

$$(\lambda\rho\upsilon\sigma.(\lambda\xi.\upsilon\downarrow 1 : F \rightarrow \zeta\rho(\langle \lambda\zeta''.sv(\xi\zeta''), \upsilon\downarrow 1\downarrow 2\rangle \S\upsilon\dagger 1)\sigma, \tau)$$

$$(\lambda\zeta''\rho''\upsilon''\sigma''.(\lambda\alpha^*.(\lambda\sigma'.\alpha^* : L^* \rightarrow (\upsilon\downarrow 1\downarrow 1 \mid 0)\zeta''\rho''(\langle \alpha^*\rangle \S\upsilon''\dagger 1)\sigma', \tau$$

$$(updates\alpha^*(holds(\upsilon''\downarrow 1)\sigma'')\sigma''))$$

$$(novels(\#\upsilon''\downarrow 1 \mid L^*)\rho''\upsilon''\sigma''))).$$

$$\mathscr{G}[\![E_0 E_1]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.(\lambda\zeta''.mete\langle\mathscr{R}[\![E_0]\!], \mathscr{G}[\![E_1]\!]\rangle(\lambda\rho\upsilon\sigma.\upsilon{\downarrow}2 : F \to \zeta'\rho\upsilon\sigma, sv\zeta''\rho\upsilon\sigma))$$

$$(\lambda\rho\upsilon\sigma.1 \le \upsilon{\downarrow}1 \mid N \le \#\upsilon{\downarrow}2 \mid L^* \to \zeta\rho(\langle\langle \upsilon{\downarrow}2{\downarrow}(\upsilon{\downarrow}1)\rangle \S \upsilon\dagger 2)\sigma, T))$$

$$(\lambda\rho\upsilon\sigma.(\upsilon{\downarrow}2{\downarrow}1 \mid 0)(\lambda\rho''.\zeta(revert\rho\rho''))(divert\rho(\upsilon{\downarrow}2{\downarrow}2))((\upsilon{\downarrow}1) \S \upsilon\dagger 2)\sigma)$$

$$\mathscr{G}[\![\texttt{val } E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{L}[\![E]\!](\lambda\rho''.\zeta(revert\rho\rho''))\rho[\langle \rho, \upsilon, \sigma\rangle / res]\upsilon\sigma.$$

$$\mathscr{G}[\![\texttt{res } E]\!] =$$

$$\lambda\zeta\,\mathscr{L}[\![E]\!](\lambda\rho\upsilon\sigma.(\rho[\![\texttt{res}]\!]{\downarrow}1{\downarrow}1 \mid Z)(\rho[\![\texttt{res}]\!]{\downarrow}1{\downarrow}2)(\rho[\![\texttt{res}]\!]{\downarrow}1{\downarrow}3)\sigma).$$

$$\mathscr{G}[\![\texttt{goto } E]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![E]\!](\lambda\rho\upsilon\sigma.(\upsilon{\downarrow}1{\downarrow}1 \mid Z)(\upsilon{\downarrow}1{\downarrow}2)(\upsilon{\downarrow}1{\downarrow}3)\sigma).$$

$$\mathscr{G}[\![\Delta \texttt{ inside } E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{D}[\![\Delta]\!](\mathscr{L}[\![E]\!](\lambda\rho'.\zeta(revert\rho\rho')))\rho\upsilon\sigma.$$

$$\mathscr{G}[\![E_0 ; E_1]\!] =$$

$$\lambda\zeta.\mathscr{G}[\![E_0]\!](\lambda\rho\upsilon\sigma.\mathscr{G}[\![E_1]\!]\zeta\rho(\upsilon\dagger 1)\sigma).$$

$$\mathscr{G}[\![\texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![E_0]\!](\lambda\rho\upsilon\sigma.\upsilon{\downarrow}1 \mid T \to \mathscr{G}[\![E_1]\!]\zeta\rho(\upsilon\dagger 1)\sigma, \mathscr{G}[\![E_2]\!]\zeta\rho(\upsilon\dagger 1)\sigma).$$

$$\mathscr{G}[\![\texttt{while } E_0 \texttt{ do } E_1]\!] =$$

$$\lambda\zeta.fix(\lambda\zeta'.(\lambda\zeta''.\mathscr{R}[\![E_0]\!](\lambda\rho\upsilon\sigma.\upsilon{\downarrow}1 \mid T \to \zeta''\rho(\upsilon\dagger 1)\sigma, \zeta\rho(\langle\langle dummy\rangle \S \upsilon\dagger 1)\sigma)$$

$$(\mathscr{G}[\![E_1]\!](\lambda\rho'\upsilon'\sigma'.\zeta'\rho'(\upsilon'\dagger 1)\sigma'))).$$

$$\mathscr{G}[\![I : E]\!] =$$

$$\lambda\zeta.\mathscr{G}[\![E]\!]\zeta.$$

$$\mathscr{G}[\![I :: E]\!] =$$

$$\lambda\zeta.\mathscr{G}[\![E]\!]\zeta.$$

$$\mathscr{G}[\![(E)]\!] =$$

$$\lambda\zeta.\mathscr{G}[\![E]\!]\zeta.$$

$$\mathscr{P}[\![E_0 ; E_1]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{P}[\![E_0]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon \S \mathscr{P}[\![E_1]\!]\zeta\rho\upsilon.$$

$$\mathscr{P}[\![\texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{P}[\![E_1]\!]\zeta\rho\upsilon \S \mathscr{P}[\![E_2]\!]\zeta\rho\upsilon.$$

$$\mathscr{P}[\![\text{while } E_0 \text{ do } E_1]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{P}[\![E_1]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{S}[\![\text{while } E_0 \text{ do } E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon.$$

$$\mathscr{P}[\![I:E]\!] =$$

$$\lambda\zeta\rho\upsilon.\langle\langle\mathscr{S}[\![E]\!]\zeta,\rho,\upsilon\rangle\rangle\S\mathscr{P}[\![E]\!]\zeta\rho\upsilon.$$

$$\mathscr{P}[\![I::E]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{P}[\![E]\!]\zeta\rho\upsilon.$$

$$\mathscr{P}[\![(E)]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{P}[\![E]\!]\zeta\rho\upsilon.$$

$$\mathscr{Q}[\![E_0;\ E_1]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{Q}[\![E_0]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{S}[\![E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon\S\mathscr{Q}[\![E_1]\!]\zeta\rho\upsilon.$$

$$\mathscr{Q}[\![\text{if } E_0 \text{ then } E_1 \text{ else } E_2]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{Q}[\![E_1]\!]\zeta\rho\upsilon\S\mathscr{Q}[\![E_2]\!]\zeta\rho\upsilon.$$

$$\mathscr{Q}[\![\text{while } E_0 \text{ do } E_1]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{Q}[\![E_1]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{S}[\![\text{while } E_0 \text{ do } E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon.$$

$$\mathscr{Q}[\![I:E]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{Q}[\![E]\!]\zeta\rho\upsilon.$$

$$\mathscr{Q}[\![I::E]\!] =$$

$$\lambda\zeta\rho\upsilon.\langle\langle\mathscr{S}[\![E]\!]\zeta,\rho,\upsilon\rangle\rangle\S\mathscr{Q}[\![E]\!]\zeta\rho\upsilon.$$

$$\mathscr{Q}[\![(E)]\!] =$$

$$\lambda\zeta\rho\upsilon.\mathscr{Q}[\![E]\!]\zeta\rho\upsilon.$$

$$\mathscr{J}[\![E_0;\ E_1]\!] =$$

$$\mathscr{J}[\![E_0]\!]\S\mathscr{J}[\![E_1]\!].$$

$$\mathscr{J}[\![\text{if } E_0 \text{ then } E_1 \text{ else } E_2]\!] =$$

$$\mathscr{J}[\![E_1]\!]\S\mathscr{J}[\![E_2]\!].$$

$$\mathscr{J}[\![\text{while } E_0 \text{ do } E_1]\!] =$$

$$\mathscr{J}[\![E_1]\!].$$

$$\mathscr{J}[\![I:E]\!] =$$

$$\langle I\rangle\S\mathscr{J}[\![E]\!].$$

$$\mathscr{J}[\![\mathrm{I}::\mathrm{E}]\!]=$$

$$\mathscr{J}[\![\mathrm{E}]\!].$$

$$\mathscr{J}[\![(\mathrm{E})]\!]=$$

$$\mathscr{J}[\![\mathrm{E}]\!].$$

$$\mathscr{K}[\![\mathrm{E}_0;\ \mathrm{E}_1]\!]=$$

$$\mathscr{K}[\![\mathrm{E}_0]\!]\,\S\,\mathscr{K}[\![\mathrm{E}_1]\!].$$

$$\mathscr{K}[\![\textbf{if } \mathrm{E}_0 \textbf{ then } \mathrm{E}_1 \textbf{ else } \mathrm{E}_2]\!]=$$

$$\mathscr{K}[\![\mathrm{E}_1]\!]\,\S\,\mathscr{K}[\![\mathrm{E}_2]\!].$$

$$\mathscr{K}[\![\textbf{while } \mathrm{E}_0 \textbf{ do } \mathrm{E}_1]\!]=$$

$$\mathscr{K}[\![\mathrm{E}_1]\!].$$

$$\mathscr{K}[\![\mathrm{I}:\mathrm{E}]\!]=$$

$$\mathscr{K}[\![\mathrm{E}]\!].$$

$$\mathscr{K}[\![\mathrm{I}::\mathrm{E}]\!]=$$

$$\langle \mathrm{I}\rangle\,\S\,\mathscr{K}[\![\mathrm{E}]\!].$$

$$\mathscr{K}[\![(\mathrm{E})]\!]=$$

$$\mathscr{K}[\![\mathrm{E}]\!].$$

$$\mathscr{S}[\![\Delta]\!]=$$

$$\lambda\rho\sigma.\mathit{fix}(\lambda\psi\mathrm{I}^*.(\lambda\xi.\#\mathrm{I}^*=0\to\langle\rangle,\langle\langle\xi,\mathit{tear}[\![\Delta]\!]\rho,\sigma\rangle\rangle\,\S\,\psi(\mathrm{I}^*\dagger 1))$$

$$(\lambda\zeta.\mathscr{T}[\![\Delta]\!](\lambda\rho'\upsilon'\sigma'.\mathit{recur}\zeta\rho'\langle\rho'[\![\mathrm{I}^*\downarrow 1]\!]\downarrow 1|\mathrm{E}\rangle\sigma'))$$

$$(\mathscr{H}[\![\Delta]\!]).$$

$$\mathscr{D}[\![\mathrm{I}=\mathrm{E}]\!]=$$

$$\lambda\zeta.\mathscr{L}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\zeta\rho[\upsilon\downarrow 1/\mathrm{I}](\upsilon\dagger 1)\sigma).$$

$$\mathscr{D}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}]\!]=$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1\,|\,\mathrm{L}^*=n\to\zeta\rho[\upsilon\downarrow 1/\langle\mathrm{I}_1,\ldots,\mathrm{I}_n\rangle](\upsilon\dagger 1)\sigma,\tau).$$

$$\mathscr{D}[\![\mathrm{I}==\mathrm{E}]\!]=$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\zeta\rho[\upsilon\downarrow 1/\mathrm{I}](\upsilon\dagger 1)\sigma).$$

$$\mathscr{D}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n==\mathrm{E}]\!]=$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1\,|\,\mathrm{L}^*=n\to\zeta\rho[\mathit{holds}(\upsilon\downarrow 1)\sigma/\langle\mathrm{I}_1,\ldots,\mathrm{I}_n\rangle](\upsilon\dagger 1)\sigma,\tau).$$

$$\mathscr{D}[\![\Delta_0 \text{ within } \Delta_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{D}[\![\Delta_0]\!](\mathscr{D}[\![\Delta_1]\!](\lambda\rho'.\zeta(trim[\![\Delta_1]\!]\rho\rho')))\rho\upsilon\sigma.$$

$$\mathscr{D}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$

$$\lambda\zeta.(\lambda\xi^*.deal\xi^*(\lambda\rho^*\rho.\zeta(pick[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!]\rho^*\rho)))$$

$$(fix(\lambda\psi m.m{\leq}n{\rightarrow}\langle\lambda\zeta\rho'.\mathscr{D}[\![\Delta_m]\!](\lambda\rho''.\zeta(clip[\![\Delta_m]\!]\rho'\rho''))\rho'\rangle \S\psi(m{+}1),\langle\rangle)1)$$

$$\mathscr{D}[\![\text{rec } \Delta]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\alpha^*.(\lambda\rho'.\alpha^*:\mathsf{L}^*{\rightarrow}\mathscr{I}[\![\Delta]\!]\zeta\rho'\upsilon(updates\alpha^*(dummy^*)\sigma),\tau)$$

$$(fix(\lambda\rho''.\rho[\alpha^*/\mathscr{I}[\![\Delta]\!]][\mathscr{I}[\![\Delta]\!]\rho''(updates\alpha^*(dummy^*)\sigma)/\mathscr{K}[\![\Delta]\!]])))$$

$$(novels(\#\mathscr{I}[\![\Delta]\!])\rho\upsilon\sigma).$$

$$\mathscr{D}[\![(\Delta)]\!] =$$

$$\lambda\zeta.\mathscr{D}[\![\Delta]\!]\zeta.$$

$$\mathscr{I}[\![\mathrm{I}{=}\mathrm{E}]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.(\lambda\delta.\delta:\mathsf{L}{\rightarrow}\zeta\rho(\upsilon\dagger 1)(update\delta(\upsilon\downarrow 1)\sigma),\tau)(\rho[\![\mathrm{I}]\!]\downarrow 1)).$$

$$\mathscr{I}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n{=}\mathrm{E}]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1 \mid \mathsf{L}^*{=}n{\rightarrow}\zeta'\rho\upsilon\sigma,\tau))$$

$$(\lambda\rho\upsilon\sigma.(\lambda\delta^*.\delta^*:\mathsf{L}^*{\rightarrow}\zeta\rho(\upsilon\dagger 1)(updates\delta^*(holds(\upsilon\downarrow 1)\sigma)\sigma),\tau)$$

$$(\langle\rho[\![\mathrm{I}_1]\!]\downarrow 1,\ldots,\rho[\![\mathrm{I}_n]\!]\downarrow 1\rangle)).$$

$$\mathscr{I}[\![\mathrm{I}{=}{=}\mathrm{E}]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\zeta(invert\rho(arid[\upsilon\downarrow 1/\mathrm{I}]))(\upsilon\dagger 1)\sigma).$$

$$\mathscr{I}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n{=}{=}\mathrm{E}]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1 \mid \mathsf{L}^*{=}n{\rightarrow}\zeta'\rho\upsilon\sigma,\tau))$$

$$(\lambda\rho\upsilon\sigma.\zeta(invert\rho(arid[holds(\upsilon\downarrow 1)\sigma/\langle\mathrm{I}_1,\ldots,\mathrm{I}_n\rangle]))(\upsilon\dagger 1)\sigma).$$

$$\mathscr{I}[\![\Delta_0 \text{ within } \Delta_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{D}[\![\Delta_0]\!](\mathscr{I}[\![\Delta_1]\!](\lambda\rho'.\zeta(trim[\![\Delta_1]\!]\rho\rho')))\rho\upsilon\sigma.$$

$$\mathscr{I}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$

$$\lambda\zeta.(\lambda\xi^*.deal\xi^*(\lambda\rho^*\rho.\zeta(pick[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!]\rho^*\rho)))$$

$$(fix(\lambda\psi m.m{\leq}n{\rightarrow}\langle\lambda\zeta\rho'.\mathscr{I}[\![\Delta_m]\!](\lambda\rho''.\zeta(clip[\![\Delta_m]\!]\rho'\rho''))\rho'\rangle \S\psi(m{+}1),\langle\rangle)1)$$

$$\mathscr{I}[\![\text{rec } \Delta]\!] =$$

$$\lambda\zeta.\mathscr{I}[\![\Delta]\!]\zeta.$$

$\mathscr{T}[\![(\Delta)]\!] =$

$\lambda\zeta.\mathscr{T}[\![\Delta]\!]\zeta .$

$\mathscr{D}[\![\mathrm{I}=\mathrm{E}]\!] =$

$\langle \mathrm{I} \rangle .$

$\mathscr{D}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}]\!] =$

$\langle \mathrm{I}_1,\ldots,\mathrm{I}_n \rangle .$

$\mathscr{D}[\![\mathrm{I}==\mathrm{E}]\!] =$

$\langle\rangle .$

$\mathscr{D}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n==\mathrm{E}]\!] =$

$\langle\rangle .$

$\mathscr{D}[\![\Delta_0 \text{ within } \Delta_1]\!] =$

$\mathscr{D}[\![\Delta_1]\!] .$

$\mathscr{D}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$

$\mathscr{D}[\![\Delta_1]\!] \S \ldots \S \mathscr{D}[\![\Delta_n]\!] .$

$\mathscr{D}[\![\text{rec } \Delta]\!] =$

$\mathscr{D}[\![\Delta]\!] .$

$\mathscr{D}[\![(\Delta)]\!] =$

$\mathscr{D}[\![\Delta]\!] .$

$\mathscr{H}[\![\mathrm{I}=\mathrm{E}]\!] =$

$\langle\rangle .$

$\mathscr{H}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}]\!] =$

$\langle\rangle .$

$\mathscr{H}[\![\mathrm{I}==\mathrm{E}]\!] =$

$\langle \mathrm{I} \rangle .$

$\mathscr{H}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n==\mathrm{E}]\!] =$

$\langle \mathrm{I}_1,\ldots,\mathrm{I}_n \rangle .$

$\mathscr{H}[\![\Delta_0 \text{ within } \Delta_1]\!] =$

$\mathscr{H}[\![\Delta_1]\!] .$

$$\mathcal{H}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$
$$\mathcal{H}[\![\Delta_1]\!] \S \ldots \S \mathcal{H}[\![\Delta_n]\!] .$$

$$\mathcal{H}[\![\text{rec } \Delta]\!] =$$
$$\mathcal{H}[\![\Delta]\!] .$$

$$\mathcal{H}[\![(\Delta)]\!] =$$
$$\mathcal{H}[\![\Delta]\!] .$$

APPENDIX THREE

STACK SEMANTICS

$\beta : V = B + L^* + J + F$	stored values
$\varepsilon : E = L + B + L^* + J + F$	expressed values
$\delta : D = L + B + L^* + J + F$	denoted values
$\omega : W = L + B + L^* + J + F + J + P$	witnessed values
$\beta : B = \{dummy\}^\circ + T + N + R + H^*$	basic values
$\varepsilon : T = \{true\}^\circ + \{false\}^\circ$	truth values
$\theta : J = Z^\circ$	label closures
$\phi : F = O^\circ$	function closures
$\zeta : Z = U \to Y \to S \to A$	consecutions
$\xi : O = Z \to Z$	controls
$\pi : P = U \times Y \times S$	states
$\rho : U = [Ide \leftrightarrow [D \times N \times N]^*] \times [J \times N \times N]^* \times [P \times N \times N]^*$	environments
$\upsilon : Y = E^*$	stacks
$\sigma : S = [L \leftrightarrow [T \times V]] \times V^* \times V^*$	stores
$o : A$	answers
$\alpha : L$	locations
$\nu : N$	integers
R	reals
H	characters

$O : Mon$	monadic operators
$\Omega : Dya$	dyadic operators
$I : Ide$	identifiers
$B : Bas$	bases
$\Phi : Abs$	abstractions
$E : Exp$	expressions
$\Delta : Dec$	declarations

$\mathcal{O}$: Mon→E→B

$\mathcal{W}$: Dya→[E×E]→B

$\mathcal{B}$: Bas→B

$\mathcal{F}$: Abs→U→F

$\mathcal{E}$: Exp→0

$\mathcal{L}$: Exp→0

$\mathcal{R}$: Exp→0

$\mathcal{G}$: Exp→0

$\mathcal{P}$: Exp→Z→U→Y→S→J*

$\mathcal{Q}$: Exp→Z→U→Y→S→J*

$\mathcal{J}$: Exp→Ide*

$\mathcal{K}$: Exp→Ide*

$\mathcal{S}$: Dec→{⟨⟩}°

$\mathcal{D}$: Dec→0

$\mathcal{T}$: Dec→0

$\mathcal{I}$: Dec→Ide*

$\mathcal{H}$: Dec→Ide*

Φ::=fn()E|fnI.E|fn$I_1,\ldots,I_n$.E|fnI..E|fn$I_1,\ldots,I_n$..E.

E::=I|B|Φ|OE|$E_0 \Omega E_1$|E_0:=E_1|$E_1,\ldots,E_n$:=E_0|get E|put E|E_0 aug E_1|
$E_1,\ldots,E_n$|$E|E$|£E|E£|E_0E_1|val E|res E|goto E|Δ inside E|
E_0; E_1|if E_0 then E_1 else E_2|while E_0 do E_1|I:E|I::E|(E).

Λ::=I=E|$I_1,\ldots,I_n$=E|I==E|$I_1,\ldots,I_n$==E|Δ_0 within Δ_1|Δ_1 and...and Δ_n|
rec Δ|(Δ).

$$\mathscr{F}[\![\ \mathsf{fn}()\mathsf{E}]\!] =$$

$$\lambda\rho.\lambda\zeta.sv(\lambda\rho'\upsilon'\sigma'.(\lambda\rho''.\#\upsilon'\downarrow 1 \mid \mathsf{L}^*=0 \to \mathscr{L}[\![\ \mathsf{E}]\!](remit\zeta)\rho''(\upsilon'\dagger 1)\sigma',\tau)$$
$$(divert\rho'(rend[\![\ \mathsf{fn}()\ \mathsf{E}]\!]\rho)[\langle\rho',\upsilon'\dagger 1,\sigma'\rangle//\mathsf{rec}])).$$

$$\mathscr{F}[\![\ \mathsf{fnI}.\mathsf{E}]\!] =$$

$$\lambda\rho.\lambda\zeta\rho'\upsilon'\sigma'.(\lambda\rho''.\mathscr{L}[\![\ \mathsf{E}]\!](remit\zeta)\rho''[\upsilon'\downarrow 1//\mathsf{I}](\upsilon'\dagger 1)\sigma')$$
$$(divert\rho'(rend[\![\ \mathsf{fnI}.\mathsf{E}]\!]\rho)[\langle\rho',\upsilon'\dagger 1,\sigma'\rangle//\mathsf{rec}]).$$

$$\mathscr{F}[\![\ \mathsf{fnI}_1,\ldots,\mathsf{I}_n.\mathsf{E}]\!] =$$

$$\lambda\rho.\lambda\zeta.sv(\lambda\rho'\upsilon'\sigma'.(\lambda\rho''.\#\upsilon'\downarrow 1 \mid \mathsf{L}^*=n \to \mathscr{L}[\![\ \mathsf{E}]\!](remit\zeta)\rho''(\upsilon'\dagger 1)\sigma',\tau)$$
$$((\lambda\rho'''.\rho'''[\langle\rho',\upsilon'\dagger 1,\sigma'\rangle//\mathsf{rec}][\upsilon'\downarrow 1//\langle\mathsf{I}_1,\ldots,\mathsf{I}_n\rangle])$$
$$(divert\rho'(rend[\![\ \mathsf{fnI}_1,\ldots,\mathsf{I}_n.\mathsf{E}]\!]\rho)))).$$

$$\mathscr{F}[\![\ \mathsf{fnI}..\mathsf{E}]\!] =$$

$$\lambda\rho.\lambda\zeta.sv(\lambda\rho'\upsilon'\sigma'.(\lambda\rho''.\mathscr{L}[\![\ \mathsf{E}]\!](remit\zeta)\rho''[\upsilon'\downarrow 1//\mathsf{I}](\upsilon'\dagger 1)\sigma',\tau)$$
$$(divert\rho'(rend[\![\ \mathsf{fnI}..\mathsf{E}]\!]\rho)[\langle\rho',\upsilon'\dagger 1,\sigma'\rangle//\mathsf{rec}])).$$

$$\mathscr{F}[\![\ \mathsf{fnI}_1,\ldots,\mathsf{I}_n..\mathsf{E}]\!] =$$

$$\lambda\rho.\lambda\zeta.sv(\lambda\rho'\upsilon'\sigma'.(\lambda\rho''.\#\upsilon'\downarrow 1 \mid \mathsf{L}^*=n \to \mathscr{L}[\![\ \mathsf{E}]\!](remit\zeta)\rho''(\upsilon'\dagger 1)\sigma',\tau)$$
$$((\lambda\rho'''.\rho'''[\langle\rho',\upsilon'\dagger 1,\sigma'\rangle//\mathsf{rec}][holds(\upsilon'\downarrow 1)\sigma'//\langle\mathsf{I}_1,\ldots,\mathsf{I}_n\rangle])$$
$$(divert\rho'(rend[\![\ \mathsf{fnI}_1,\ldots,\mathsf{I}_n..\mathsf{E}]\!]\rho)))).$$

$$\mathscr{E}[\![\ \mathsf{E}]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\alpha^*.(\lambda\rho'.(\lambda\sigma'.(\lambda\rho''.(\lambda\sigma''.\alpha^*:\mathsf{L}^* \to \mathscr{G}[\![\ \mathsf{E}]\!](remit\zeta)\rho''\upsilon\sigma'',\tau)$$
$$(updates\alpha^*(\mathscr{P}[\![\ \mathsf{E}]\!](remit\zeta)\rho''\upsilon\sigma')\sigma))$$
$$(\rho'[\mathscr{Q}[\![\ \mathsf{E}]\!](remit\zeta)\rho'[dummy^*//\mathscr{K}[\![\ \mathsf{E}]\!]]\upsilon\sigma'//\mathscr{K}[\![\ \mathsf{E}]\!]])$$
$$(updates\alpha^*(dummy^*)\sigma))$$
$$(\rho[\langle\rho,\upsilon,\sigma\rangle//\mathsf{rec}][\alpha^*//\mathscr{J}[\![\ \mathsf{E}]\!]]))$$
$$(news(\#\mathscr{J}[\![\ \mathsf{E}]\!])\sigma).$$

$$\mathscr{L}[\![\ \mathsf{E}]\!] =$$

$$\lambda\zeta.\mathscr{E}[\![\ \mathsf{E}]\!](mv\zeta).$$

$$\mathscr{R}[\![\ \mathsf{E}]\!] =$$

$$\lambda\zeta.\mathscr{E}[\![\ \mathsf{E}]\!](sv\zeta).$$

$\mathscr{G}[\![\mathrm{I}]\!] =$

$\lambda\zeta\rho\upsilon\sigma.\zeta\rho(\langle\, ravel\rho[\![\mathrm{I}]\!]\downarrow 1\downarrow 1\rangle\, \S\upsilon)\sigma.$

$\mathscr{G}[\![\mathrm{B}]\!] =$

$\lambda\zeta\rho\upsilon\sigma.\zeta\rho(\langle\, \mathscr{B}[\![\mathrm{B}]\!]\rangle\, \S\upsilon)\sigma.$

$\mathscr{G}[\![\Phi]\!] =$

$\lambda\zeta\rho\upsilon\sigma.\zeta\rho(\langle\, \mathscr{F}[\![\Phi]\!]\rho\rangle\, \S\upsilon)\sigma.$

$\mathscr{G}[\![\mathrm{OE}]\!] =$

$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!]\,(\lambda\rho\upsilon\sigma.sv\zeta\rho(\langle\, \mathscr{O}[\![\mathrm{O}]\!]\,(\upsilon\downarrow 1)\rangle\, \S\upsilon\dagger 1)\sigma).$

$\mathscr{G}[\![\mathrm{E}_0 \Omega \mathrm{E}_1]\!] =$

$\lambda\zeta.mete\langle \mathscr{R}[\![\mathrm{E}_0]\!], \mathscr{R}[\![\mathrm{E}_1]\!]\rangle\,(\lambda\rho\upsilon\sigma.sv\zeta\rho(\langle \mathscr{W}[\![\Omega]\!]\langle \upsilon\downarrow 2, \upsilon\downarrow 1\rangle\rangle\, \S\upsilon\dagger 2)\sigma)).$

$\mathscr{G}[\![\mathrm{E}_0 := \mathrm{E}_1]\!] =$

$\lambda\zeta.mete\langle \mathscr{L}[\![\mathrm{E}_0]\!], \mathscr{R}[\![\mathrm{E}_1]\!]\rangle\,(\lambda\rho\upsilon\sigma.\zeta\rho(\langle\, dummy\rangle\, \S\upsilon\dagger 2)\,(update\,(\upsilon\downarrow 2)\,(\upsilon\downarrow 1)\sigma).$

$\mathscr{G}[\![\mathrm{E}_1, \ldots, \mathrm{E}_n := \mathrm{E}_0]\!] =$

$\lambda\zeta.(\lambda\zeta'.mete\langle \mathscr{R}[\![\mathrm{E}_0]\!], \mathscr{L}[\![\mathrm{E}_1]\!], \ldots, \mathscr{L}[\![\mathrm{E}_n]\!]\rangle\,(\lambda\rho\upsilon\sigma.\#\upsilon\downarrow(n+1)\,|\,\mathrm{L}^* = n \rightarrow \zeta'\rho\upsilon\sigma, \mathrm{T}))$

$(\lambda\rho\upsilon\sigma.\zeta\rho(\langle\, dummy\rangle \S\upsilon\dagger(n+1)(updates\langle \upsilon\downarrow n, \ldots, \upsilon\downarrow 1\rangle(holds\,(\upsilon\downarrow(n+1))\sigma)\sigma).$

$\mathscr{G}[\![\mathrm{get\ E}]\!] =$

$\lambda\zeta.\mathscr{L}[\![\mathrm{E}]\!]\,(\lambda\rho\upsilon\sigma.\#(\sigma\downarrow 2) > 0 \rightarrow \zeta\rho\upsilon(update\,(\upsilon\downarrow 1)(\sigma\downarrow 2\downarrow 1)\langle \sigma\downarrow 1, (\sigma\downarrow 2)\dagger 1, \sigma\downarrow 3\rangle), \mathrm{T}).$

$\mathscr{G}[\![\mathrm{put\ E}]\!] =$

$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!]\,(\lambda\rho\upsilon\sigma.\zeta\rho\upsilon\langle \sigma\downarrow 1, \sigma\downarrow 2, \langle \upsilon\downarrow 1\rangle \S \langle \sigma\downarrow 3\rangle\rangle).$

$\mathscr{G}[\![\mathrm{E}_0\ \mathrm{aug}\ \mathrm{E}_1]\!] =$

$\lambda\zeta.mete\langle \mathscr{R}[\![\mathrm{E}_0]\!], \mathscr{L}[\![\mathrm{E}_1]\!]\rangle\,(\lambda\rho\upsilon\sigma.\upsilon\downarrow 2 : \mathrm{L}^* \rightarrow \zeta\rho(\langle\, (\upsilon\downarrow 2) \S \langle \upsilon\downarrow 1\rangle\rangle\, \S\upsilon\dagger 1)\sigma, \mathrm{T}).$

$\mathscr{G}[\![\mathrm{E}_1, \ldots, \mathrm{E}_n]\!] =$

$\lambda\zeta.mete\langle \mathscr{L}[\![\mathrm{E}_1]\!], \ldots, \mathscr{L}[\![\mathrm{E}_n]\!]\rangle\,(\lambda\rho\upsilon\sigma.\zeta\rho(\langle\langle \upsilon\downarrow n, \ldots, \upsilon\downarrow 1\rangle\,|\,\mathrm{L}^*\rangle\, \S\upsilon\dagger n)\sigma).$

$\mathscr{G}[\![\$\mathrm{E}]\!] =$

$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!]\,(mv\zeta).$

$\mathscr{G}[\![\mathrm{E}\$]\!] =$

$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!]\,(\lambda\rho\upsilon\sigma.\upsilon\downarrow 1 : \mathrm{F} \rightarrow \zeta\rho(\langle\, \lambda\zeta'.sv\,((\upsilon\downarrow 1\,|\,0)\zeta')\rangle\, \S\upsilon\dagger 1)\sigma, \mathrm{T}).$

$\mathscr{G}[\![\pounds\mathrm{E}]\!] =$

$\lambda\zeta.\,(\lambda\zeta'.\mathscr{R}[\![\mathrm{E}]\!]\zeta')$

$(\lambda\rho\upsilon\sigma.(\lambda\alpha^*.\alpha^* : \mathrm{L}^* \rightarrow \zeta\rho(\langle \alpha^*\rangle\, \S\upsilon\dagger 1)\,(updates\alpha^*(holds(\upsilon\downarrow 1)\sigma)\sigma), \mathrm{T})$

$(news\,(\#\upsilon\downarrow 1\,|\,\mathrm{L}^*)\sigma)).$

$$\mathscr{G}[\![\, E\pounds]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.\mathscr{R}[\![\, E]\!]\zeta')$$

$$(\lambda\rho\upsilon\sigma.(\lambda\xi.\upsilon\downarrow 1 : F\to\zeta\rho(\langle\lambda\zeta''.sv(\xi\zeta'')\rangle\,\S\upsilon\dagger 1)\sigma, T)$$

$$(\lambda\zeta''\rho''\upsilon''\sigma''.(\lambda\alpha^*.(\lambda\sigma'.\alpha^* : L^*\to(\upsilon\downarrow 1\,|\,0)\zeta''\rho''(\langle\alpha^*\rangle\,\S\upsilon''\dagger 1)\sigma', T)$$

$$(updates\alpha^*(holds(\upsilon''\downarrow 1)\sigma'')\sigma''))$$

$$(news(\#\upsilon''\downarrow 1\,|\,L^*)\sigma''))).$$

$$\mathscr{G}[\![\, E_0E_1]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.(\lambda\zeta''.mete(\mathscr{R}[\![\, E_0]\!], \mathscr{G}[\![\, E_1]\!])(\lambda\rho\upsilon\sigma.\upsilon\downarrow 2 : F\to\zeta'\rho\upsilon\sigma, sv\zeta''\rho\upsilon\sigma))$$

$$(\lambda\rho\upsilon\sigma.1\le\upsilon\downarrow 1\,|\,N\le\#\upsilon\downarrow 2\,|\,L^*\to\zeta\rho(\langle\upsilon\downarrow 2\downarrow(\upsilon\downarrow 1)\rangle\,\S\upsilon\dagger 2)\sigma, T))$$

$$(\lambda\rho\upsilon\sigma.(\upsilon\downarrow 2\,|\,0)\zeta\rho(\langle\upsilon\downarrow 1\rangle\,\S\upsilon\dagger 2)\sigma).$$

$$\mathscr{G}[\![\, \mathbf{val}\ E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\zeta'.\mathscr{L}[\![\, E]\!](remit\zeta)\rho[\langle\rho,\upsilon,\sigma\rangle\ //\mathbf{rec}][\zeta'//\mathbf{res}]\upsilon\sigma)$$

$$(\lambda\rho'\upsilon'\sigma'.remit\zeta\rho'[\langle\rho,\upsilon,\sigma\rangle\ //\mathbf{rec}]\upsilon'\sigma').$$

$$\mathscr{G}[\![\, \mathbf{res}\ E]\!] =$$

$$\lambda\zeta.\mathscr{L}[\![\, E]\!](\lambda\rho\upsilon\sigma.(ravel\rho[\![\, \mathbf{res}]\!]\downarrow 1\downarrow 1\,|\,Z)\rho\upsilon\sigma).$$

$$\mathscr{G}[\![\, \mathbf{goto}\ E]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\, E]\!](\lambda\rho\upsilon\sigma.(\upsilon\downarrow 1\,|\,Z)\rho(\upsilon\dagger 1)\sigma).$$

$$\mathscr{G}[\![\, \Delta\ \mathbf{inside}\ E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{D}[\![\, \Delta]\!](\mathscr{L}[\![\, E]\!](remit\zeta))\rho[\langle\rho,\upsilon,\sigma\rangle\ //\mathbf{rec}]\upsilon\sigma.$$

$$\mathscr{G}[\![\, E_0\ ;\ E_1]\!] =$$

$$\lambda\zeta.\mathscr{G}[\![\, E_0]\!](\lambda\rho\upsilon\sigma.\mathscr{G}[\![\, E_1]\!]\zeta\rho(\upsilon\dagger 1)\sigma).$$

$$\mathscr{G}[\![\, \mathbf{if}\ E_0\ \mathbf{then}\ E_1\ \mathbf{else}\ E_2]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\, E_0]\!](\lambda\rho\upsilon\sigma.\upsilon\downarrow 1\,|\,T\to\mathscr{G}[\![\, E_1]\!]\zeta\rho(\upsilon\dagger 1)\sigma, \mathscr{G}[\![\, E_2]\!]\zeta\rho(\upsilon\dagger 1)\sigma).$$

$$\mathscr{G}[\![\, \mathbf{while}\ E_0\ \mathbf{do}\ E_1]\!] =$$

$$\lambda\zeta.fix(\lambda\zeta'.(\lambda\zeta''.\mathscr{R}[\![\, E_0]\!](\lambda\rho\upsilon\sigma.\upsilon\downarrow 1\,|\,T\to\zeta''\rho(\upsilon\dagger 1)\sigma, \zeta\rho(\langle dummy\rangle\,\S\upsilon\dagger 1)\sigma)$$

$$(\mathscr{G}[\![\, E_1]\!](\lambda\rho'\upsilon'\sigma'.\zeta'\rho'(\upsilon'\dagger 1)\sigma'))).$$

$$\mathscr{G}[\![\, I : E]\!] =$$

$$\lambda\zeta.\mathscr{G}[\![\, E]\!]\zeta.$$

$$\mathscr{G}[\![\, I :: E]\!] =$$

$$\lambda\zeta.\mathscr{T}[\![\, E]\!]\zeta.$$

$$\mathscr{G}[\![(E)]\!] =$$

$$\lambda\zeta.\mathscr{G}[\![E]\!]\zeta .$$

$$\mathscr{P}[\![E_0 ; E_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{P}[\![E_0]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon\sigma \S \mathscr{P}[\![E_1]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{P}[\![\texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{P}[\![E_1]\!]\zeta\rho\upsilon\sigma \S \mathscr{P}[\![E_2]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{P}[\![\texttt{while } E_0 \texttt{ do } E_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{P}[\![E_1]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![\texttt{while } E_0 \texttt{ do } E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon\sigma .$$

$$\mathscr{P}[\![I:E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![E]\!]\zeta(revert\rho\rho')(pop\upsilon\upsilon')(restore\sigma\sigma'))\S\mathscr{P}[\![E]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{P}[\![I::E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{P}[\![E]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{P}[\![(E)]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{P}[\![E]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{Q}[\![E_0 ; E_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{Q}[\![E_0]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon\sigma \S \mathscr{Q}[\![E_1]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{Q}[\![\texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{Q}[\![E_1]\!]\zeta\rho\upsilon\sigma \S \mathscr{Q}[\![E_2]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{Q}[\![\texttt{while } E_0 \texttt{ do } E_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{Q}[\![E_1]\!](\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![\texttt{while } E_0 \texttt{ do } E_1]\!]\zeta\rho'(\upsilon'\dagger 1)\sigma')\rho\upsilon\sigma .$$

$$\mathscr{Q}[\![I:E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{Q}[\![E]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{Q}[\![I::E]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\rho'\upsilon'\sigma'.\mathscr{G}[\![E]\!]\zeta(revert\rho\rho')(pop\upsilon\upsilon')(restore\sigma\sigma'))\S\mathscr{Q}[\![E]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{Q}[\![(E)]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{Q}[\![E]\!]\zeta\rho\upsilon\sigma .$$

$$\mathscr{J}[\![E_0 ; E_1]\!] =$$

$$\mathscr{J}[\![E_0]\!] \S \mathscr{J}[\![E_1]\!] .$$

$$\mathcal{F}[\![\text{if } E_0 \text{ then } E_1 \text{ else } E_2]\!] =$$
$$\quad \mathcal{F}[\![E_1]\!] \S \mathcal{F}[\![E_2]\!].$$
$$\mathcal{F}[\![\text{while } E_0 \text{ do } E_1]\!] =$$
$$\quad \mathcal{F}[\![E_1]\!].$$
$$\mathcal{F}[\![I:E]\!] =$$
$$\quad \langle I \rangle \S \mathcal{F}[\![E]\!].$$
$$\mathcal{F}[\![I::E]\!] =$$
$$\quad \mathcal{F}[\![E]\!].$$
$$\mathcal{F}[\![(E)]\!] =$$
$$\quad \mathcal{F}[\![E]\!].$$

$$\mathcal{X}[\![E_0;\ E_1]\!] =$$
$$\quad \mathcal{X}[\![E_0]\!] \S \mathcal{X}[\![E_1]\!].$$
$$\mathcal{X}[\![\text{if } E_0 \text{ then } E_1 \text{ else } E_2]\!] =$$
$$\quad \mathcal{X}[\![E_1]\!] \S \mathcal{X}[\![E_2]\!].$$
$$\mathcal{X}[\![\text{while } E_0 \text{ do } E_1]\!] =$$
$$\quad \mathcal{X}[\![E_1]\!].$$
$$\mathcal{X}[\![I:E]\!] =$$
$$\quad \mathcal{X}[\![E]\!].$$
$$\mathcal{X}[\![I::E]\!] =$$
$$\quad \langle I \rangle \S \mathcal{X}[\![E]\!].$$
$$\mathcal{X}[\![(E)]\!] =$$
$$\quad \mathcal{X}[\![E]\!].$$

$$\mathcal{P}[\![\Delta]\!] =$$
$$\quad \langle\rangle.$$

$$\mathcal{D}[\![I=E]\!] =$$
$$\quad \lambda\zeta.\mathcal{L}[\![E]\!](\lambda\rho\upsilon\sigma.\zeta\rho[\upsilon\downarrow 1//I](\upsilon\dagger 1)\sigma).$$
$$\mathcal{D}[\![I_1,\ldots,I_n=E]\!] =$$
$$\quad \lambda\zeta.\mathcal{R}[\![E]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1 \mid L^*=n \rightarrow \zeta\rho[\upsilon\downarrow 1//\langle I_1,\ldots,I_n\rangle](\upsilon\dagger 1)\sigma,\tau).$$

$$\mathscr{D}[\![\mathrm{I} == \mathrm{E}]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\zeta\rho[\upsilon\downarrow 1//\mathrm{I}](\upsilon\uparrow 1)\sigma).$$

$$\mathscr{D}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n == \mathrm{E}]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1 \,|\, \mathsf{L}^* = n \to \zeta\rho[holds(\upsilon\downarrow 1)\sigma//\langle \mathrm{I}_1,\ldots,\mathrm{I}_n\rangle](\upsilon\uparrow 1)\sigma,\tau).$$

$$\mathscr{D}[\![\Delta_0 \text{ within } \Delta_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{D}[\![\Delta_0]\!](\mathscr{D}[\![\Delta_1]\!](\lambda\rho'.\zeta(trim[\![\Delta_1]\!]\rho\rho')))\rho\upsilon\sigma.$$

$$\mathscr{D}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$

$$\lambda\zeta.(\lambda\xi^*.deal\xi^*(\lambda\rho^*\rho.\zeta(pick[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!]\rho^*\rho)))$$

$$(fix(\lambda\psi m.m \le n \to \langle \lambda\zeta\rho'.\mathscr{D}[\![\Delta_m]\!](\lambda\rho''.\zeta(clip[\![\Delta_m]\!]\rho'\rho''))\rho'\rangle \S\psi(m+1),\langle\rangle)1)$$

$$\mathscr{D}[\![\text{rec } \Delta]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.(\lambda\alpha^*.(\lambda\rho'.\alpha^*:\mathsf{L}^* \to \mathscr{T}[\![\Delta]\!]\zeta\rho'\upsilon(updates\alpha^*(dummy^*)\sigma),\tau)$$

$$(\rho[\alpha^*//\mathscr{I}[\![\Delta]\!]][dummy^*//\mathscr{K}[\![\Delta]\!]]))$$

$$(news(\#\mathscr{I}[\![\Delta]\!])\sigma).$$

$$\mathscr{D}[\![(\Delta)]\!] =$$

$$\lambda\zeta.\mathscr{D}[\![\Delta]\!]\zeta.$$

$$\mathscr{T}[\![\mathrm{I} = \mathrm{E}]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.(\lambda\delta.\delta:\mathsf{L} \to \zeta\rho(\upsilon\uparrow 1)(update\delta(\upsilon\downarrow 1)\sigma),\tau)(ravel\rho[\![\mathrm{I}]\!]\downarrow 1\downarrow 1)).$$

$$\mathscr{T}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n = \mathrm{E}]\!] =$$

$$\lambda\zeta.(\lambda\zeta'.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1 \,|\, \mathsf{L}^* = n \to \zeta'\rho\upsilon\sigma,\tau))$$

$$(\lambda\rho\upsilon\sigma.(\lambda\delta^*.\delta^*:\mathsf{L}^* \to \zeta\rho(\upsilon\uparrow 1)(updates\delta^*(holds(\upsilon\downarrow 1)\sigma)\sigma),\tau)$$

$$(\langle ravel\rho[\![\mathrm{I}_1]\!]\downarrow 1\downarrow 1,\ldots,ravel\rho[\![\mathrm{I}_n]\!]\downarrow 1\downarrow 1\rangle)).$$

$$\mathscr{T}[\![\mathrm{I} == \mathrm{E}]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\zeta\rho[\upsilon\downarrow 1///\mathrm{I}](\upsilon\uparrow 1)\sigma).$$

$$\mathscr{T}[\![\mathrm{I}_1,\ldots,\mathrm{I}_n == \mathrm{E}]\!] =$$

$$\lambda\zeta.\mathscr{R}[\![\mathrm{E}]\!](\lambda\rho\upsilon\sigma.\#\upsilon\downarrow 1 \,|\, \mathsf{L}^* = n \to \zeta\rho[holds(\upsilon\downarrow 1)\sigma///\langle \mathrm{I}_1,\ldots,\mathrm{I}_n\rangle](\upsilon\uparrow 1)\sigma,\tau).$$

$$\mathscr{T}[\![\Delta_0 \text{ within } \Delta_1]\!] =$$

$$\lambda\zeta\rho\upsilon\sigma.\mathscr{D}[\![\Delta_0]\!](\mathscr{T}[\![\Delta_1]\!](\lambda\rho'.\zeta(trim[\![\Delta_1]\!]\rho\rho')))\rho\upsilon\sigma.$$

$$\mathscr{T}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$

$$\lambda\zeta.(\lambda\xi^*.deal\xi^*(\lambda\rho^*\rho.\zeta(pick[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!]\rho^*\rho)))$$

$$(fix(\lambda\psi m.m \le n \to \langle \lambda\zeta\rho'.\mathscr{T}[\![\Delta_m]\!](\lambda\rho''.\zeta(clip[\![\Delta_m]\!]\rho'\rho''))\rho'\rangle \S\psi(m+1),\langle\rangle)1)$$

$$\mathscr{T}[\![\ \mathtt{rec}\ \Delta]\!] =$$

$$\lambda\zeta.\mathscr{T}[\![\ \Delta]\!]\ \zeta\ .$$

$$\mathscr{T}[\![\ (\Delta)]\!] =$$

$$\lambda\zeta.\mathscr{T}[\![\ \Delta]\!]\ \zeta\ .$$

$$\mathscr{I}[\![\ \mathrm{I}=\mathrm{E}]\!] =$$

$$\langle\ \mathrm{I}\rangle\ .$$

$$\mathscr{I}[\![\ \mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}]\!] =$$

$$\langle\ \mathrm{I}_1,\ldots,\mathrm{I}_n\rangle\ .$$

$$\mathscr{I}[\![\ \mathrm{I}==\mathrm{E}]\!] =$$

$$\langle\rangle\ .$$

$$\mathscr{I}[\![\ \mathrm{I}_1,\ldots,\mathrm{I}_n==\mathrm{E}]\!] =$$

$$\langle\rangle\ .$$

$$\mathscr{I}[\![\ \Delta_0\ \mathtt{within}\ \Delta_1]\!] =$$

$$\mathscr{I}[\![\ \Delta_1]\!]\ .$$

$$\mathscr{I}[\![\ \Delta_1\ \mathtt{and}\ldots\mathtt{and}\ \Delta_n]\!] =$$

$$\mathscr{I}[\![\ \Delta_1]\!]\ \S\ldots\S\mathscr{I}[\![\ \Delta_n]\!]\ .$$

$$\mathscr{I}[\![\ \mathtt{rec}\ \Delta]\!] =$$

$$\mathscr{I}[\![\ \Delta]\!]\ .$$

$$\mathscr{I}[\![\ (\Delta)]\!] =$$

$$\mathscr{I}[\![\ \Delta]\!]\ .$$

$$\mathscr{K}[\![\ \mathrm{I}=\mathrm{E}]\!] =$$

$$\langle\rangle\ .$$

$$\mathscr{K}[\![\ \mathrm{I}_1,\ldots,\mathrm{I}_n=\mathrm{E}]\!] =$$

$$\langle\rangle\ .$$

$$\mathscr{K}[\![\ \mathrm{I}==\mathrm{E}]\!] =$$

$$\langle\ \mathrm{I}\rangle\ .$$

$$\mathscr{K}[\![\ \mathrm{I}_1,\ldots,\mathrm{I}_n==\mathrm{E}]\!] =$$

$$\langle\ \mathrm{I}_1,\ldots,\mathrm{I}_n\rangle\ .$$

$$\mathcal{H}[\![\Delta_0 \text{ within } \Delta_1]\!] =$$
$$\mathcal{H}[\![\Delta_1]\!].$$

$$\mathcal{H}[\![\Delta_1 \text{ and} \ldots \text{and } \Delta_n]\!] =$$
$$\mathcal{H}[\![\Delta_1]\!] \S \ldots \S \mathcal{H}[\![\Delta_n]\!].$$

$$\mathcal{H}[\![\text{rec } \Delta]\!] =$$
$$\mathcal{H}[\![\Delta]\!].$$

$$\mathcal{H}[\![(\Delta)]\!] =$$
$$\mathcal{H}[\![\Delta]\!].$$