Unbounded Nondeterminism in CSP

by A.W. Roscoe[1] and Geoff Barrett[2]

Oxford University Computing Laboratory
8-11 Keble Road
Oxford OX1 3QD

ABSTRACT. *We extend the failures/divergences model for CSP to include a component of infinite traces. This allows us to give a denotational semantics for a version of CSP including general nondeterministic choice and infinite hiding. Unfortunately the model is an incomplete partial order, so it is by no means obvious that the necessary fixed points exist. We have two proofs of this result, one via a congruence theorem with operational semantics and one via a careful analysis of operators' behaviour on a subset of the model.*

0. Introduction

As is well known to the theoretical community, it is generally far easier to model finite nondeterminism (where a process can only choose between finitely many options at any one time) than unbounded nondeterminism (where no such restriction applies). The difficulties encountered with unbounded nondeterminism have hitherto forced us to restrict the language and semantics of CSP to avoid it: the most obvious restrictions being our inability to define the hiding operator $P \backslash B$ when B is infinite and the absence of an infinite nondeterminism operator $\bigsqcap S$ for arbitrary nonempty sets S of processes.

In an earlier paper [R2] one of us showed how many of the restrictions on unbounded nondeterminism could be lifted by separating the nondeterminism order from the order used for finding fixed points. Unfortunately the structure of the model used there (failures and divergences using only finite traces) means that the semantics given by that model to unboundedly nondeterministic operators is not sufficiently discriminating. That model can successfully model a process which will, on its first step, nondeterministically choose any integer, but cannot tell between a process which can communicate any finite number of a's and one which may also choose to communicate an infinite number. The main purpose of this paper is to develop a

[1] A.W. Roscoe gratefully acknowledges support from ONR grant N00014-87-G-0242.

[2] Geoff Barrett's current address: inmos Ltd., 1000 Aztec West, Almondsbury, Bristol BS12 4SQ

more refined model which can make this sort of distinction. This is done by adding a component of infinite traces so that any CSP process is represented by $\langle F, D, I\rangle$ where F is its set of failures (still with finite traces), D is its set of (finite) divergence traces and I is the set of infinite traces it can communicate.

Unfortunately the obvious orders on this new model fail to be complete, though they do have greatest lower bounds for arbitrary nonempty sets, which means that the standard iterative technique will produce the least fixed point of monotone f provided there is any x with $f(x) \leq x$. The first reaction to this failure was to look for a new order coarser than the obvious one which was complete (for this was precisely what had been done in the paper mentioned above for the $\langle F, D\rangle$ model without the finite subsets axiom). However one can prove that no order which gives the right semantics can be complete. Specifically we find an $\omega-$sequence of CSP-definable processes whose semantic values are provably ordered in any sensible order but which can have no least upper bound.

If recursions are well defined we must therefore find some special property of CSP-definable functions which leads them to have fixed points. We have found two methods for proving the existence of these fixed points. The first was to define an operational semantics for the language and to prove simultaneously that the fixed points exist and that the denotational semantics is congruent to the operational one via the natural abstraction map. Barrett's contribution to this paper was to find a much shorter proof[3] which is also more satisfactory in some ways because it rests entirely within the model itself, rather than going outside to operational semantics.

The rest of the paper is structured as follows. In the next section we develop the new model, discover its partial order properties, and show how to define the CSP operators over it. The second section gives Barrett's proof that all CSP definable functions have fixed points. Because the semantics contains a number of features which are difficult or unusual, it is even more important than usual to have evidence that they are 'right'. For example many of the operators definable turn out to be non-continuous (though monotone) and require iteration past ω to reach their fixed points – and on first inspection it is not obvious whether the meaning of a recursion $\mu P.F(P)$ should be the ωth iterate $\bigsqcup\{F^n(\bot) \mid n \in \mathbb{N}\}$ or the least fixed point (the latter is the right answer). Therefore in the final section we outline the proof of the congruence theorem which was formerly used to prove the existence of fixed points, but now stands in its own right.

The way this congruence theorem can be used to establish the existence of fixed points is explained in an earlier presentation of the new model [R3], which also gives the proofs of various results on operational semantics that have been omitted from this article due to lack of space.

[3] The new proof was discovered in May 1989, shortly after the Tulane workshop.

1. Adding infinite traces to the failures model

The failures/divergences model, developed in [R1,B,BHR,BR], has become the standard abstract model for CSP. CSP is based on atomic, handshaken communications drawn from an alphabet Σ, which may be finite or infinite. The model describes every process as a pair $\langle F, D\rangle$, where $F \subseteq \Sigma^* \times \mathcal{P}(\Sigma)$ is the (nonempty) set of a process' *failures* ($(s, X) \in F$ is failure of the process if it can perform the trace s and then refuse to accept any communication from the set X) and $D \subseteq \Sigma^*$ is the set of its *divergences* (traces on which the process can loop – perform an infinite sequence of internal actions). The usual version of this model – often called $\mathcal{N}$ – is defined by a number of axioms, (1)-(5) below plus an axiom of bounded nondeterminism

$$\forall Y \subseteq^{\text{fin}} X.(s,Y) \in F \;\Rightarrow\; (s,X) \in F\,.$$

This axiom is necessary to make the nondeterminism partial order

$$\langle F,D\rangle \sqsubseteq \langle F',D'\rangle \Leftrightarrow F \supseteq F' \wedge D \supseteq D'$$

complete. However, in [R2], a stronger order (in that it orders less things) was developed, which gives exactly the same fixed point theory but which no longer requires this axiom for completeness. This new 'definedness' order is defined

$$\begin{aligned} P \leq Q \;\Leftrightarrow\; & \mathcal{D}[\![Q]\!] \subseteq \mathcal{D}[\![P]\!] \;\wedge \\ & s \notin \mathcal{D}[\![P]\!] \Rightarrow \mathcal{R}[\![P]\!]s = \mathcal{R}[\![Q]\!]s \;\wedge \\ & \mu(\mathcal{D}[\![P]\!]) \subseteq \mathit{traces}(Q) \end{aligned}$$

where μT denotes the minimal elements of a set T of finite traces and $\mathcal{R}[\![P]\!]s$ denotes $\{X \mid (s,X) \in \mathcal{F}[\![P]\!]\}$. $P \leq Q$ means that Q has less divergences than P, but that all of P's non-divergent behaviour is copied exactly in Q. This extended model was termed $\mathcal{N}'$.

Our new model will have the same structure as $\mathcal{N}'$ except that it will have an extra component representing infinite traces. Thus a process P will be a triple $\langle F, D, I\rangle$, where $F \subseteq \Sigma^* \times \mathcal{P}(\Sigma)$, $D \subseteq \Sigma^*$ and $I \subseteq \Sigma^\omega$. F should be nonempty and the eight axioms must be satisfied. The first seven are tabulated below.

$$\begin{array}{lrcl}
(1) & (st,\emptyset) \in F & \Rightarrow & (s,\emptyset) \in F \\
(2) & (t,X) \in F \wedge Y \subseteq X & \Rightarrow & (t,Y) \in F \\
(3) & (t,X) \in F \wedge \forall a \in Y.(t\langle a\rangle,\emptyset) \notin F & \Rightarrow & (t,X \cup Y) \in F \\
(4) & s \in D & \Rightarrow & st \in D \\
(5) & s \in D & \Rightarrow & (st,X) \in F \\
(6) & su \in I & \Rightarrow & (s,\emptyset) \in F \\
(7) & s \in D & \Rightarrow & su \in I
\end{array}$$

Here, and in the rest of this paper, $a, b, \ldots$ range over Σ, $X, Y, A, B, \ldots$ over $\mathcal{P}(\Sigma)$, s, t, v, w over the finite traces Σ^* and u over infinite traces Σ^ω.

Axioms (6) and (7) are both new but straightforward because they are simple extensions to axioms (1) and (4) respectively. One more axiom is required, which can be thought of as an infinite trace analogue to axiom (3). The latter says that anything which, on one step, cannot be refused, must be a possible communication. The new axiom will say that when one, from the finite convergent behaviour, can show that there must be infinite traces, then there are enough of them.

One can often prove from the failures of a nondivergent process that some infinite trace is possible because one can formulate a strategy for *forcing* one. The most simple-minded form of strategy is that based on a single infinite trace u. If $(s,\{a\}) \notin F$ for all $s\langle a\rangle < u$ then it is intuitively clear that a user single mindedly striving for the infinite trace u must be successful. However there are more subtle versions of this. Consider a process whose failure-set is

$$F_0 = \{(s,X) \mid s \in \{a,b\}^* \wedge \{a,b\} \not\subseteq X\}\,.$$

Imagine always offering this process the set $\{a,b\}$: it is never refused, so we can guarantee that an infinite trace must arise. However we have no finer control over exactly which infinite trace it is, though on further reflection we can observe that, since every finite sequence s of a's and bs is possible there must be an infinite trace su extending every such s. The necessity of some axiom reflecting the forcing of infinite traces is demonstrated by the definition of the hiding operator below. Studying this will reveal that if a process P with the above failures did not have an infinite trace, then $P\backslash\{a,b\}$ would not have any failures, divergences or infinite traces!

The final axiom proved quite hard to find – several quite plausible versions turned out to be incorrect. There are a number of equivalent (in the presence of the other 7) versions of axiom (8). Several are given in [R3] and [Blam]. Perhaps the nicest formulation is the following, which was derived from the first author's earlier version by Stephen Blamey. Here T ranges over finite prefix closed sets of finite traces and $\overline{T} = \{u \in \Sigma^\omega \mid \forall t < u.t \in T\}$.

$$(8) \quad (s,\emptyset) \in F \Rightarrow \exists T.(\forall t \in T.(st,\{a \mid t\langle a\rangle \notin T\}) \in F \wedge \{su \mid u \in \overline{T}\} \subseteq I)$$

This can be interpreted as saying that one can never tell on a single interaction that a process is not deterministic, unless it diverges. T represents one of the deterministic forms the process might take after the trace s. It is worth noting that this axiom in no way restricts the sets of failures that are possible – if $\langle F,D\rangle$ is any element of $\mathcal{N}'$ and $T = \{s \mid (s,\emptyset) \in F\}$ then $\langle F,D,\overline{T}\rangle$ is always in $\mathcal{U}$. Furthermore, if $\langle F,D,I\rangle \in \mathcal{U}$ and $I \subseteq I' \subseteq \overline{T}$, then $\langle F,D,I'\rangle \in \mathcal{U}$.

The reader might like to check that the elements of $\mathcal{U}$ with failure set F_0 as defined above are precisely $\langle F_0,\emptyset,I\rangle$ where I is a set of nonempty infinite traces such that every element of $\{a,b\}^*$ is a prefix of some element of I. This follows in part from the fact that, if P is a process with the given failure set and T is a prefix closed set of finite traces satisfying the conditions of axiom (8) (relative to any s), $\overline{T}$ must have an infinite trace as it is easy to prove that it has arbitrarily long finite traces. Some possible I's are $\{a,b\}^\omega$, and $\{su \mid s \in \{a,b\}^*\}$ for any fixed $u \in \{a,b\}^\omega$.

The notation of $\mathcal{N}'$ is easily extended to cover the new model. If $P = \langle F, D, I\rangle$ then we write $\mathcal{F}[\![P]\!] = F$, $\mathcal{D}[\![P]\!] = D$ and $\mathcal{I}[\![O]\!] = I$. The set $\{s \mid (s, \emptyset) \in F\}$ of finite traces of P will be denoted $\mathit{traces}(P)$; while $\mathit{Traces}(P) = \mathit{traces}(P) \cup I$ will denote the set of its finite and infinite traces.

The nondeterminism order $\sqsubseteq$ extends trivially to the new model. If $P = \langle F, D, I\rangle$ and $P' = \langle F', D', I'\rangle$ are any two triples we say

$$P \sqsubseteq P' \;\equiv\; F \supseteq F' \wedge D \supseteq D' \wedge I \supseteq I' .$$

If S is any nonempty set of processes we define its nondeterministic composition $\sqcap S$ to be $\langle F, D, I\rangle$, where

$$\begin{array}{rcl} F & = & \bigcup\{F' \mid \langle F', D', I'\rangle \in S\} \\ D & = & \bigcup\{D' \mid \langle F', D', I'\rangle \in S\} \\ I & = & \bigcup\{I' \mid \langle F', D', I'\rangle \in S\} . \end{array}$$

This is just the process which can exhibit any behaviour of any element of S. It is straightforward to verify that $\sqcap S$ is always in $\mathcal{U}$.

It is clear that $P \sqsubseteq \sqcap S$ whenever $P \in S$. We say that a process $\langle F, D, I\rangle$ is *deterministic* if it satisfies $D = \emptyset$ and

$$(s, X) \in F \Leftrightarrow X \cap \{a \mid (s\langle a\rangle, \emptyset) \in F\} . \qquad (*)$$

Thus it never has the choice of accepting or rejecting an event. The infinite traces of a deterministic process $P = \langle F, D, I\rangle$ are completely determined by the failures – $I = \overline{\mathit{traces}(P)}$. To see this, observe that the only T satisfying the first conclusion of axiom (8) for any s is $\{t \mid st \in \mathit{traces}(P)\}$. Actually, a deterministic process P is completely determined by $\mathit{traces}(P)$. The deterministic processes are precisely the greatest elements of $\mathcal{U}$ under $\sqsubseteq$ (just as over the model without infinite traces) – for it is easy to see that they *are* maximal and also to show that each process has a deterministic process above it. It is not true that for every P

$$P = \bigsqcap\{Q \mid Q \text{ is deterministic } \wedge P \sqsubseteq Q\}$$

because the right hand side always has $D = \emptyset$. However it is possible to extend the class of deterministic processes in such a way that this becomes true – say that a process $P = \langle F, D, I\rangle$ is *pre-deterministic* if whenever $s \notin D$ equation $(*)$ above holds. In other words, a process is predeterministic if it is deterministic until it diverges. We write $\mathcal{P}$ for the set of all predeterministic processes. A predeterministic process is completely determined by its sets of traces and divergences.

Ordered by $\sqsubseteq$, $\mathcal{P}$ forms a complete partial order whose least element is the immediately diverging processes and whose greatest elements are the deterministic ones. $\mathcal{P}$ plays a very important role in the next section when we come to discuss fixed points of CSP operators over $\mathcal{U}$.

We get the following fundamental result. (Indeed, before the discovery of the form of axiom (8) quoted above, this took its place.)

Lemma 1.1 For all $P \in \mathcal{U}$,

$$P = \bigsqcap imp(P), \quad \text{where} \quad imp(P) = \{Q \in \mathcal{P} \mid P \sqsubseteq Q\}\,.$$

Proof. The fact that $imp(P)$ is nonempty is a trivial consequence of axiom (8), since the T produced by the empty trace $\langle\rangle$ yields an implementation. Thus $\bigsqcap imp(P)$ is well-defined, and trivially $\bigsqcap imp(P) \sqsupseteq P$. Thus the Lemma will be proved if we can demonstrate the existence of an element of $imp(P)$ containing each behaviour (failure, divergence, infinite trace) of P. For failures it is guaranteed by axiom (8) as follows; suppose the failure is (s, X) and for any trace t, T_t is a set of traces generated by axiom (8) relative to t. Let the set S of finite traces be

$$\{t \mid t \leq s\} \cup \{t\langle a\rangle v \mid t < s \wedge t\langle a\rangle \in traces(P) \wedge t\langle a\rangle \not\leq s \wedge v \in T_{t\langle a\rangle}\}$$

$$\cup\{s\langle a\rangle v \mid a \notin X \wedge s\langle a\rangle \in traces(P) \wedge v \in T_{s\langle a\rangle}\}\,.$$

It is easy to show that the deterministic process corresponding to S is an element of $imp(P)$ with the failure (s, X). Similarly, given a divergence s, take the sets $S_1 = \{t \mid s \leq s\}$ and

$$S_2 = \{t \mid t \leq s \vee s \leq t\} \cup \{t\langle a\rangle v \mid t < s \wedge t\langle a\rangle \in traces(P) \wedge t\langle a\rangle \not\leq s \wedge v \in T_{t\langle a\rangle}\}\,.$$

The unique predeterministic process with divergence set S_1 and trace set S_2 is the required element of $imp(P)$.

It only remains to consider infinite traces u, where we define S to be

$$\{t \mid t < u\} \cup \{t\langle a\rangle v \mid t < u \wedge t\langle a\rangle \in traces(P) \wedge t\langle a\rangle \not< u \wedge v \in T_{t\langle a\rangle}\}\,.$$

The only difficulty in proving that the deterministic process determined by this trace set is in $imp(P)$ is in proving $\overline{S} \subseteq I$. This is a consequence of the fact that if $u' \in \overline{S}$ then *either* $u' = u$ *or*, letting $s\langle a\rangle$ be chosen (necessarily uniquely) such that $s\langle a\rangle < u' \wedge s < u \wedge s\langle a\rangle \not< u$, we know that $u'' \in T_{s\langle a\rangle}$, where $u' = s\langle a\rangle u''$. ▮

Since divergence (and hence undefinedness) always appears after a finite length of trace, there is no obvious way of extending the idea of definedness to infinite traces. We therefore extend $\leq$ in the same way as above: the order on the infinite traces being by reverse inclusion.

$$\begin{aligned} P \leq Q \;\Leftrightarrow\;& \mathcal{D}[\![Q]\!] \subseteq \mathcal{D}[\![P]\!] \wedge \\ & s \notin \mathcal{D}[\![P]\!] \Rightarrow \mathcal{R}[\![P]\!]s = \mathcal{R}[\![Q]\!]s \wedge \\ & \mu(\mathcal{D}[\![P]\!]) \subseteq traces(Q) \wedge \\ & \mathcal{I}[\![P]\!] \supseteq \mathcal{I}[\![Q]\!] \end{aligned}$$

In general we have $P \leq Q \Rightarrow P \sqsubseteq Q$ but not the reverse; however it is interesting to note that if P and Q are two predeterministic processes then $P \leq Q$ if and only if $P \sqsubseteq Q$. Obviously all deterministic processes are maximal under $\leq$, but there are other maximal elements as well – however they do not seem to be as useful or tangible a class as the deterministic processes and so we do not discuss them further here.

We will return to examine some more properties of the partial orders shortly.

All the usual operators may be defined over $\mathcal{U}$. As one would expect, in most cases the finite parts of these definitions are exactly the same as before (with the notable exception of hiding). They are given in full below.

STOP and *SKIP* are defined

$$STOP = \langle\{(\langle\rangle, X) \mid X \subseteq \Sigma\}, \emptyset, \emptyset\rangle$$

$$SKIP = \langle\{(\langle\rangle, X) \mid \surd \notin X\} \cup \{(\langle\surd\rangle, X) \mid X \subseteq \Sigma\}, \emptyset, \emptyset\rangle \,.$$

Let $P = \langle F, D, I\rangle$, $P' = \langle F', D', I'\rangle$ and, for $b \in B$, $P_b = \langle F_b, D_b, I_b\rangle$ be processes. Then

$$\begin{array}{rcl}
\mathcal{D}[\![a \rightarrow P]\!] & = & \{\langle a\rangle s \mid s \in D\} \\
\mathcal{I}[\![a \rightarrow P]\!] & = & \{\langle a\rangle u \mid u \in I\} \\
\mathcal{F}[\![a \rightarrow P]\!] & = & \{(\langle\rangle, X) \mid a \notin X\} \cup \{(\langle a\rangle s, X) \mid (s, X) \in F\}
\end{array}$$

$$\begin{array}{rcl}
\mathcal{D}[\![x : B \rightarrow P_x]\!] & = & \{\langle b\rangle s \mid b \in B \wedge s \in D_b\} \\
\mathcal{I}[\![x : B \rightarrow P_x]\!] & = & \{\langle b\rangle u \mid b \in B \wedge u \in I_b\} \\
\mathcal{F}[\![x : B \rightarrow P_x]\!] & = & \{(\langle\rangle, X) \mid B \cap X = \emptyset\} \cup \{(\langle b\rangle s, X) \mid b \in B \wedge (s, X) \in F_b\}
\end{array}$$

$$\begin{array}{rcl}
\mathcal{D}[\![P \sqcap P']\!] & = & D \cup D' \\
\mathcal{I}[\![P \sqcap P']\!] & = & I \cup I' \\
\mathcal{F}[\![P \sqcap P']\!] & = & F \cup F'
\end{array}$$

$$\begin{array}{rcl}
\mathcal{D}[\![P \Box P']\!] & = & D \cup D' \\
\mathcal{I}[\![P \Box P']\!] & = & I \cup I' \\
\mathcal{F}[\![P \Box P']\!] & = & \{(\langle\rangle, X) \mid (\langle\rangle, X) \in F \cap F'\} \cup \\
 & & \{(s, X) \mid s \neq \langle\rangle \wedge (s, X) \in F \cup F'\} \cup \\
 & & \{(s, X) \mid s \in \mathcal{D}[\![P \Box P']\!]\} \\
\mathcal{D}[\![P\ {}_B\|_C\ P']\!] & = & \{st \mid s \in (B \cup C)^* \wedge s\upharpoonright B \in D \wedge s\upharpoonright C \in traces(P')\} \\
 & & \cup\{st \mid s \in (B \cup C)^* \wedge s\upharpoonright B \in traces(P) \wedge s\upharpoonright C \in D'\} \\
\mathcal{I}[\![P\ {}_B\|_C\ P']\!] & = & \{u \in (B \cup C)^\omega \mid u\upharpoonright B \in Traces(P) \wedge u\upharpoonright C \in Traces(P')\} \\
 & & \cup\{su \mid s \in \mathcal{D}[\![P\ {}_B\|_C\ P']\!]\} \\
\mathcal{F}[\![P\ {}_B\|_C\ P']\!] & = & \{(s, (X \cap B) \cup (Y \cap C) \cup Z) \mid s \in (B \cup C)^* \wedge (s\upharpoonright B, X) \in F \wedge \\
 & & (s\upharpoonright C, Y) \in F' \wedge Z \cap (B \cup C) = \emptyset\} \\
 & & \cup\{(s, X) \mid s \in \mathcal{D}[\![P\ {}_B\|_C\ P']\!]\}
\end{array}$$

$$\begin{array}{rcl}
\mathcal{D}[\![P \,|||\, P']\!] & = & \bigcup\{merge\langle s, t\rangle \mid s \in D \wedge t \in traces(P')\} \\
 & & \cup\bigcup\{merge\langle s, t\rangle \mid s \in D' \wedge t \in traces(P)\} \\
\mathcal{I}[\![P \,|||\, P']\!] & = & \bigcup\{merge\langle s, t\rangle \mid s \in Traces(P) \wedge t \in Traces(P') \wedge s \text{ or } t \text{ infinite}\} \\
\mathcal{F}[\![P \,|||\, P']\!] & = & \{(s, X) \mid \exists t, t'.\, s \in merge\langle t, t'\rangle \wedge (t, X) \in F \wedge (t', X) \in F'\} \\
 & & \cup\{(s, X) \mid s \in \mathcal{D}[\![P \,|||\, P']\!]\}
\end{array}$$

$$\begin{aligned}
\mathcal{D}\llbracket P;P'\rrbracket &= \{st \mid s \in D \wedge s \text{ tick-free}\} \\
&\quad \cup\{st \mid s\langle\surd\rangle \in \mathit{traces}(P) \wedge t \in D' \wedge s \text{ tick-free}\} \\
\mathcal{I}\llbracket P;P'\rrbracket &= \{u \mid u \in I \wedge u \text{ tick-free}\} \\
&\quad \cup\{su \mid s\langle\surd\rangle \in \mathit{traces}(P) \wedge u \in I' \wedge s \text{ tick-free}\} \\
&\quad \cup\{su \mid s \in \mathcal{D}\llbracket P;P'\rrbracket\} \\
\mathcal{F}\llbracket P;P'\rrbracket &= \{(s,X) \mid (s, X \cup \{\surd\}) \in F \wedge s \text{ tick-free}\} \\
&\quad \cup\{(st,X) \mid s\langle\surd\rangle \in \mathit{traces}(P) \wedge s \text{ tick-free} \wedge (t,X) \in F'\} \\
&\quad \cup\{(s,X) \mid s \in \mathcal{D}\llbracket P;P'\rrbracket\} \\
\mathcal{D}\llbracket P\backslash X\rrbracket &= \{(u\backslash X)t \mid u \in I \wedge u\backslash X \text{ is finite}\} \cup \{(s\backslash X)t \mid s \in D\} \\
\mathcal{I}\llbracket P\backslash X\rrbracket &= \{u\backslash X \mid u \in I \wedge u\backslash X \text{ is infinite}\} \cup \{su \mid s \in \mathcal{D}\llbracket P\backslash X\rrbracket\} \\
\mathcal{F}\llbracket P\backslash X\rrbracket &= \{(s\backslash X, Y) \mid (s, X \cup Y) \in F\} \cup \{(s,Y) \mid s \in \mathcal{D}\llbracket P\backslash X\rrbracket\} \\
\mathcal{D}\llbracket f[P]\rrbracket &= \{(f(s))t \mid s \in D\} \\
\mathcal{I}\llbracket f[P]\rrbracket &= \{f(u) \mid u \in I\} \cup \{(f(s))u \mid s \in D\} \\
\mathcal{F}\llbracket f[P]\rrbracket &= \{(f(s),X) \mid (s, f^{-1}(X)) \in F\} \cup \{(s,X) \mid s \in \mathcal{D}\llbracket f[P]\rrbracket\} \\
\mathcal{D}\llbracket f^{-1}[P]\rrbracket &= \{s \mid f(s) \in D\} \\
\mathcal{I}\llbracket f^{-1}[P]\rrbracket &= \{u \mid f(u) \in I\} \\
\mathcal{F}\llbracket f^{-1}[P]\rrbracket &= \{(s,X) \mid (f(s), f(X)) \in F\}
\end{aligned}$$

We have already seen how $\bigsqcap S$ is defined.

The only definition that really requires comment is that of hiding. The definition of $\mathcal{D}\llbracket P\backslash X\rrbracket$ is rather simpler than in earlier models, since a divergence caused by the hiding now arises from a single infinite behaviour rather than from an infinite collection of finite ones. With this exception, failures and divergences never depend on the infinite traces of the operands.

We can easily extend the *after* operator to the new model. If $P = \langle F, D, I\rangle \in \mathcal{U}$ and $s \in \mathit{traces}(P)$ then P *after* s is defined to be $\langle F', D', I'\rangle$, where

$$\begin{aligned}
F' &= \{(t,X) \mid (st,X) \in F\} \\
D' &= \{t \mid st \in D\} \\
I' &= \{u \mid su \in I\}\,.
\end{aligned}$$

P *after* s is the process which behaves like P after communicating the trace s.

Theorem 1.2. All the operators are well defined (i.e., preserve the axioms) and monotonic with respect to both orders. All operators are both finitely and infinitely distributive: i.e., $F(\bigsqcap S) = \bigsqcap\{F(P) \mid P \in S\}$ for all operators F and nonempty $S \subseteq \mathcal{U}$.

We should perhaps note that no claim has been made for the continuity of the operators, which is because many of them are not continuous as a consequence of unbounded nondeterminism. The main consequence of this lack of continuity is that the fixed points of recursively defined programs need not have appeared by the ωth iteration from $\perp$ so familiar to computer scientists. However, once we can show that necessary least upper bounds exist there is no problem in defining the meaning of any recursive term to be the least fixed point of the appropriate monotone function: it is given by $f^{\alpha}(\perp)$ for sufficiently large α. Once one can do this, we can define a semantic function $\mathcal{S} : \mathbf{E} \rightarrow UEnv \rightarrow \mathcal{U}$, where $\mathbf{E}$ is the set of all CSP terms and $UEnv$ is the set of mappings from process variables to $\mathcal{U}$, in the obvious way. ∎

Properties of the partial orders

The following Lemma records some of the facts we have already noted and one or two other elementary facts about the two partial orders.

Lemma 1.3.

a) $P \sqsubseteq Q$ if, and only if, $imp(P) \supseteq imp(Q)$.

b) $P \leq Q \Rightarrow P \sqsubseteq Q$

c) $\perp = \langle \Sigma^* \times \mathcal{P}(\Sigma), \Sigma^*, \Sigma^{\omega} \rangle$ is the least element of $\mathcal{U}$ for both orders.

d) If $P \leq R$ and $P \sqsubseteq Q \sqsubseteq R$, then $P \leq Q$.

e) A process P is pre-deterministic if and only if there is a deterministic Q such that $P \leq Q$.

f) The $\sqsubseteq$-maximal elements of $\mathcal{U}$ are precisely the deterministic processes.

Proof. (a), (b) and (c) are trivial. For (d), we observe that $P \leq Q$ if and only if $P \sqsubseteq Q$ and

(i) $(s, X) \in \mathcal{F}[\![P]\!] \wedge s \notin \mathcal{D}[\![P]\!] \Rightarrow (s, X) \in \mathcal{F}[\![Q]\!]$, and

(ii) $\mu(\mathcal{D}[\![P]\!]) \subseteq traces(Q)$,

so to prove the result it will be sufficient to prove (i) and (ii). If $(s, X) \in \mathcal{F}[\![P]\!] \wedge s \notin \mathcal{D}[\![P]\!]$ then, since $P \leq R$, we know $(s, X) \in \mathcal{F}[\![R]\!]$. Hence $(s, X) \in \mathcal{F}[\![Q]\!]$ as $Q \sqsubseteq R$. Exactly the same argument applies for (ii).

Part (e) is elementary once we observe that if P is not pre-deterministic then its nondeterministic convergent behaviour must be present in any Q such that $P \leq Q$.

It is easy to show that if P and Q are both deterministic and $P \sqsubseteq Q$ then $P = Q$. It follows that if P is deterministic then $imp(P) = \{P\}$, and, by (a), that all deterministic processes are maximal. It is easy to see that, for any $P \in \mathcal{U}$, $imp(P)$ contains a deterministic process Q (since any pre-deterministic process is weaker than some deterministic one by (e)). It follows that $P \sqsubseteq Q$ and hence that no nondeterministic process can be maximal. This proves (f). ∎

We cannot hope that $\sqsubseteq$ is complete in general, for it is not complete over $\mathcal{N}'$ when Σ is infinite. Unfortunately, *neither* order is complete, even when $\Sigma = \{a, b\}$. It is easy to construct increasing $\leq$-sequences of processes, all with $F = F_0$ as defined above and $D = \emptyset$ which can have no upper bound. As a simple example, let $u_n = (\langle a\rangle^n\langle b\rangle)^\omega$ be the infinite trace which has n a's then a b cyclically. It is clear that the sets $\{su_n \mid s \in \{a,b\}^*\}$ are disjoint as n varies, and therefore that, if we set $I_n = \{su_m \mid s \in \{a,b\}^* \wedge m \geq n\}$, any upper bound for the sequence $\langle\langle F_0, \emptyset, I_n\rangle \mid n \in \mathbb{N}\rangle$ must have an empty set of infinite traces. This is impossible for $\leq$ as, since all the processes are divergence-free, any upper bound must have failure set F_0. (And we have already observed that all such elements of $\mathcal{U}$ have nonempty I.) It is also impossible for $\sqsubseteq$ since any upper bound must have an implementation Q (necessarily deterministic). Q must also be an implementation of all processes in the sequence and therefore have an infinite trace – a contradiction.

We will return to this incompleteness shortly and show that it is, to some extent at least, inevitable. Before we do this, however, it will be nice to establish a few positive properties.

Theorem 1.4

a) Any nonempty subset S of $\mathcal{U}$ has greatest lower bounds with respect to both $\leq$ and $\sqsubseteq$. In general, $\bigsqcap_{\leq} S \sqsubseteq \bigsqcap_{\sqsubseteq} S$.

b) In either order, any subset of $\mathcal{U}$ with any upper bound has a least upper bound.

c) If $\bigsqcup_{\leq} S$ is defined then so is $\bigsqcup_{\sqsubseteq} S$ and the two are equal. Furthermore $\bigsqcup_{\leq} S = P^* = \langle F^*, D^*, I^*\rangle$, where $F^* = \bigcap\{F \mid \langle F, D, I\rangle \in S\}$, $D^* = \bigcap\{D \mid \langle F, D, I\rangle \in S\}$ and $I^* = \bigcap\{I \mid \langle F, D, I\rangle \in S\}$.

d) If S is a nonempty set then $\bigsqcup_{\sqsubseteq} S$ exists if and only if $\bigcap\{imp(P) \mid P \in S\}$ is nonempty, and in that case $\bigsqcup_{\sqsubseteq} S = \bigsqcap(\bigcap\{imp(P) \mid P \in S\})$.

e) If $f : \mathcal{U} \to \mathcal{U}$ is a function which is monotone with respect to one of the orders and there is $P \in \mathcal{U}$ such that $f(P) \leq P$ (respectively $f(P) \sqsubseteq P$), then f has a least fixed point given by $f^\alpha(\bot)$ for some ordinal α.

f) If $f : \mathcal{U} \to \mathcal{U}$ is monotone with respect to both orders then any least fixed point for one order is also the least fixed point for the other.

Proof. It is easy to see that $\bigsqcap S$ is the $\sqsubseteq$-greatest lower bound of any nonempty set S. It does not work in general for the definedness order $\leq$, however, since one does not in general have $P \in S \Rightarrow \bigsqcap S \leq P$. The greatest lower bound of $S = \{\langle F_i, D_i, I_i\rangle \mid i \in \Lambda\}$ is, as was the case in $\mathcal{N}'$, constructed so that it diverges as soon as the finite behaviour of any two elements of S starts to differ. We define $\bigsqcap_{\leq} S$ to be $\langle F, D, I\rangle$, where

- $D = \bigcup\{D_i \mid i \in \Lambda\} \cup \{st \mid \exists i, j.(\exists Y.(s, Y) \in F_i \setminus F_j) \vee (\exists a.(s\langle a\rangle, \emptyset) \in F_i \setminus F_j)\}$
- $F = \bigcup\{F_i \mid i \in \Lambda\} \cup \{(s, X) \mid s \in D\}$

- $I = \bigcup\{I_i \mid i \in \Lambda\} \cup \{su \mid s \in D\}$

It is easy to show that this process is in $\mathcal{U}$ and is indeed the $\leq$ greatest lower bound of S. Trivially $\bigsqcap_{\leq} S \sqsubseteq \bigsqcap S$. This completes the proof of (a).

(b) follows because, as is fairly well known, *any* partial order which has greatest lower bounds for nonempty sets has this property. The usual argument is repeated here. If S is a set with an upper bound, then U_S, the set of upper bounds of S is nonempty and so $x = \bigsqcap U_S$ exists. Since $y \leq z$ whenever $y \in S$ and $z \in U_S$ it follows that each $y \in S$ is a lower bound for U_S. As x is the *greatest* lower bound for S it follows that $x \geq y$ for all $y \in S$ and therefore that $x \in U_S$. Plainly x is the least element of U_S and is therefore the least upper bound of S.

The first part of (c) follows trivially from the formula which is the second part. However it has an interesting separate proof. Note that, since $Q \leq P \Rightarrow Q \sqsubseteq P$, if $P = \bigsqcap_{\leq} S$ exists then it is a $\sqsubseteq$-upper bound for S and hence $Q = \bigsqcup_{\sqsubseteq} S$ exists and $Q \sqsubseteq P$. Whenever $R \in S$ we then have $R \sqsubseteq Q \sqsubseteq P$ and $R \leq P$. Lemma 1.3 (d) above then tells us that $R \leq Q$. It follows that Q is a $\leq$-upper bound for S and hence that $Q \geq P$. We then have $Q \sqsubseteq P$ and $P \sqsubseteq Q$. The result follows immediately.

For the second part, we show first that if $P' = \langle F', D', I' \rangle$ is the actual least upper bound of S then $D^* = D'$. For trivially $D' \subseteq D^*$ so let $s \in \mu D^*$ (where recall $D^* = \bigcap\{D \mid \langle F, D, I \rangle \in S\}$). Note that there must be $P = \langle F, D, I \rangle \in S$ such that $s \in \mu D$. Since $P \leq P'$ we must have $s \in \mathit{traces}(P')$. If $s \notin D'$ then consider $P'' = \langle F'', D'', I'' \rangle$ defined as

$$\begin{aligned} F'' &= F' \cup \{(st, X) \mid t \in \Sigma^* \wedge X \subseteq \Sigma\} \\ D'' &= D' \cup \{st \mid t \in \Sigma^*\} \\ I'' &= I' \cup \{su \mid u \in \Sigma^\omega\}\,. \end{aligned}$$

$\mathit{traces}(P'')$ is prefix closed by the observation above. It is thus easy to see that P'' is a process, that $P \leq P''$ for all $P \in S$ and that $P' \not\leq P''$. It follows that P' cannot be the least upper bound on S, a contradiction. Hence $\mu D^* \subseteq D'$; it easily follows that $D^* \subseteq D'$, so the two are equal as desired.

That P^* defined in the statement of the theorem satisfies axioms (1), (2), (4), (5), (6) and (7) is trivial. We next note that trivially $F^* \supseteq F'$. Now by the above paragraph those parts of F^* and F' implied by divergence and axiom (5) are equal. Suppose that $s \notin D' = D^*$. Then there is $P = \langle F, D, I \rangle \in S$ such that $s \notin D$. Necessarily $\mathcal{R}[\![P]\!]s = \mathcal{R}[\![P']\!]s$ as $P \leq P'$. It follows that $\mathcal{R}[\![P']\!]s \supseteq \mathcal{R}[\![P^*]\!]s$ (for the latter is the intersection of a set containing $\mathcal{R}[\![P]\!]s$). Putting these facts together yields $F' \supseteq F^*$, proving that in fact $F' = F^*$. Note that this implies that P^* satisfies axiom (3).

Since we have now shown that $D^* = D'$ and $F^* = F'$, and it is trivial that $I^* \supseteq I'$ and $\langle F', D', I' \rangle \in \mathcal{U}$ it follows directly that P^* satisfies axiom (8) and is therefore in $\mathcal{U}$. The fact that it is the $\leq$-least upper bound for S is then trivial. This completes the proof of (c).

(d) follows easily from Lemma 1.3 and (b) above.

(e) is true in any partial order with property (a). By another standard argument, if f is monotonic and $x = \bigsqcap\{P \mid f(P) \leq P\}$ exists in a partial order then it is the least fixed point of f. We still have to show that the least fixed point can also be found by iterating $f^\alpha(\bot)$. The only place at which the standard cpo proof of this could go wrong is where, for limit ordinals λ, one defines $f^\lambda(\bot) = \bigsqcup\{f^\alpha(\bot) \mid \alpha \in \lambda\}$ since this least upper bound might not be defined. But it always is, since it is easy to prove by transfinite induction that all the $f^\alpha(\bot)$ are bounded above by the least fixed point x constructed above so that we can always apply (b) when constructing $f^\lambda(\bot)$.

(f) follows easily from (c) and (e). If f is monotonic with respect to both orders and has any fixed point then it follows easily from (e) that it has least fixed points $f^\alpha_\leq(\bot)$ and $f^\beta_\sqsubseteq(\bot)$ with respect to these two orders. But one can prove from (c) that if both of these exist then the value of $f^\gamma(\bot)$ is independent of whether it was defined using $\leq$ or $\sqsubseteq$ by an easy transfinite induction on γ. From this it is easily seen that both processes reach the same fixed point, and do so at the same time.

(f) can alternatively be proved by observing that, by (e), if f has a fixed point then it has a least fixed point with respect to both orders. If x and y denote the $\leq$-least and $\sqsubseteq$-least fixed points respectively, we have $x \leq y$ and hence $x \sqsubseteq y$ by Lemma 1.2. But we know $y \sqsubseteq x$ so it follows that $x = y$. ∎

We should remark now that all of the properties of the partial orders identified in Lemma 1.3 and Theorem 1.4 extend easily (some of them appropriately amended) to products of $\mathcal{U}$, i.e., $\mathcal{U}^\Lambda (= \Lambda \rightarrow \mathcal{U})$ for an arbitrary nonempty set Λ, with the order $\underline{P} \leq \underline{Q}$ (or $\underline{P} \sqsubseteq \underline{Q}$) if and only if $P_\lambda \leq Q_\lambda$ (or $P_\lambda \sqsubseteq Q_\lambda$) for all $\lambda \in \Lambda$. Some of the more useful properties of these product spaces, which are important in the consideration of mutual recursions and in the definition of the partial abstraction functions later on, are summarised below. All the proofs are either standard or straightforward extensions of what we have already seen.

Theorem 1.5.

a) $\bot^\Lambda$ is least element of $\mathcal{U}$ with respect to both orders.

b) Any nonempty subset S of $\mathcal{U}^\Lambda$ has greatest lower bounds with respect to both $\leq$ and $\sqsubseteq$. In general, $\bigsqcap_\leq S \sqsubseteq \bigsqcap_\sqsubseteq S$. In either case the greatest lower bound's λ-component is given by $\bigsqcap\{P_\lambda \mid P \in S\}$, where $\bigsqcap$ here denotes the greatest lower bound operator over $\mathcal{U}$ in the appropriate order.

c) In either order, any subset of $\mathcal{U}$ with any upper bound has a least upper bound. In that case its λ-component is given by $\bigsqcup\{P_\lambda \mid P \in S\}$.

d) If $\bigsqcup_\leq S$ is defined then so is $\bigsqcup_\sqsubseteq S$ and the two are equal. Furthermore $(\bigsqcup_\leq S)_\lambda = P^*_\lambda = \langle F^*_\lambda, D^*_\lambda, I^*_\lambda \rangle$, where $F^*_\lambda = \bigcap\{\mathcal{F}[\![P_\lambda]\!] \mid \underline{P} \in S\}$, $D^*_\lambda = \bigcap\{\mathcal{D}[\![P_\lambda]\!] \mid \underline{P} \in S\}$ and $I^*_\lambda = \bigcap\{\mathcal{I}[\![P_\lambda]\!] \mid \underline{P} \in S\}$.

e) If $f : \mathcal{U}^\Lambda \rightarrow \mathcal{U}^\Lambda$ is a function which is monotone with respect to one of the orders and there is $\underline{P} \in \mathcal{U}^\Lambda$ such that $f(\underline{P}) \leq \underline{P}$ (respectively $f(\underline{P}) \sqsubseteq \underline{P}$), then f has a least fixed point given by $f^\alpha(\bot^\Lambda)$ for some ordinal α.

f) If $f : \mathcal{U}^{\mathrm{A}} \to \mathcal{U}^{\mathrm{A}}$ is monotone with respect to both orders then any least fixed point for one order is also the least fixed point for the other. ∎

These theorems and what we have shown up to now show that $\leq$ and $\sqsubseteq$ are exceptionally well-behaved partial orders. It is interesting to note that $\sqsubseteq$ has its lower bounds given by union and $\leq$ has its upper bounds given by intersection, but that the reverse facts are not true. For example $\bigsqcap\{a \to STOP, b \to STOP\} = \bot$ or $(a \to STOP) \sqcap (b \to STOP)$ depending on which order is chosen, and $\bigsqcup\{(a \to STOP) \sqcap (b \to STOP, (a \to STOP) \sqcap (b \to SKIP)\} = a \to STOP$ under $\sqsubseteq$ which is not the intersection of the two. Indeed even in cases where S is a chain, $\bigsqcup_{\sqsubseteq} S$ might exist but not be given by component-wise intersection. If P_n is the nth process in the chain seen earlier with no upper bound, then, if we define

$$Q_n = (c \to STOP) \sqcap (d \to P_n)$$

the least upper bound of this sequence is $c \to STOP$ even though $(\langle d\rangle, \emptyset)$ is a failure of every Q_n.

The first author's first reaction on finding that the two "natural" partial orders were incomplete was to try to find another one that was but which gave the same semantics. After all, that had been one of the main reasons for the development of the $\leq$ order over $\mathcal{N}'$ since it gave exactly the same least fixed point semantics but was complete, showing that all desired fixed points actually exist. We should perhaps remark at this point that the given orders do actually compute the correct values for CSP definable recursions and that the least upper bounds required to compute them always exist. Of course the proof of these facts will be the subject of much work later, but it is worthwhile seeing some examples here.

Examples. Abbreviate by a^n the process that performs n a's and then $STOP$s. Set $P = \bigsqcap\{a^n \mid n \in \mathbb{N}\}$, so that P can perform any finite number of a's but not an infinite sequence of them. Operationally we can think of P as a process which, as its first action, takes a secret decision on exactly how many a's to perform. Now consider the recursively defined process

$$Q \;=\; (a \to Q)_{\{a\}}\|_{\{a\}}P$$

and let $F : \mathcal{U} \to \mathcal{U}$ be the function associated with the right hand side of this recursion. Since the right hand side of the highest level parallel construct initially imposes a bound on the number of a's Q can perform, it is clear that Q itself cannot perform an infinite sequence of them. On the other hand it is clear that Q can perform as large a finite number of a's as it pleases. We would therefore expect $P = Q$. However, as is easily verified, $F^{\omega}(\bot)$ can perform an infinite sequence of a's (it is equal to $P \sqcap R$, where $R = a \to R$). On the other hand, $F^{\omega+1}(\bot) = (a \to (P \sqcap R))_{\{a\}}\|_{\{a\}}P = P$ and $F(P) = P$, so this recursion reaches the operationally correct fixed point at $\omega + 1$. Some more examples of recursions, their fixed points and the ordinal required to reach them are summarised below. The reader might enjoy constructing a few of his own.

- If $f : \Sigma \to \Sigma$ is such that $f^n(a) \neq f^m(a)$ when $n \neq m$ then the recursion

$$Q_1 = STOP \sqcap a \to ((Q_1\, {}_{\Sigma}\|_{\Sigma}\, P) \sqcap f(Q_1))$$

(with P as above) reaches its fixed point (which is the same as that of the recursion $P' = P \sqcap a \rightarrow f[P']$ which converges in ω steps), in exactly $\omega.2$ iterations.

- Let α be an infinite ordinal and $\Sigma = \alpha$ (the set of all $\beta < \alpha$). Then the recursion

$$Q_2 = \beta : \alpha \rightarrow ((\gamma : \beta \rightarrow STOP)\ {}_{\Sigma}\|_{\Sigma}\ Q_2) \backslash \alpha$$

takes exactly α steps to converge to its fixed point $\beta : \alpha \rightarrow STOP$. Q_2 is a process that inputs any element β of α and then outputs any element of β to a copy of itself or deadlocks if $\beta = 0$. (The fact that this is the natural fixed point is an easy consequence of the fact that there is no infinite descending sequence of ordinals.)

Suppose $\preceq$ is some partial order which does all we want: namely give the same fixed point theory and make $\mathcal{U}$ complete. Clearly it must make all CSP operators monotonic and have the same minimal element $\perp$. To give the same fixed point theory it must have the property that, when C is a linearly ordered subset of $\mathcal{U}$ with respect to $\preceq$ and one of our existing orders, then a least upper bound for $\preceq$ is also a least upper bound for the other. (Note that $\sqsubseteq$ and $\leq$ are in this relationship.) It must also make $P' \prec Q$, where Q is defined as in the example above and $P' = STOP \sqcap a \rightarrow P'$. For Q is a fixed point of this recursion but is distinct from the natural fixed point (by assumption the $\prec$-least) which has the infinite sequence of a's. ($P' \prec Q$ can also be proved by looking at the recursion of Q, where P' is the ωth iterate.)

From these simple facts and assumptions we will be able to prove that $\preceq$ cannot exist: for there is a sequence of processes in $\mathcal{U}$ which are provably ordered by $\preceq$ but which can have no upper bound. Set $\Sigma = \{a, b\}$. Recall that the set F_0 of failures was defined

$$F_0 = \{(s, X) \mid s \in \{a, b\}^* \wedge \{a, b\} \not\subseteq X\}\,.$$

The corresponding set where a process can refuse anything at any time is

$$F_1 = \{(s, X) \mid s \in \{a, b\}^* \wedge X \subseteq \{a, b\}\}\,.$$

Recall that the triples $\langle F_0, \emptyset, I\rangle$ satisfying the axioms were those where I contains an extension of every finite trace. All triples $\langle F_1, \emptyset, I\rangle$ satisfy the axioms.

We will now construct some subsets of $\{a, b\}^\omega$ to go along with F_0 and F_1. If $u \in \{a, b\}^\omega$ and $n \in \mathbb{N}$, define $r_n(u)$ to be the ratio of the number of a's to the number of b's plus one in the first n elements of u. (The "plus one" is to make this always defined.) We should perhaps remark that some traces u have $\lim_{n\rightarrow\infty} r_n(u)$ existing and some do not. (In fact, there are uncountably many u's with any given limit in $[0, \infty)$.) In the first author's experience the ratios $r_n(u)$ are very useful when it comes to choosing pathological subsets of $\{a, b\}^\omega$ and similar.

For $n \in \{1, 2, 3, \ldots\}$ we define

$$I_n \;=\; \{u \in \{a, b\}^\omega \mid \exists \epsilon > 0.\exists m.\forall k \geq m.\, \epsilon < r_k(u) < \frac{1}{n} - \epsilon\}\,.$$

Thus $u \in I_n$ if and only if the ratios eventually stay within $(0, \frac{1}{n})$ and away from the boundaries of that interval. This last condition means, amongst other things, that I_n contains no sequence with limit 0 or $\frac{1}{n}$. Notice that $u \in I_n$ does not imply that $\lim_{n \to \infty} r_n(u)$ exists. The sets I_n have some interesting properties. First, the I_n all contain elements beginning with any chosen $s \in \{a,b\}^*$ (in fact, uncountably many). Also $I_{n+1} \subseteq I_n$ and $\bigcap\{I_n \mid n \in \{1,2,\ldots\}\} = \emptyset$. Perhaps the most interesting property is that, if $m \leq n$ then

$$\bigcup\{merge\langle s,t\rangle \mid s \in I_n \cup \{a,b\}^* \wedge t \in I_m \cup \{a,b\}^* \wedge s \text{ or } t \text{ is infinite}\} = I_m \, .$$

Also, the insertion or deletion of finitely many elements of a sequence u does not effect membership of any I_n since the limiting behaviour $r_n(u)$ is not affected by such manipulations. We can now define some processes

$$\begin{array}{rcll} P_n & = & \langle F_0, \emptyset, I_n\rangle & \text{for } n \in \{1,2,3,\ldots\} \\ Q_n & = & \langle F_1, \emptyset, I_n\rangle & \text{for } n \in \{1,2,3,\ldots\} \\ P_0 & = & \langle F_0, \emptyset, \{a,b\}^\omega\rangle & \\ Q_0 & = & \langle F_1, \emptyset, \{a,b\}^\omega\rangle & \\ Q_\infty & = & \langle F_1, \emptyset, \emptyset\rangle & \end{array}$$

We will prove that the P_n are a $\preceq$-increasing sequence.

Now if $f : \Sigma \to \Sigma$ is defined by $f(a) = f(b) = a$, we have $f^{-1}[P'] = Q_0$ and $f^{-1}[Q] = Q_\infty$, where P' and Q are as described at the start of this discussion. Hence $Q_0 \preceq Q_\infty$ as f^{-1} is monotonic.

Now for all n it is not too hard to see that $P_n \,|||\, Q_0 = P_0$ and $P_n \,|||\, Q_\infty = P_n$. It follows that $P_0 \preceq P_n$ for all $n \geq 1$ as $|||$ is monotonic.

Next, observe that $P_n \,{}_\Sigma\|_\Sigma\, P_m = Q_n$ if $m \leq n$. (The transition from F_0 to F_1 arises because one side of the parallel may refuse a and the other b.) It follows that $Q_m = (P_0 \,{}_\Sigma\|_\Sigma\, P_m) \preceq (P_n \,{}_\Sigma\|_\Sigma\, P_m) = Q_n$ when $m \leq n$.

The property of the I_n described above implies that $P_m \,|||\, Q_n = P_k$, where k is the lesser of n and m. Hence, when $m \leq n$, $P_m = P_n \,|||\, Q_m \preceq P_n \,|||\, Q_n = P_n$. This completes the proof that the P_n form an increasing sequence.

The fact that the P_n are $\preceq$-increasing is unsurprising, since they are increasing with respect to $\sqsubseteq$ and $\leq$. We have specified that all $\preceq$ least upper bounds of sequences increasing in both orders are also $\sqsubseteq$ least upper bounds. Since $\bigcap\{I_n \mid n \in \mathbb{N}\}$ is empty, any $\sqsubseteq$ least upper bound for this sequence has $I = \emptyset$. But there is no element of $\mathcal{U}$ with $F \subseteq F_0$ and $I = \emptyset$. It follows that this sequence has no upper bound with respect to $\preceq$. Therefore $\preceq$ cannot be complete.

We therefore have to give up all hope of a conventional fixed point theory, though note that, by Theorem 1.4, if we can show every CSP term has some fixed point, or even maps some point down in either order, then we essentially have one. The first author's proof that these fixed points exist was via a congruence theorem with operational semantics; this was both complex and, because it relied on structures outside the model, not fully satisfying. Recently the second author has discovered a much simpler proof, within the model, which is described in the next section.

2. The fixed point theorem

To show that all the CSP operators have least fixed points, we appeal to a sort of 'dominated convergence theorem', which states that if $F \sqsubseteq G$ and G has a fixed point, ϕG, then F has a least fixed point for:

$$F(\phi G) \sqsubseteq G(\phi G) = \phi G$$

so that the fixed point of G is mapped down by F and then Theorem 1.4 (e) implies the existence of a least fixed point for F.

The usefulness of this observation is that we may find a dominating G which is monotonic and preserves predeterminism so that the completeness of that subspace guarantees a fixed point for G. Indeed, we can go one step further for, suppose we can find a monotonic function $G : \mathcal{P} \rightarrow \mathcal{P}$ such that $F{\restriction}\mathcal{P} \sqsubseteq G$, then we may extend G by:

$$G^{\star}(P) = \bigsqcap \{G(Q) \mid Q \in \mathit{imp}(P)\}$$

which agrees with G on $\mathcal{P}$ since if $P \in \mathcal{P}$, then $P \in \mathit{imp}(P)$ and since G is monotonic, $G(P) \sqsubseteq G(Q)$ for all $Q \in \mathit{imp}(P)$ giving $G^{\star}(P) = G(P)$. Furthermore, G dominates F everywhere for:

$$\begin{aligned} F(P) &= F\left(\bigsqcap \mathit{imp}(P)\right) \\ &\sqsubseteq \bigsqcap \{F(Q) \mid Q \in \mathit{imp}(P)\} \\ &\sqsubseteq \bigsqcap \{G(Q) \mid Q \in \mathit{imp}(P)\} \\ &= G^{\star}(P) \end{aligned}$$

We know that any fixed point of G is a fixed point of $G^{\star}$ and that G has a fixed point, so we can now see that F has a least fixed point.

Let us see this restriction in action. We will exhibit a monotonic function with no fixed point and show that its restriction has no dominating function.

$$F(X) = \begin{cases} (a \rightarrow X)_{\Sigma}\|_{\Sigma} P, & P \not\sqsubseteq X \\ a \rightarrow X, & P \sqsubseteq X \end{cases}$$

where

$$P = \bigsqcap_{n<\omega} a^n$$

This function really is monotonic for if $X \sqsubseteq Y$ and $P \sqsubseteq X$ or $P \not\sqsubseteq Y$, then both X and Y follow the same branch of F; otherwise $P \not\sqsubseteq X$ and $P \sqsubseteq Y$ and

$$F(X) = (a \rightarrow X)_{\Sigma}\|_{\Sigma} P \sqsubseteq (a \rightarrow Y)_{\Sigma}\|_{\Sigma} P \sqsubseteq a \rightarrow Y = F(Y)$$

The chain got by applying this function to $\perp$ is:

$$\begin{array}{c}\perp\\ STOP\sqcap a\perp\\ \vdots\\ \bigsqcap_{k<n} a^k\sqcap a^n\perp\\ \vdots\\ P\\ \vdots\\ \bigsqcap_{n\le k<\omega} a^k\\ \vdots\end{array}$$

which has no supremum since any limit would be unable to refuse $\{a\}$ at any time but would not have the infinite sequence of a's among its infinite traces. If we restrict ourselves to a model whose alphabet is just $\{a\}$, the only predeterministic processes are a^n, a^ω and $a^n\perp$ and the application of F to each of these gives:

$$\begin{array}{rcl}F(a^n) & = & a^{n+1}\\ F(a^\omega) & = & P\\ F(a^n\perp) & = & a^{n+1}\perp\sqcap\bigsqcap_{k\le n} a^k\end{array}$$

We know that any dominating G with a fixed point must fix one of the predeterministic processes. However, there is no predeterministic process which is mapped down by F so we cannot find such a G. (Clearly F would map any fixed point of G down, because G is assumed to dominate F.)

Proceeding with the proof, note that composition is a monotonic function on the function space of a partial order. Therefore, if two CSP functions are dominated by predeterminism-preserving functions, the composition is, also. Further, the property is preserved through recursion because if $F(P,Q)$ is dominated by $G(P,Q)$ and $Q\in\mathcal{P}$, then $\mu p.F(p,Q)$ exists (by the argument given earlier) and is dominated by $\mu p.G(p,Q)$, which is predeterministic.

So, we only need to show that the restriction of each primitive CSP function to $\mathcal{P}$ is dominated by a predeterminism-preserving function. We list a set of algebraic laws which the CSP operators satisfy and which show just where the operators introduce nondeterminism and use these laws to motivate the definition of a bounding function for each CSP operator. Each of the following operators is strict and distributive in each of its arguments. If we let P abbreviate $x:B\rightarrow P_x$ and Q abbreviate $y:C\rightarrow Q_y$, we have:

$$P\Box Q = z:(B\cup C)\rightarrow R_z$$
$$where\quad R_z=\begin{cases}P_z, & z\in B-C\\ P_z\sqcap Q_z, & z\in B\cap C\\ Q_z, & z\in C-B\end{cases}$$

$$\begin{aligned} P_X\|_Y Q &= z : D \to R_z \\ \textit{where} \quad D &= (B \cap (X - Y)) \cup (B \cap C \cap X \cap Y) \cup (C \cap (Y - X)) \\ R_z &= \begin{cases} P_{zX}\|_Y Q, & z \in B \cap (X - Y) \\ P_{zX}\|_Y Q_z, & z \in B \cap C \cap X \cap Y \\ P_X\|_Y Q_z, & z \in C \cap (Y - X) \end{cases} \end{aligned}$$

$$P \,|||\, Q = \left(x : B \to \left(P_x \,|||\, Q\right)\right) \square \left(y : C \to \left(P \,|||\, Q_y\right)\right)$$

$$P;Q = \begin{cases} x : B \to (P_x;Q), & \surd \notin B \\ ((x : B - \{\surd\} \to (P_x;Q)) \square Q) \sqcap Q, & \surd \in B \end{cases}$$

$$P\backslash X = \begin{cases} x : B \to (P_x\backslash X), & B \cap X = \emptyset \\ ((x : B - X \to (P_x\backslash X)) \square Q) \sqcap Q, & B \cap X \neq \emptyset \end{cases}$$
$$\textit{where} \quad Q = \textstyle\bigsqcap\{P_x\backslash X \mid x \in B \cap X\}$$

$$f[P] = y : f(B) \to \bigsqcap\left\{f[P_x] \mid x \in f^{-1}(y) \cap B\right\}$$

$$f^{-1}[P] = y : f^{-1}(B) \to f^{-1}\left[P_{f(y)}\right]$$

It is helpful to note that any predeterministic process P is either $\perp$ or can be written $x : B \to P_x$ for some set $B \subseteq \Sigma$ and predeterministic processes P_x. For those operators which introduce nondeterminism on $\mathcal{P}$, we aim to define new operators which make a particular nondeterministic choice. This is done by cases. We give below the equations which we expect the dominating operators to satisfy. In fact, we define the new operators (over $\mathcal{P}$ only) to be the least ones which satisfy the given equations.

C (a constant process) Let Q be some fixed predeterministic implementation of C. Dominating function: Q.

$F(P)$ **where** $F_\lambda(\underline{P}) = P_{\mu_\lambda}$ (This function is needed so that functions such as $F(p) = p$ and $F(p) = p_X\|_Y p$ can be written as compositions of primitive functions; it is a sort of syntactic glue.) Dominating function: itself.

$x : B \to P_x$ Dominating function: itself.

$\bigsqcap_{\lambda \in \Lambda} P_\lambda$ Choose $\lambda_0 \in \Lambda$. Dominating function: P_{λ_0}.

$P \square Q$ Dominating function: $P \boxminus Q$ defined to be bi-strict (*i.e.*, $P \boxminus \perp$ and $\perp \boxminus Q$ are both $\perp$) and to satisfy:

$$\begin{aligned} (x : B \to P_x) \boxminus (y : C \to Q_y) &= z : (B \cup C) \to R_z \\ \textit{where} \quad R_z &= \begin{cases} P_z, & z \in B \\ Q_z, & z \in C - B \end{cases} \end{aligned}$$

$P_X\|_Y Q$ Dominating function: itself.

$P \mathbin{|||} Q$ Dominating function: $P \mathbin{+\!\!\!+\!\!\!+} Q$ defined to be bi-strict and to satisfy:

$$P \mathbin{+\!\!\!+\!\!\!+} Q = \left(x : B \rightarrow \left(P_x \mathbin{+\!\!\!+\!\!\!+} Q\right)\right) \boxminus \left(y : C \rightarrow \left(P \mathbin{+\!\!\!+\!\!\!+} Q_y\right)\right)$$

where $P = x : B \rightarrow P_x$ and $Q = y : C \rightarrow Q_y$.

$P ; Q$ Nondeterminism is only introduced when P offers termination and other events. We choose to make it terminate immediately. Dominating function: $P \boxed{;} Q$ defined to be strict in its first argument and to satisfy:

$$(x : B \rightarrow P_x) \boxed{;} Q = \begin{cases} x : B \rightarrow \left(P_x \boxed{;} Q\right), & \surd \notin B \\ Q, & \surd \in B \end{cases}$$

$P \backslash X$ Nondeterminism arises through choices of hidden events. Let c be a choice function on X. Dominating function: $P \backslash^{c} X$ defined to be strict and the least operator to satisfy:

$$(x : B \rightarrow P) \backslash^{c} X = \begin{cases} x : B \rightarrow (P \backslash^{c} X), & B \cap X = \emptyset \\ P_{c(B \cap X)} \backslash^{c} X, & B \cap X \neq \emptyset \end{cases}$$

$f\,[P]$ Nondeterminism is introduced by mapping two events to the same event. Let c be a choice function on the domain of f. Dominating function: $f\,\{P\}$ defined to be strict and to satisfy:

$$f\,\{x : B \rightarrow P_x\} = x : f\,(B) \rightarrow f\left\{P_{c(f^{-1}(x) \cap B)}\right\}$$

$f^{-1}\,[P]$ Dominating function: itself.

We must now verify that the functions $\boxminus$, $\boxed{;}$, $\mathbin{+\!\!\!+\!\!\!+}$, $\backslash^{c}$ and $f\,\{\cdot\}$ exist and that they do indeed bound the CSP operators on $\mathcal{P}^{\Lambda}$. First note that if Y is complete (consistently complete) and X is a partial order, then the space of monotonic functions $X \rightarrow Y$ is also complete (consistently complete). Each of the above operators is defined to be the least fixed point of some function on $\mathcal{P}^{\Lambda} \rightarrow \mathcal{P}$, *e.g.* $\backslash^{c} X$ is the least fixed point of the function F where:

$$\begin{aligned} F\,G(\bot) &= \bot \\ F\,G(x : B \rightarrow P_x) &= \begin{cases} x : B \rightarrow G\,(P_x), & B \cap X = \emptyset \\ G\left(P_{c(B \cap X)}\right), & B \cap X \neq \emptyset \end{cases} \end{aligned}$$

That F has a least fixed point which is a monotonic function $\mathcal{P} \rightarrow \mathcal{P}$ can be seen because if G is monotonic then $F\,G$ is monotonic and if G is predeterminism-preserving then so is $F\,G$. Furthermore, if $G \sqsubseteq G'$ then $F\,G \sqsubseteq F\,G'$ so that F is a monotonic function on the complete space $\mathcal{P} \rightarrow \mathcal{P}$.

We now turn to showing that these operators dominate the CSP operators on $\mathcal{P}^{\Lambda}$. We continue with the example of hiding. First of all we note that $\backslash X$ is a fixed point of a monotonic function F' on $\mathcal{P} \rightarrow \mathcal{U}$ given by the algebraic law above. Since

$F' \, G(P) \sqsubseteq F \, G(P)$ for all $P \in \mathcal{P}$ and $G : \mathcal{P} \rightarrow \mathcal{U}$ we have that $F' \sqsubseteq F$. Now, since the monotonic functions $\mathcal{P} \rightarrow \mathcal{U}$ form a consistently complete space, we may infer the existence of a least fixed point of F' which is weaker than the least fixed point of F. All we have to do is to show that the CSP operators (restricted to $\mathcal{P}$) are indeed the least fixed points of those laws. The rest of this proof is devoted to establishing this fact. We shall refer to the algebraic laws as the fixed point equations for the operators.

We can put a sort of metric on the space as follows. We first define the nth-restriction operator, $P \downarrow n$, which gives a process which behaves like P for the first n steps and then diverges:

$$\begin{aligned}
\mathcal{D}[\![P \downarrow n]\!] &= \{st \mid s \in traces P \wedge \#s \geq n\} \cup \mathcal{D}[\![P]\!] \\
\mathcal{F}[\![P \downarrow n]\!] &= \mathcal{F}[\![P]\!] \cup \{(s, X) \mid s \in \mathcal{D}[\![P \downarrow n]\!]\} \\
\mathcal{I}[\![P \downarrow n]\!] &= \{u \in \alpha P^{\omega} \mid \forall s < u. \, s \in traces P \downarrow n\}
\end{aligned}$$

The 'metric' is defined by:

$$d(P, Q) = \inf \left\{2^{-n} \mid n < \omega \wedge P \downarrow n = Q \downarrow n\right\}$$

which satisfies the ultra-metric form of the triangle inequality, namely:

$$d(P, R) \leq \max\left(d(P, Q), d(Q, R)\right)$$

but if $d(P, Q) = 0$, we may only deduce that the failure and divergence sets of P and Q are equal. In fact, if we define the closure of a process, $\overline{P}$, to be P with all possible infinite traces, *i.e*:

$$\begin{aligned}
\mathcal{I}[\![\overline{P}]\!] &= \{u \in \alpha P^{\omega} \mid \forall s < u. \, s \in traces P\} \\
\mathcal{D}[\![\overline{P}]\!] &= \mathcal{D}[\![P]\!] \\
\mathcal{F}[\![\overline{P}]\!] &= \mathcal{F}[\![P]\!]
\end{aligned}$$

then we notice that the distance between processes is equal just when their closures are equal:

$$d(P, Q) = 0 \Leftrightarrow \overline{P} = \overline{Q}$$

This sort of 'metric' is usually known as a *pseudo-metric*.

Note also that the closure of a process is always weaker than the process in both orderings.

Since the pseudo-metric is bounded (it is never bigger than 1), we may define a corresponding pseudo-metric on any function space $X \rightarrow \mathcal{U}$ by the usual construction:

$$d(f, g) = \sup_{x \in X} d(f(x), g(x))$$

and note that $d(f, g) = 0$ just when $\overline{f} = \overline{g}$ where $\overline{f}(x) = \overline{f(x)}$.

If we study the fixed point equations for the operators, we find that in all cases except that for hiding, the recursions are 'guarded'. That is, all the recursions are given in the form $F\,G\,P = P'$ and we can easily show that $d(F\,G, F\,G') \leq \frac{1}{2}d(G, G')$ so that if we have two operators G and G' which satisfy the equations then

$$d(G, G') = d(F\,G, F\,G') \leq \frac{1}{2}d(G, G')$$

giving $d(G, G') = 0$ and so $\overline{G} = \overline{G'}$. In all cases except hiding and forward renaming ($f\,[-]$) the result of applying the operators to a tuple of predeterministic processes is a closed process (as can be verified by inspection of the operator definitions in the last section – effectively this is because these are the only operators other than $\sqcap$ which can introduce unbounded nondeterminism). This means that all possible infinite traces are present and there can be no smaller fixed point.

Of these guarded recursions we are only left to dispose of forward renaming. Since any other fixed point has the same divergence and failure set, we need only consider the non-divergent infinite traces. Suppose G satisfies the equation and that u is a non-divergent infinite trace of $G\,(P)$. We shall construct a sequence of traces s_i such that $s_i < s_{i+1}$, $f\,(s_i)\,u_i = u$ and u_i is an infinite trace of $G\,(P\ \mathit{after}\ s_i)$. Then the existence of the traces s_i imply that P must have an infinite trace whose image under f is u, so u is an infinite of $f\,[P]$.

We take s_0 to be $\langle\rangle$ and $u_0 = u$. Since u is non-divergent, s_n is not a divergence of $G\,(P\ \mathit{after}\ s_n)$, therefore, there is a B such that $P\ \mathit{after}\ s_n = x : B \rightarrow P\ \mathit{after}\ s_n\,\langle x\rangle$ so that the equation which G satisfies tells us that:

$$G\,(P\ \mathit{after}\ s_n) = y : f\,(B) \rightarrow \bigsqcap\left\{G\,(P\ \mathit{after}\ s_n\,\langle x\rangle) \mid x \in f^{-1}\,(y) \cap B\right\}$$

Now, if b is the first element of u_n, then $b \in f\,(B)$ and there must be some $a \in f^{-1}\,(b) \cap B$ such that the tail of u_n is an infinite trace of $G\,(P\ \mathit{after}\ s_n\,\langle a\rangle)$. We take $s_{n+1} = s_n\,\langle a\rangle$ and u_{n+1} to be the tail of u_n. Lastly, $f\,(s_{n+1})\,u_{n+1} = f\,(s_n)\,u_n = u$ as required.

The only operator left is hiding (whose fixed point equation is not guarded). We will assume that the set X to be hidden is nonempty, the result being trivial if $X = \emptyset$ since both $\backslash\emptyset$ and $\backslash^c\emptyset$ are the identity function. The first observation to make is that the fixed point theory of $\leq$ and $\sqsubseteq$ are the same. This follows because each fixed point equation is $\leq$-monotonic and preserves $\leq$-monotonic functions; since $\perp$ is the least element for each order and $\bigsqcup_{\leq} S = \bigsqcup_{\sqsubseteq} S$ whenever the first exists, the standard iterative technique of finding the least fixed point must produce the same result. Now, if G is the least fixed point of the equation for hiding, then $G \leq \backslash X$. All we have to do is check that the minimal divergences of any fixed point are divergences of $\backslash X$ and that convergent infinite traces of a fixed point are infinites of $\backslash X$. Both parts of the proof are achieved by a construction similar to that which we used for the forward renaming operator.

If $t \in \mu\mathcal{D}[\![G\,(P)]\!]$ ($u \in \mathcal{I}[\![G\,(P)]\!]$), the idea is to construct a sequence of traces s_i such that $s_i < s_{i+1}$ and

$$\begin{array}{cc} s_i\backslash X t_i = t & (s_i\backslash X u_i = u) \\ t_i \in \mu\mathcal{D}[\![G\,(P\ \mathit{after}\ s_i)]\!] & (u_i \in \mathcal{I}[\![G\,(P\ \mathit{after}\ s_i)]\!]) \end{array}$$

for then P has an infinite trace, v, which is the limit of the s_i with $v\backslash X = t$ ($v\backslash X = u$) giving $t \in \mathcal{D}[\![P\backslash X]\!]$ ($u \in \mathcal{I}[\![P\backslash X]\!]$).

Choose $s_0 = \langle\rangle$ and $t_0 = t$ ($u_0 = u$). We define the $n+1$th sequences from the nth. If $P\,\mathit{after}\,s_n = \bot$ then we must have $t_n = \langle\rangle$ so we may set s_{n+1} to be any extension of s_n by an element of X (the situation cannot arise if u is a non-divergent infinite); otherwise $P_n\,\mathit{after}\,s_n = x : B \rightarrow P\,\mathit{after}\,s_n\,\langle x\rangle$ for some B. If $B \cap X = \emptyset$, then

$$G\,(P\,\mathit{after}\,s_n) = x : B \rightarrow G\,(P\,\mathit{after}\,s_n\,\langle x\rangle)$$

by the fixed point equation so there must be a $b \in B$ and t_{n+1} (resp. u_{n+1}) such that $t_n = \langle b\rangle\,t_{n+1}$ (resp. $u_n = \langle b\rangle\,u_{n+1}$) and $t_{n+1} \in \mu\mathcal{D}[\![G\,(P\,\mathit{after}\,s_n\,\langle b\rangle)]\!]$ (resp. $u_{n+1} \in \mathcal{I}[\![G\,(P\,\mathit{after}\,s_n\,\langle b\rangle)]\!]$) so take $s_{n+1} = s_n\,\langle b\rangle$.

If $B \cap X \neq \emptyset$, then

$$G\,(P\,\mathit{after}\,s_n) = ((x : B - X \rightarrow G\,(P\,\mathit{after}\,s_n\,\langle x\rangle))\,\Box Q)\,\sqcap Q$$

where

$$Q = \bigsqcap_{x \in B \cap X} G\,(P\,\mathit{after}\,s_n\,\langle x\rangle)$$

so that either the required behaviour comes from performing some action from $B - X$ immediately, in which case we employ the same construction as in the last case; or else, the behaviour comes from $G\,(P\,\mathit{after}\,s_n\,\langle b\rangle)$ for some $b \in B \cap X$, in which case we define $s_{n+1} = s_n\,\langle b\rangle$ and $t_{n+1} = t_n$ ($u_{n+1} = u_n$).

The results of this section are summarised in the following theorem.

2.1 Theorem. Every CSP definable function has a least fixed point, and therefore its denotational semantics over $\mathcal{U}$ is well-defined. ∎

3. Operational semantics

In this section we present an operational semantics for CSP with unbounded nondeterminism, and summarise the main details of the congruence proof referred to in the introduction. But first we will define the abstraction functions from transition systems to $\mathcal{U}$ that will play a crucial role in the statement and proof of this theorem. The proofs of all but a very few results are omitted from this presentation – readers wishing more details should consult [R3].

Summary of notation, nomenclature and results. A *transition system* is a set of states with a binary relation $\xrightarrow{\delta}$ for each element δ of the set $\Sigma^+ = \Sigma \cup \{\tau\}$ of transitions, where τ denotes an internal transition. We should note that Σ (the set of visible actions) is an implicit parameter of almost everything we do from now on, as indeed it was in the last section.

A *morphism* [R1,R4] is a function from one transition system to another which characterises the property of indistinguishability in that no experimenter who can only see transitions (visible or invisible) should be able to tell P from $F(P)$ if F is a morphism. $F : C \rightarrow D$ is said to be a morphism if and only if:

(i) $P \xrightarrow{\delta} Q \;\Rightarrow\; F(P) \xrightarrow{\delta} F(Q)$, and

(ii) $F(P) \xrightarrow{\delta} X \;\Rightarrow\; \exists Q.P \xrightarrow{\delta} Q \wedge F(Q) = X$.

Morphisms are closely related to the idea of bisimulation but differ in that they treat internal actions in exactly the same rigid way that they treat visible ones, and that they are functions rather than relations.

The *index of nondeterminism* $i(C)$ of a transition system C is the smallest infinite regular cardinal[4] which is strictly larger than $card\{Q \mid P \xrightarrow{\delta} Q\}$ for all $P \in C$ and $\delta \in \Sigma^+$.

The functions. Given an element P of a transition system, we can construct its sets of failures, divergence and infinite traces in natural ways which are described below.

We first define two multi-step versions of the transition relation. If $P, Q \in C$ and $s = \langle x_i \mid 0 \leq i < n\rangle \in (\Sigma^+)^*$ we say $P \overset{s}{\longmapsto} Q$ if there exist $P_0 = P, P_1, \ldots, P_n = Q$ such that $P_k \xrightarrow{x_k} P_{k+1}$ for $k \in \{0, 1, \ldots, n-1\}$. Unlike this first version, the second ignores τs. For $s \in \Sigma^*$ we write $P \overset{s}{\Longrightarrow} Q$ if there exists $s' \in (\Sigma^+)^*$ such that $P \overset{s'}{\longmapsto} Q$ and $s' \setminus \tau = s$. The following properties of $\overset{s}{\Longrightarrow}$ and $\overset{s}{\longmapsto}$ are all obvious.

Lemma 3.1.

(a) $P \overset{\langle\rangle}{\Longrightarrow} P \wedge P \overset{\langle\rangle}{\longmapsto} P$

(b) $P \overset{s}{\Longrightarrow} Q \wedge Q \overset{t}{\Longrightarrow} R$ implies $P \overset{st}{\Longrightarrow} R$

(c) $P \overset{s}{\longmapsto} Q \wedge Q \overset{t}{\longmapsto} R$ implies $P \overset{st}{\longmapsto} R$

(d) $P \overset{st}{\Longrightarrow} R$ implies $\exists Q.P \overset{s}{\Longrightarrow} Q \wedge Q \overset{t}{\Longrightarrow} R$

(e) $P \overset{st}{\longmapsto} R$ implies $\exists Q.P \overset{s}{\longmapsto} Q \wedge Q \overset{t}{\longmapsto} R$ ∎

Suppose C is a transition system and $P \in C$. We say P can *diverge*, written $P\uparrow$, if there exist $P_0 = P, P_1, P_2, \ldots$ such that, for all $n \in \mathbb{N}$, $P_n \xrightarrow{\tau} P_{n+1}$.

$$divergences(P) = \{st \mid \exists Q.P \overset{s}{\Longrightarrow} Q \wedge Q\uparrow\}$$

Notice that we have said that st is a divergence trace whenever s is. This is motivated by a desire (inspired by our abstract semantics) to make all possibly divergent processes undefined. (As will be apparent from a careful reading of the proofs below, the fact that our semantic models and functions are strict with respect to divergence is sometimes of great importance.)

Say $P \in C$ is *stable* provided there is no Q such that $P \xrightarrow{\tau} Q$ (in other words, if P cannot make any internal progress). If $B \subseteq \Sigma$ we say P *ref* B if

$$\forall a \in B \cup \{\tau\}.\neg\exists Q \in C.P \xrightarrow{a} Q\,.$$

[4] A regular cardinal λ is one which is not the union of less than λ sets all of which are of size less than λ. There are arbitrarily large regular cardinals, since for example every *successor* cardinal is regular. The combinatorial properties which make *regular* cardinals the natural bounds for nondeterminism are well illustrated in [R3].

Thus P *ref* B implies that P is stable. We can now define

$$\mathit{failures}(P) = \{(s,B) \mid \exists Q.P \overset{s}{\Longrightarrow} Q \wedge Q \mathrel{\mathit{ref}} B\} \cup \{(s,B) \mid s \in \mathit{divergences}(P)\}\ .$$

The point of these definitions is that a process can properly refuse B only when it is in a stable state, for as long as it is performing internal actions one cannot be sure that it will not come into a state where a desired event is possible. On the other hand, when a process diverges it also refuses (in a different sense perhaps) all communications offered to it. The second part of the definition is also motivated by the desire to make a divergent process undefined.

If $u \in \Sigma^\omega$ is an infinite trace and $P \in C$, we write $P \overset{u}{\Longrightarrow}$ if there are $P = P_0, P_1, P_2, \ldots \in C$ and $x_i \in \Sigma^+$ such that $\forall k.P_k \xrightarrow{a_k} P_{k+1}$ and $\langle a_k \mid k \in \mathbf{N} \wedge a_k \neq \tau \rangle = u$. This lets us define

$$\mathit{infinites}(P) = \{u \in \Sigma^\omega \mid P \overset{u}{\Longrightarrow}\} \cup \{su \mid s \in \mathit{divergences}(P) \wedge u \in \Sigma^\omega\}\ .$$

Similarly, if $\langle x_i \mid i \in \omega \rangle = u \in (\Sigma^+)^\omega$ we can write $P \overset{u}{\longmapsto}$ if there exist $P = P_0, P_1, P_2, \ldots$ such that, for all i, $P_i \xrightarrow{x_i} P_{i+1}$.

Clearly it is possible to define other functions, and to vary these definitions for another definition of divergence. However the above are exactly the required maps to define the abstraction map into $\mathcal{U}$.

Definition. If C is any transition system then we define the abstraction map $\Phi : C \to \mathcal{U}$ as follows.

$$\Phi(P) = \langle \mathit{failures}(P),\ \mathit{divergences}(P), \mathit{infinites}(P) \rangle$$

We now state a theorem which establishes some basic properties of Φ.

Theorem 3.2. Φ is well defined, and furthermore

a) If $F : C \to D$ is a morphism then $\Phi(F(P)) = \Phi(P)$ for all $P \in C$.

b) If $P \in C$ and C is a sub-system of D (i.e., a subset closed under all the transition relations) then $\Phi(P)$ does not depend on whether we think of P as an element of C or of D.

c) Given any transition system C there is another one C' such that C is a subsystem of C' and $\Phi : C' \to \mathcal{U}$ is onto. ∎

It might seem a little curious that we have gone to the trouble of extending an arbitrary transition to one on which Φ is onto. The reason for this will become apperent when this result is used later.

In proving our congruence theorem later we will need not only the map $\Phi : C \to \mathcal{U}$ but also a sequence of approximations to it. We will define a map $\Phi_\alpha : C \to \mathcal{U}$ for each ordinal α. (Once again, C is here an arbitrary transition system.) It is convenient to define Φ_α in terms of a functional

$$\mathcal{G} : (C \to \mathcal{U}) \to (C \to \mathcal{U})\,.$$

If $\Psi : C \to \mathcal{U}$ and $P \in C$, we define $\mathcal{G}(\Psi)(P) = \langle F', D', I' \rangle$, where

$$\begin{aligned}
F' &= \{(\langle\rangle, X) \mid P \; ref \; X\} \\
&\ \cup\{(s,X) \mid \exists Q.P \xrightarrow{\tau} Q \wedge (s,X) \in \mathcal{F}\llbracket\Psi(Q)\rrbracket\} \\
&\ \cup\{(\langle a\rangle s,X) \mid \exists Q.P \xrightarrow{a} Q \wedge (s,X) \in \mathcal{F}\llbracket\Psi(Q)\rrbracket\} \\
D' &= \{s \mid \exists Q.P \xrightarrow{\tau} Q \wedge s \in \mathcal{D}\llbracket\Psi(Q)\rrbracket\} \\
&\ \cup\{\langle a\rangle s \mid \exists Q.P \xrightarrow{a} Q \wedge s \in \mathcal{D}\llbracket\Psi(Q)\rrbracket\} \\
I' &= \{u \mid \exists Q.P \xrightarrow{\tau} Q \wedge u \in \mathcal{I}\llbracket\Psi(Q)\rrbracket\} \\
&\ \cup\{\langle a\rangle u \mid \exists Q.P \xrightarrow{a} Q \wedge u \in \mathcal{I}\llbracket\Psi(Q)\rrbracket\}
\end{aligned}$$

The following Theorem establishes some useful properties of $\mathcal{G}$.

Theorem 3.3.

a) $\mathcal{G}$ is well defined and monotonic with respect to both orders.

b) Φ, as defined earlier in this section, is a fixed point of $\mathcal{G}$.

Proof. The whole of part (a) follows immediately from the fact that $\mathcal{G}$ can be re-written entirely in CSP. The operator $P \rhd Q$ used below is an abbreviation for $(P \Box Q) \sqcap Q$ (the process which can offer the choice between P and Q but which must eventually make an internal transition to become Q if no action occurs). It is a useful operator since it allows more conciseness, and has appeared before in similar circumstances in the literature, e.g. [HH].

$$\begin{aligned}
\mathcal{G}(\Psi)(P) &= x : P^0 \to \sqcap\{\Psi(Q) \mid P \xrightarrow{x} Q\} && \text{if } \nexists Q.P \xrightarrow{\tau} Q \\
\mathcal{G}(\Psi)(P) &= ((x : P^0 \to \sqcap\{\Psi(Q) \mid P \xrightarrow{x} Q\}) && \text{otherwise} \\
&\ \rhd \sqcap\{\Psi(Q) \mid P \xrightarrow{\tau} Q\}
\end{aligned}$$

where P^0 denotes $\{a \in \Sigma \mid \exists Q.P \xrightarrow{a} Q\}$. It is easy to see that our two definitions of $\mathcal{G}$ are equivalent. Note that the overall structure of this CSP definition depends only on the transitions within C, and is therefore independent of the value of Ψ. It is this last fact which proves that $\mathcal{G}$ is monotone with respect to both orders.

Part (b) is intuitively obvious. Consider, for example, the divergence component. It follows immediately from the definition of Φ that $\mathcal{D}\llbracket\Phi(P)\rrbracket = divergences(P)$ is equal to

$$\{st \mid P \xrightarrow{\tau} Q \wedge Q \overset{s}{\Longrightarrow} R \wedge R\uparrow\} \cup \{\langle a\rangle st \mid P \xrightarrow{a} Q \wedge Q \overset{s}{\Longrightarrow} R \wedge R\uparrow\}$$

which in turn is equal to

$$\{s \mid P \xrightarrow{\tau} Q \wedge s \in divergences(Q)\} \cup \{\langle a\rangle s \mid P \xrightarrow{a} Q \wedge s \in divergences(Q)\}$$

which is $\mathcal{D}\llbracket\mathcal{G}(\Phi)(P)\rrbracket$ by definition of $\mathcal{G}$. Both the other cases are similar and depend on this one. The failures case divides into three components rather than two for obvious reasons. ∎

By Theorem 1.5 applied to the product space $\mathcal{U}^C$ $(= C \to \mathcal{U})$, it follows from the existence of one fixed point that $\mathcal{G}$ has a least fixed point which is equal to Φ_α for some α where

$$\begin{array}{rcll} \Phi_0(P) & = & \bot & \text{for all } P \in C \\ \Phi_\mu(P) & = & \bigsqcup\{\Phi_\beta(P) \mid \beta \in \mu\} & \text{if } \mu \text{ is a limit ordinal} \\ \Phi_{\beta+1} & = & \mathcal{G}(\Phi_\beta) & \end{array}$$

since Φ_0 is the least element of the product space and $\Phi_\beta = \mathcal{G}^\beta(\Phi_0)$. (Since $\mathcal{G}$ is CSP-definable it also follows from the result of Section 2.) These Φ_β will play a crucial role in the main congruence theorem in the next section. This is essentially because of the next theorem, whose proof may be found in [R3].

Theorem 3.4. Φ is the least fixed point of $\mathcal{G}$. Hence there exists α such that $\Phi_\alpha = \Phi$.

∎

This result shows the equivalence of the natural operationally defined abstraction function and one which it obtained by iterating a CSP definition through the ordinals. This is exactly what we shall want to do on a much wider scale when we seek to prove the congruence theorem in the final section. It will turn out that this last result is perhaps the most important component of the proof of that theorem.

The operational semantics

This section is devoted to the definition of the operational semantics for CSP and closely related semantics over more general transition systems.

A crucial starting point for the creation of any semantics is the definition of the programming language. The definition we take is just the usual core CSP extended by unbounded nondeterminism and infinite hiding. For formal reasons we must fix *ab initio* the range of unbounded nondeterminism allowed. However this may be as large as we please. In particular, it is convenient to fix it strictly larger than the cardinality of the alphabet Σ. Thus the following language is implicitly parameterised both by the alphabet Σ of all possible communications and by the bound λ, an infinite regular cardinal on the unbounded nondeterminism.

Because the unbounded nondeterminism operator (unavoidably) and the guarded choice operator (avoidably at a price) are infinitary operators (take a potentially infinite number of process arguments) one should, for rigour, be rather careful over the definition of the syntax of this version of CSP. On the one hand we can write down the usual sort of BNF definition.

$$\begin{array}{rl} P ::= & p \mid \mathit{STOP} \mid \mathit{SKIP} \mid a \to P \mid x : B \to g(x) \mid P \Box Q \mid P \sqcap Q \mid \\ & P\ {}_B\|_C\ Q \mid P \mid\mid\mid Q \mid P;Q \mid P \backslash B \mid f[P] \mid f^{-1}[P] \mid \mu p.P \mid \bigsqcap S \end{array}$$

where g is any function from B (a subset of Σ) to processes, S ranges over nonempty sets of processes smaller than λ, f ranges over the set AT of (not necessarily finite-to-one) alphabet transformations, p over the set *Var* of process variables, etc.

When there are infinitary operators in a syntax, like those in this language, the idea of what is defined by a syntax like this one is less obvious than it usually is and should therefore be discussed briefly. If we are to have a principle of structural induction and have a way of defining the semantics of programs we cannot have a program of the form $\sqcap S$ or $x : B \rightarrow g(x)$ which is itself in S or in the range of g. One can, of course, regard BNF definitions like the above as fixed point equations, defining the smallest syntactic class which is closed under the various operations on the right. For a language with only finitary constructs this fixed point is reached by ω iterations (every program is "born on a finite day") but we have to go further, to cater for programs like $n : \mathbb{N} \rightarrow P_n$ where P_n is born on day n. The functional implied by the right hand side of the above BNF definition is clearly monotone (the more programs there are, the more it delivers) but since it is not operating over a set (rather over the proper Class of all syntactic objects) it is by no means obvious it even has a fixed point. Fortunately it does, and is guaranteed to reach it by λ iterations, where λ is the bound on nondeterminism and the size of Σ already mentioned. (See [BRW] for some more discussion of this question.) The principle of structural induction is then perfectly valid and corresponds to the principle of transfinite induction on the "birthday" of a term.

To simplify the operational semantics a little it is convenient, as was done in [BRW], to treat the constructs *STOP*, *SKIP* and $a \rightarrow P$ as special cases of the construct $x : B \rightarrow g(x)$: *STOP* has B empty, $a \rightarrow P$ has $B = \{a\}$ and $g(a) = P$, and $SKIP = \surd \rightarrow STOP$.

Let $\mathbf{E}$ be the set of all CSP terms defined by the above. An element of $\mathbf{E}$ may have free process variables, in which case it is said to be *open*. If it has none it is said to be *closed*; we denote the set of all closed terms by $\mathbf{P}$. Closed terms are of importance since their meaning is fully determined; there are no slots for processes waiting to be filled in.

If $P, Q \in \mathbf{E}$ and $p \in Var$ then $P[Q/p]$ denotes the term where Q has been substituted for all free occurrences of p is P. When Q is not closed (though for us it usually will be) some care will be necessary to prevent P binding any of Q's free variables.

The Plotkin-style semantics regards the set $\mathbf{P}$ of all closed CSP-terms as a transition system, since it describes the set of all actions each closed term can perform and which new terms it may then become. The clauses of this operational semantics are given in the usual "natural deduction" style below.

Below, a, b range over Σ and x, y over $\Sigma^+ = \Sigma \cup \{\tau\}$. Alphabet transformations (functions from Σ to Σ) are extended to Σ^+ by setting $f(\tau) = \tau$.

$$\frac{}{(x : B \rightarrow g(x)) \xrightarrow{b} g(b)} \quad (b \in B)$$

$$\frac{}{P \sqcap Q \xrightarrow{\tau} P} \qquad \frac{}{P \sqcap Q \xrightarrow{\tau} Q}$$

$$\frac{}{\mu p.P \xrightarrow{\tau} P[\mu p.P/p]}$$

$$\frac{P \xrightarrow{\tau} P'}{P \Box Q \xrightarrow{\tau} P' \Box Q} \qquad \frac{Q \xrightarrow{\tau} Q'}{P \Box Q \xrightarrow{\tau} P \Box Q'}$$

$$\frac{P \xrightarrow{a} P'}{P \Box Q \xrightarrow{a} P'} \qquad \frac{Q \xrightarrow{a} Q'}{P \Box Q \xrightarrow{a} Q'}$$

$$\frac{P \xrightarrow{\tau} P'}{P \,{}_B\|_C\, Q \xrightarrow{\tau} P' \,{}_B\|_C\, Q} \qquad \frac{Q \xrightarrow{\tau} Q'}{P \,{}_B\|_C\, Q \xrightarrow{\tau} P \,{}_B\|_C\, Q'}$$

$$\frac{P \xrightarrow{a} P'}{P \,{}_B\|_C\, Q \xrightarrow{a} P' \,{}_B\|_C\, Q} \quad (a \in B - C)$$

$$\frac{Q \xrightarrow{a} Q'}{P \,{}_B\|_C\, Q \xrightarrow{a} P \,{}_B\|_C\, Q'} \quad (a \in C - B)$$

$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q'}{P \,{}_B\|_C\, Q \xrightarrow{a} P' \,{}_B\|_C\, Q'} \quad (a \in B \cap C)$$

$$\frac{P \xrightarrow{x} P'}{P \,|||\, Q \xrightarrow{x} P' \,|||\, Q} \qquad \frac{Q \xrightarrow{x} Q'}{P \,|||\, Q \xrightarrow{x} P \,|||\, Q'}$$

$$\frac{P \xrightarrow{x} P'}{P; Q \xrightarrow{x} P'; Q} \quad (x \neq \surd)$$

$$\frac{\exists P'.P \xrightarrow{\surd} P'}{P; Q \xrightarrow{\tau} Q}$$

$$\frac{P \xrightarrow{x} P'}{P \backslash B \xrightarrow{x} P' \backslash B} \quad (x \notin B)$$

$$\frac{P \xrightarrow{a} P'}{P \backslash B \xrightarrow{\tau} P' \backslash B} \quad (a \in B)$$

$$\frac{P \xrightarrow{x} P'}{f[P] \xrightarrow{y} f[P']} \quad (y = f(x))$$

$$\frac{P \xrightarrow{x} P'}{f^{-1}[P] \xrightarrow{y} f^{-1}[P']} \quad (f(y) = x)$$

$$\frac{P \in S}{\sqcap S \xrightarrow{\tau} P}$$

Note at this point that the operationally natural element of $\mathcal{U}$ corresponding to each closed term P is given by $\Phi(P)$, where Φ is as defined in Section 2 and P is considered to be an element of the transition system **P** defined above. Theorem 3.2 shows that this is equal to $\Phi(F(P))$ for any morphism F. We can now state the main congruence result that we would like to prove, namely that for all closed CSP terms P, $\Phi(P) = \mathcal{S}[\![P]\!]$, where $\mathcal{S}[\![P]\!]$ denotes the value in $\mathcal{U}$ defined by the semantics defined earlier.

There are two structure clashes between the operational and denotational semantics. The first is the obvious one that one is given in terms of transition systems and the other in terms of the abstract model $\mathcal{U}$. But perhaps the more difficult one to resolve is the clash between the term rewriting style of the operational semantics and the denotational style of the other. Of course the latter means that the semantic value of each term is deduced from the semantic value of its subcomponents in a transparent way and that an abstract fixed point theory is used. In the earlier paper on the operational semantics of CSP [BRW] these two issues were resolved separately by creating an intermediate, denotational tree semantics. Unfortunately the complete metric spaces of trees used in that paper no longer exist because of the introduction here of infinite branching.

The main result of [R3] is that, for each infinite regular cardinal λ, there exists a transition system T_λ such that for all transition systems C with $i(C) \leq \lambda$, there exists a unique morphism $H_\lambda : C \to T_\lambda$. Thus T_λ is a final object in the category of transition systems with morphisms as arrows. Analogues of the contraction mapping theorem and related results hold which are useful when one uses these systems. T_λ can be used to give an intermediate denotational semantics to CSP in the style of [BRW]. However, because of the complexity of this new theory and thanks mainly to the construction of the Φ_α in the previous section we do not now need to do so.

It is useful to extend the operational space defined above to include non-closed terms with their variables instantiated by elements of an arbitrary transition system.

Definition. If C is any transition system then C^{CSP} is the system of CSP syntactic terms over C: namely the set of all substitutions by elements of C for all free variables of general terms in the language. All terms are distinct. Note that C^{CSP} contains every closed CSP term and every element of C. The transitions of each term are those of P if $P \in C$ (i.e., $P \xrightarrow{\delta} Q$ in C^{CSP} if and only if $P \xrightarrow{\delta} Q$ in C). The transitions of proper syntactic terms are determined from the operational semantic clauses above (from those of their subterms or otherwise).

The stipulation that all terms are distinct means that each possible construction of a term leads to a different element of the system. For example, in $(C^{CSP})^{CSP}$, for each $P \in C$ the terms $a \to a \to \ulcorner P \urcorner$, $a \to \ulcorner a \to P \urcorner$ and $\ulcorner a \to a \to P \urcorner$ are all different, where the syntactic quotes $\ulcorner \cdot \urcorner$ denote the boundary between the inner and outer syntactic construction. However the obvious map from $(C^{CSP})^{CSP}$ to C^{CSP} which "forgets" these boundaries is easily shown to be a morphism.

Note that Theorem 3.2 (a) tells us that the image under Φ of a closed term P is independent of whether it is considered to belong to the space $\mathbf{P}$ of closed terms or any C^{CSP}, since there is an obvious morphism embedding $\mathbf{P}$ into any C^{CSP}.

We are now in a position to begin the proof of the main theorem of this section, namely that the $\mathcal{U}$ semantics for CSP is congruent to the operational semantics. We will eventually complete the proof by performing a structural induction over C^{CSP}, but before we do that it is helpful to establish that the operational and denotational versions of all the non-recursive operators are congruent.

Theorem 3.5. The operational versions of the various CSP operators are all congruent to the denotational versions over $\mathcal{U}$. In other words, for each operator $\odot$ and each $P, Q \in C^{CSP}$,

$$\Phi(P \odot Q) = \Phi(P) \odot \Phi(Q)\,.$$

Furthermore all the operators are well behaved with respect to the partial abstraction functions Φ_α in the sense that

$$\Phi_\alpha(P \odot Q) \leq \Phi_\alpha(P) \odot \Phi_\alpha(Q)$$

for each α. (The form of these clauses is modified suitably when the operator $\odot$ is not binary. The precise statement for each operator in turn can be found in the Lemmas below.)

Proof. This theorem is no more nor less than a convenient grouping of a large number of similar though separate results. These are stated below, grouped by operator, plus for each operator a further result which is crucial in the proof of the full congruence part of the Lemma. In each of these Lemmas it is assumed that the given term is an element of C^{CSP} of the given form; the immediate subterms being unrestricted elements of C^{CSP} (i.e., not necessarily elements of C itself).

The operators break into two classes as far as style of proof is concerned: prefixing and nondeterministic choice, which are easiest, and the rest of the operators, which require very similar though more difficult arguments – these proofs are omitted here but may be found in [R3]. As usual, recursion is a special case and will be dealt with on its own later.

Lemma 3.5.1 (i). For all terms P_x denoting functions from A into C^{CSP}, we have

$$\Phi(x : A \to P_x) = x : A \to \Phi(P_x)\,.$$

Lemma 3.5.1 (ii). For all terms P_x denoting functions from A to C^{CSP} and all ordinals α we have

$$\Phi_\alpha(x : A \to P_x) \;\leq\; x : A \to \Phi_\alpha(P_x)\,.$$

Lemma 3.5.2 (i). For all $S \subseteq C^{CSP}$ (of size less than our bound on nondeterminism) we have

$$\Phi(\bigsqcap S) = \bigsqcap\{\Phi(P) \mid P \in S\}\,.$$

Lemma 3.5.2 (ii). For all P, Q in C^{CSP} and all ordinals α we have

$$\Phi_\alpha(\bigsqcap S) \leq \bigsqcap\{\Phi_\alpha(P) \mid P \in S\}\,.$$

Lemma 3.5.3 (i) If $P, Q \in C^{CSP}$ and $\odot$ is any one of $\Box$, ;, ${}_X\|_Y$, $|||$, $\sqcap$ then $\Phi(P \odot Q) = \Phi(P) \odot \Phi(Q)$.

Lemma 3.5.3 (ii) If $P, Q \in C^{CSP}$, $\odot$ is any one of $\Box$, ;, ${}_X\|_Y$, $|||$, $\sqcap$ and α is any ordinal then $\Phi_\alpha(P \odot Q) \leq \Phi_\alpha(P) \odot \Phi_\alpha(Q)$.

Lemma 3.5.4 (i). If $P \in C^{CSP}$ and $\ddagger$ is any one of $f[\cdot]$, $f^{-1}[\cdot]$ and $\backslash X$ then $\Phi(\ddagger(P)) = \ddagger(\Phi(P))$.

Lemma 3.5.4 (ii). If $P \in C^{CSP}$,$\ddagger$ is any one of $f[\cdot]$, $f^{-1}[\cdot]$ and $\backslash X$ and α is any ordinal then

$$\Phi_\alpha(\ddagger(P)) \leq \ddagger(\Phi_\alpha(P))\,.$$

This completes Theorem 3.5. ∎

These results provide the building blocks of the proof of the main result, and are put together below. The next Theorem is the main result of this section.

Definitions. Given a CSP term P and a $\rho \in OEnv = Var \longrightarrow C^{CSP}$, we can define an operational "semantic function": $\mathcal{O}[\![P]\!]\rho \in C^{CSP}$ is defined to be the result of substituting each free variable p in P by $\rho(p)$. (Note that P may have no free variables, finitely many, or infinitely many. This last possibility arises because of the two infinitary operations $\sqcap$ and $x : A \longrightarrow P_x$.) Given $\rho \in OEnv$ we can define the corresponding element $\overline{\rho}$ of $UEnv = Var \longrightarrow \mathcal{U}$ by

$$\overline{\rho}[\![p]\!] = \Phi(\rho(p))$$

and also, for each α, an approximation

$$\overline{\rho}^{\alpha}[\![p]\!] = \Phi_{\alpha}(\rho(p))\,.$$

In this theorem we will assume that the basic transition system C is such that $\Phi : C \longrightarrow \mathcal{U}$ is onto (following Theorem 3.2 (c)). This is helpful in the proof, since it means that for each $\sigma \in UEnv$ there is a $\rho \in OEnv$ such that $\overline{\rho} = \sigma$.

Theorem 3.6. Suppose P is any CSP term. Then the following hold.

a) $\mathcal{S}[\![P]\!]\overline{\rho} = \Phi(\mathcal{O}[\![P]\!]\rho)$ for all $\rho \in OEnv$.

b) For each ordinal α and each $\rho \in OEnv$ we have $\mathcal{S}[\![P]\!]\overline{\rho}^{\alpha} \geq \Phi_{\alpha}(\mathcal{O}[\![P]\!]\rho)$.

Proof. This is by structural induction on P. Given the sequence of Lemmas above, the cases of all the non-recursive operators are trivial, parts (a) and (b) respectively following from the (i) and (ii) of the Lemmas under Theorem 3.5 above.

It only remains to consider the case of a recursively defined term $\mu p.P$, where the result is known to hold of P.

For part (a), observe that $\mathcal{S}[\![\mu p.P]\!]\overline{\rho}$ is defined to be the *least* fixed point of $F(Y) = \mathcal{S}[\![P]\!]\overline{\rho}[Y/p]$. Now set $X = \Phi(\mathcal{O}[\![\mu p.P]\!]\rho)$ and note that since the only transition of $\mathcal{O}[\![\mu p.P]\!]\rho$ is a τ-transition to $\mathcal{O}[\![P[\mu p.P/p]]\!]\rho$, we have

$$\begin{array}{rcll} X & = & \Phi(\mathcal{O}[\![P[\mu p.P/p]]\!]\rho) & \\ & = & \Phi(\mathcal{O}[\![P]\!]\rho[\mathcal{O}[\![\mu p.P]\!]\rho/p]) & \\ & = & \mathcal{S}[\![P]\!]\overline{\rho[\mathcal{O}[\![\mu p.P]\!]\rho/p]} & \text{by induction} \\ & = & \mathcal{S}[\![P]\!]\overline{\rho}[X/p] & \\ & = & F(X) & \end{array}$$

and so X is a fixed point of F and so is certainly greater than $\mathcal{S}[\![\mu p.P]\!]\overline{\rho}$.

For the reverse inequality we will prove by induction on α that $\Phi_{\alpha}(\mathcal{O}[\![\mu p.P]\!]\rho) \leq F^{\alpha}(\bot)$ for all α. This is enough since we know that for sufficiently large α the left hand side equals $\Phi(\mathcal{O}[\![\mu p.P]\!]\rho)$ and the right hand side is $\mathcal{S}[\![\mu p.P]\!]\overline{\rho}$. The cases of $\alpha = 0$ and α a limit ordinal are both trivial, the latter because both sides are defined to be the least upper bounds of the terms for smaller ordinals. So suppose it holds for β and $\alpha = \beta + 1$. Then $\Phi_{\alpha}(\mathcal{O}[\![\mu p.P]\!]\rho) = \Phi_{\beta}(\mathcal{O}[\![P[\mu p.P/p]]\!]\rho)$ because of the initial τ-transition and the definition of $\mathcal{G}$. But this by induction and monotonicity is weaker than $F^{\alpha}(\bot)$, as required.

It only remains to prove (c), in other words that, given ρ and α,

$$\Phi_{\alpha}(\mathcal{O}\llbracket \mu p.P\rrbracket\rho) \leq \mathcal{S}\llbracket \mu p.P\rrbracket\overline{\rho}^{\alpha}\,.$$

Once again we prove this by transfinite induction on α. Again the result is easy for $\alpha = 0$ since the left hand side is $\bot$ and also for the limit ordinal case since the left hand side at α is then the least upper bound of the previous left hand sides, and $\mathcal{S}\llbracket P\rrbracket$ is monotone. So suppose $\alpha = \beta + 1$ and that the result holds at β. Then

$$\begin{array}{rcll}
\Phi_{\beta+1}(\mathcal{O}\llbracket \mu p.P\rrbracket\rho) & = & \Phi_{\beta}(\mathcal{O}\llbracket P[\mu p.P/p]\rrbracket\rho) & \\
 & = & \Phi_{\beta}(\mathcal{O}\llbracket P\rrbracket\rho[\mathcal{O}\llbracket \mu p.P\rrbracket\rho/p]) & \\
 & \leq & \mathcal{S}\llbracket P\rrbracket\overline{\rho[\mathcal{O}\llbracket \mu p.P\rrbracket\rho/p]}^{\beta} & \text{by (c) of } P \\
 & \leq & \mathcal{S}\llbracket P\rrbracket\overline{\rho}^{\beta}[\Phi_{\beta}(\mathcal{O}\llbracket \mu p.P\rrbracket\rho)/p] & \\
 & \leq & \mathcal{S}\llbracket P\rrbracket\overline{\rho}^{\beta}[\mathcal{S}\llbracket \mu p.P\rrbracket\overline{\rho}^{\beta}/p] & \text{induction and monotonicity} \\
 & = & \mathcal{S}\llbracket \mu p.P\rrbracket\overline{\rho}^{\beta} & \text{as recursions denote fixed points} \\
 & \leq & \mathcal{S}\llbracket \mu p.P\rrbracket\overline{\rho}^{\beta+1} & \text{by monotonicity}
\end{array}$$

which proves it for $\beta + 1$. This completes the proof of Theorem 3.5. ▮

The reader may have noticed that this section has not discussed the operational semantics of processes defined by mutual recursion, whether finite or infinite. This was because the proof for single recursion was quite difficult enough, and that for the more general case adds little except complexity. Also, as is pointed out in [R3], there is a simple CSP transformation which allows one to re-cast any mutual recursion as a single recursion. It would certainly be possible to base a proof of the operational validity of mutual recursion on that.

4. Conclusions

We have seen how to construct the infinite traces model $\mathcal{U}$, how it has unusual partial order properties, and how to overcome the incompleteness of the underlying orders. We have also seen the main points of the proof that our denotational semantics are congruent to the natural operational semantics.

The two orders $\leq$ and $\sqsubseteq$ have both been used throughout the paper in various ways. This leads to the same question as was posed in [R2], namely that of which is the natural order to use when presenting the model and semantics, given that both work. Here the arguments are slightly different. On the one hand now neither order is complete (whereas only $\leq$ was over $\mathcal{N}'$). However $\leq$ does still have a nicer theory of least upper bounds than $\sqsubseteq$, for they are always given by intersection where they exist while this is not even true for directed sets for $\sqsubseteq$. On the other, $\sqsubseteq$ is simpler to define and is perhaps more intuitive, but it does not have such a claim over $\mathcal{U}$ to be the "established" order as over $\mathcal{N}$ or $\mathcal{N}'$. And also $\sqsubseteq$ played an important part in the proof of the existence of fixed points seen in Section 2. This question will be best resolved by time and experience.

On the technical side we have seen in this work that completeness and continuity are natural casualties of the introduction of unbounded nondeterminism, but that their absence does not matter unduly except in the sense that proofs become more difficult. It will be interesting to see whether similar problems arise in other formalisms.

We have also sketched the proof that our semantics is operationally valid. Perhaps the most interesting feature of the proof is the way the approximate abstraction functions Φ_α show that the least fixed point corresponds with the operationally natural one via a type of "non-destructiveness" argument.

Future work on this model will include a fuller investigation of its algebraic properties. Another issue will be the study of other unboundedly nondeterministic constructs such as fair hiding operators. We should note that it is only permissible to add a new operator (other than one derived from existing operators) to this version of CSP if its restriction to $\mathcal{P}$ can is dominated by an operator which preserves predeterminism, to allow the proof in Section 2 to carry through. It will also be interesting to see what use can be made of the infinite traces component in the *specifications* of processes. For example one could add a clause to the usual specification of a buffer which stated that the buffer never does infinitely many inputs without an output, so that anything one puts in is eventually going to come out (even in the presence of an environment which eagerly places as much as possible into the buffer at all times).

The difficulties one encounters when dealing with unbounded nondeterminism, particularly the sort which is only detectable from infinite behaviours, are certainly not restricted to the models seen in this paper. Hopefully some of the work reported here will transfer to other formalisms for concurrency. One place where valuable work could be done is in timed CSP (see [RR1, RR2, Re]). The incorporation of infinite behaviours there (were it possible) would allow more abstract and general expressions of such modalities as "eventually" which appear in some forms of temporal logic.

Acknowledgements

Stephen Blamey has put a lot of work into analysing, refining and understanding the axioms for $\mathcal{U}$. In addition this work has been assisted by conversations with a number of colleagues, notably Paul Gardiner, Michael Goldsmith, Alan Jeffrey and David Walker.

References

[B] Brookes, S.D., *A Model for Communicating Sequential Processes*, Oxford University D.Phil. thesis, 1983.

[Blam] Blamey, S.R., *The soundness and completeness of axioms for CSP processes*, submitted for publication.

[BHR] Brookes, S.D., Hoare, C.A.R., and Roscoe, A.W., *A theory of communicating sequential processes*, JACM Vol. 31, No. 3 (July 1984) 560-599.

[BR] Brookes, S.D. and Roscoe, A.W., *An improved failures model for communicating processes* in Proc. of Pittsburgh symposium on concurrency, Springer LNCS !97 (1985).

[BRW] Brookes, S.D., Roscoe A.W., and Walker, D.J., *An operational semantics for CSP*, Submitted for publication.

[H] Hoare, C.A.R., *Communicating sequential processes*, Prentice-Hall, 1985

[HH] He Jifeng and Hoare, C.A.R., *Algebraic specification and proof of properties of communicating sequential processes*, Technical monograph PRG-52, Oxford University Computing Laboratory.

[R1] Roscoe, A.W., *A mathematical theory of communicating processes*, Oxford University D.Phil. thesis, 1982.

[R2] Roscoe, A.W., *An alternative order for the failures model*, in 'Two Papers on CSP', Technical monograph PRG-67, Oxford University Computing Laboratory.

[R3] Roscoe, A.W., *Unbounded nondeterminism in CSP*, in 'Two Papers on CSP', Technical monograph PRG-67, Oxford University Computing Laboratory.

[R4] Roscoe, A.W., *Analysing infinitely branching trees*, in preparation.

[Re] Reed, G.M., *A uniform mathematical theory for real-time distributed computing*, Oxford University D.Phil. thesis, 1988.

[RR1] Reed, G.M., and Roscoe, A.W., *A timed model for communicating sequential processes*, Proceedings of ICALP'86, Springer LNCS 226 (1986), 314-323.

[RR2] Reed, G.M., and Roscoe, A.W., *Metric spaces as models for real-time concurrency*, in the proceedings of MFPLS87 Springer LNCS 298 (1988).