New Foundations for the Geometry of Interaction*

Samson Abramsky

Dept. of Computing
Imperial College
London

Radha Jagadeesan

Dept. of Computing
Imperial College
London.

December 10, 1992

*This research was supported by grants from UK SERC and ESPRIT Basic Research Action 3003 "CLICS"

Contents

1 Introduction

A basic dichotomy runs through much of programming theory, assuming a number of guises: denotational/operational, declarative/procedural, logical/computational. Mathematical structure is seen as residing chiefly in the first term of each dichotomy, computational dynamics and its associated intuitions in the second. A number of recent developments, perhaps most notably Girard's Linear Logic [Gir87] and "Geometry of Interaction" [Gir89b], aim to find a new middle ground between these dichotomies. The Geometry of Interaction programme is to give a semantics of computation, specifically of Cut–elimination in Linear Logic, with the following key features:

- The semantics is syntax–free and uses denotational tools, yet it captures the essential features of the computational dynamics.
- The process of Cut–elimination is modelled by the flow of information tokens around a network, rather than by graph–rewriting.
- There is a *normal form* analogous to the Kleene normal form in recursion theory: the entire process of cut–elimination is described by the iterations of a single operator.

Girard has implemented this programme in a sequence of papers [Gir89b, Gir89a, Gir88a], using the formalism of $C^\star$–algebras. While the ideas are highly original and striking, and the technical execution must be considered a *tour de force*, some desiderata remain.

1. Can one give a more systematic account, making clear what structure is really needed to carry out the interpretation, and showing how it arises more or less inevitably from some simple, basic ideas?
2. The interpretation does not succeed in interpreting the whole of Linear Logic, and the main result establishing the soundness of the interpretation is subject to certain restrictions; the interpretation is actually *unsound* in general.
3. The connection with any concrete implementation is left unclear.

In this paper, we present a new formal embodiment of Girard's programme, with the following salient features.

1. Our formalisation is based on elementary Domain Theory rather than $C^\star$–algebras. It exposes precisely what structure is required of the ambient category in order to carry out the interpretation. Furthermore, we show how the interpretation arises from the construction of a categorical model of Linear Logic; this provides the basis for a rational reconstruction which makes the structure of the interpretation much easier to understand.

2. The key definitions in our interpretation differ from Girard's. Most notably, we replace the "execution formula" by a least fixpoint, essentially a generalisation of Kahn's semantics for feedback in dataflow networks [Kah77, KM77]. This, coupled with the use of the other distinctive construct of Domain theory, the lifting monad, enables us to interpret the whole of Linear Logic, and to prove soundness in full generality.

3. Our general notion of interpretation has simple examples, providing a suitable basis for concrete implementations. In fact, we sketch a computational interpretation of the Geometry of Interaction in terms of *dataflow networks*. Recall that computation in dataflow networks is asynchronous, *i.e.* "no global time", and proceeds by purely local "firing rules" that manipulate tokens.

The further structure of this paper is as follows. In Section 2, we review the syntax of Linear Logic, and present the basic, and quite simple intuitions underlying the interpretation. In Section 3, we use these ideas to construct models of Linear Logic. In Section 4 we define the Geometry of Interaction interpretations, and how that they arise from the model constructed previously in a natural fashion. In Section 5, we give a computational interpretation of these model in terms of dataflow. The soundness of the interpretation, including the semantic analogue of Cut–elimination, is proved in Section 6. In Section 7, a Characterisation theorem is proved giving a more explicit description of the interpretation. This provides the basis both for a stronger soundness result and for establishing a connection with a denotational semantics based on coherence spaces.

2 Basic Intuitions

For general background on Linear Logic, we refer the reader to the original paper by Girard [Gir87]. Here, we shall briefly recall the syntax of $\mathtt{CLL}_2$, second order propositional Classical Linear Logic.

The formulae of $\mathtt{CLL}_2$ are in *negation normal form*; they are built from propositional variables α and their linear negations $\alpha^\perp$ by the following connectives:

Multiplicatives	$A \otimes B$	$A \parr B$
Multiplicative Units	I	$\perp$
Additives	$A \& B$	$A \oplus B$
Additive Units	$\top$	0
Exponentials	$!A$	$?A$
Quantifiers	$\forall\alpha.A$	$\exists\alpha.A$

Linear negation is extended to general formulae as an operation defined by the following equations.

$$(A \otimes B)^\perp = A^\perp \parr B^\perp$$

$$
\begin{array}{rcl}
I^{\perp} & = & \perp \\
(A \& B)^{\perp} & = & A^{\perp} \oplus B^{\perp} \\
(!A)^{\perp} & = & ?A^{\perp} \\
(\forall \alpha . A)^{\perp} & = & \exists \alpha . A^{\perp} \\
A^{\perp\perp} & = & A
\end{array}
$$

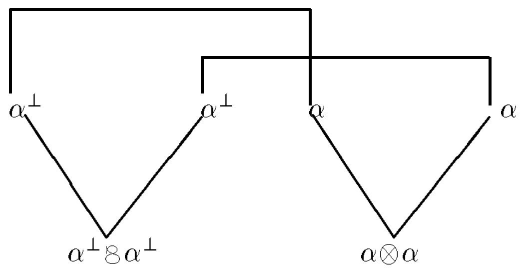
with linear implication treated as a derived connective defined by

$$A \multimap B = A^{\perp} \parr B$$

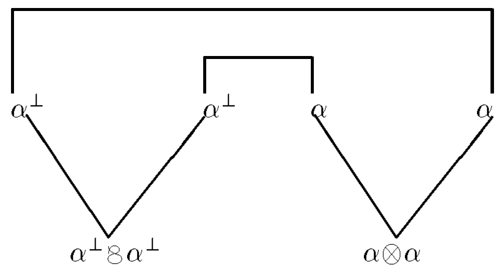
The objects to be derived in $\mathtt{CLL}_2$ are right-sided sequents $\vdash \Gamma$, where Γ is a list of formulas.

The original version of the Geometry of Interaction was developed by Girard for the multiplicative fragment [Gir88b]. This is still the best setting in which to explain the basic ideas on which the interpretation is based.

Consider then the multiplicative fragment of $\mathtt{CLL}_2$, with the restriction that the Axiom is only used for propositional atoms, $\vdash \alpha^{\perp}, \alpha$. Now, if we look at the cut-free proofs, say of $\alpha \otimes \alpha \multimap \alpha \otimes \alpha$, *i.e.* of the sequent $\vdash \alpha^{\perp} \parr \alpha^{\perp}, \alpha \otimes \alpha$, there are in fact just two, corresponding to the identity and twist maps.

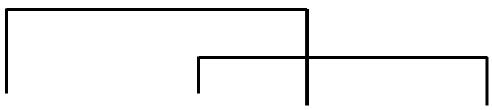


Id

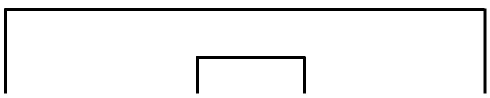


Twist

As we see from these examples, cut-free proof nets in this fragment have the structure of a set of trees, one for each formula in the conclusion, with the leaves connected up in pairs by the axiom links. Moreover, the structure of the trees is determined uniquely by the formulae in the sequent (this is where the restriction on axioms is applied). Hence, a complete invariant to distinguish the different cut-free proofs of a given sequent is given by the information as to how the leaves are joined up.

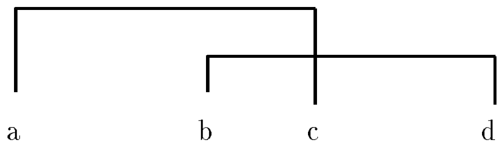


Id



Twist

We can model this information by a permutation on the set of leaves, obtained as the product of the transpositions corresponding to the axiom links. Thus,

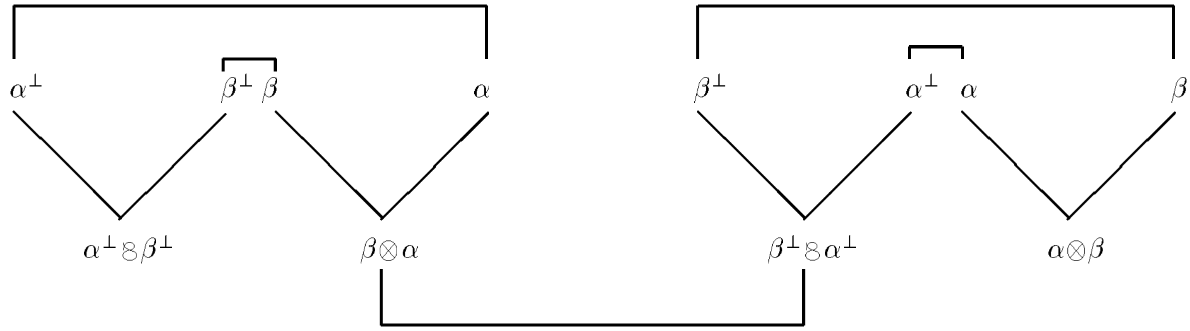


corresponds to the permutation

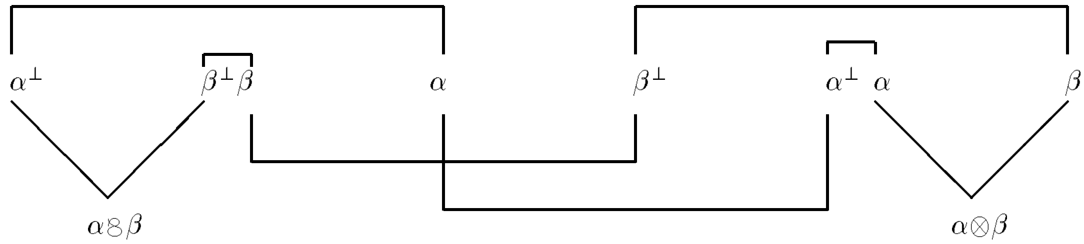
$$\begin{pmatrix} a & b & c & d \\ c & d & a & b \end{pmatrix}$$

Note that these transpositions are *disjoint*. So, a cut-free proof is represented by an *involution*, *i.e.* a self-inverse permutation. This representation of cut-free proofs can be thought of as modelling the "information flow" between the leaves in a dynamic fashion—think of tokens travelling in both directions across the axiom links—as opposed to modelling the linkage statically by a graph. Note that we are using functions to represent this information flow, but without input-output bias, since the flow is bidirectional and symmetric.

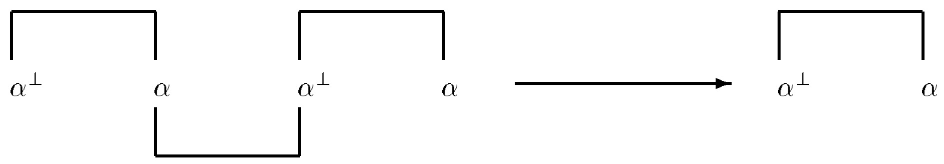
Returning to our example, consider performing cut-elimination on `twist` $\circ$ `twist`: The proof net for `twist` $\circ$ `twist` before cut elimination is:



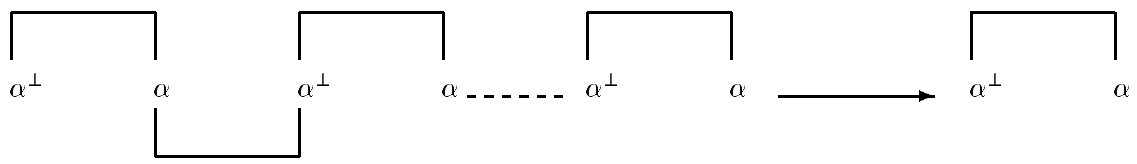
The proof net for `twist` $\circ$ `twist` after one step of reduction is:



Generally, in this fragment, we can apply this "decomposition rule" repeatedly for tensor cut against par until all cuts are between axiom links. We can say that the whole purpose of these transformations is to match up the corresponding axiom links correctly; the "real" information flow is then accomplished by the axiom reductions:



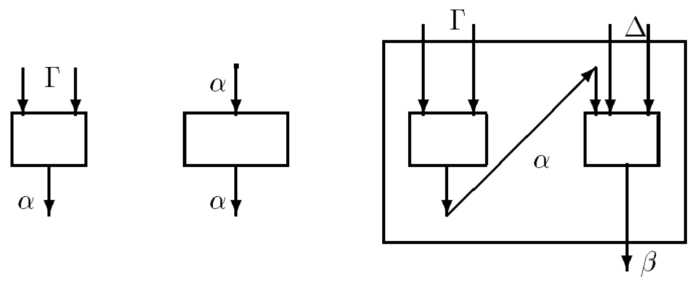
or more generally,



The idea, as with cut-free proofs, is to model these transformations dynamically, by the flow of information tokens, rather than by graph rewriting.

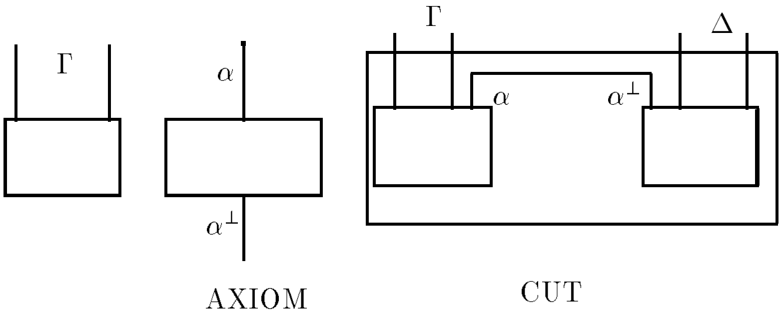
An interpretation of the multiplicative fragment can be given using just permutations on finite sets, as described in [Gir88b]. Extending this to the whole of Linear Logic is much harder and requires more sophisticated tools. In particular, the shape of the trees, and hence how the leaves are to be matched up, is no longer determined by the types, and must instead be computed dynamically; the different shapes that can arise correspond to the different computation paths of the program. An important consequence of this is that the objects corresponding to proofs will have to be "partial permutations", which both pick out the subdomains of trees corresponding to the possible computation paths, and for each such tree induce a permutation on the leaves corresponding to the flow of information along that path.

We conclude this section with some remarks on the use of functions in the interpretation. For intuitionistic sequents $\Gamma \vdash A$, whether in Intuitionistic or Intuitionistic Linear Logic, we have the usual functional interpretation.

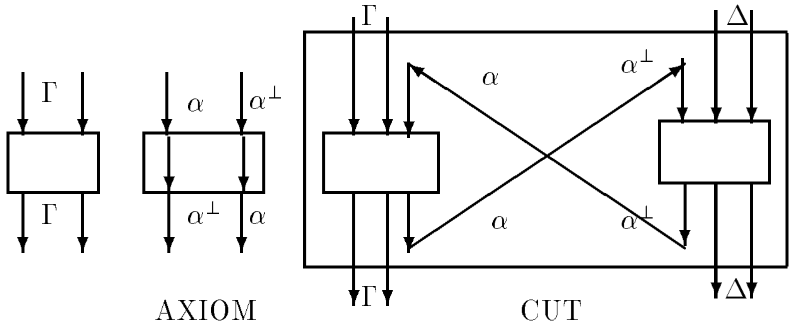


AXIOM CUT

For sequents in Classical Linear Logic, the first author has proposed a process interpretation [Abr91b], where Γ is an *interface specification* describing how the process P can be connected to its environment; the links are no longer directed since they correspond to a symmetric notion of communication. Axiom is interpreted by a communication buffer; and Cut by "communicating parallel composition + restriction".



In the Geometry of Interaction interpretation, we shall reconcile the functional and process views. The computational object corresponding to a sequent Γ will be a function for which every formula in Γ appears *twice*: once as (the type of) an input, once as (the type of) an output. The idea is that this function models the (bidirectional, symmetric) flow of information within the proof of the sequent.



Note that Cut involves feedback; this is precisely where the computational dynamics is located. We model this, following Kahn's semantics for dataflow [Kah77, KM77], with a least fixpoint. If $f : D^n \times D \to D^n \times D$ and $g : D \times D^m \to D \times D^m$, where D is a domain and f and g are continuous functions, we can define $\mathtt{Cut}(f,g) : D^n \times D^m \to D^n \times D^m$ as follows: $\mathtt{Cut}(f,g)\,(\vec{x},\vec{y})$ is the projection onto $\vec{x}',\vec{y}'$ of the least solution of the equations:

$$\begin{array}{rcl} \langle \vec{x}', x \rangle & = & f\langle \vec{x}, y \rangle \\ \langle y, \vec{y}' \rangle & = & g\langle x, \vec{y} \rangle \end{array}$$

The usual sequence of iterations to the least fixpoint is the representation of the dynamics provided by our interpretation. It takes the place of Girard's execution formula.

3 Geometry of Interaction Models

We shall assume some familiarity with elementary Domain theory and Category theory.

3.1 Preliminaries

Idempotents

Let $\mathbb{C}$ be a category. An idempotent in $\mathbb{C}$ is an endomorphism $e : A \to A$ with $e^2 = e$. Idempotents have been used in semantics [Sco76] to specify datatypes as "subdomains" of some "universal domain". We recall the construction of $\mathtt{Split}(\mathbb{C})$ [FS91] [also known as the "Karoubi envelope"], which formally splits the idempotents in $\mathbb{C}$ into retraction-coretraction pairs. The objects of $\mathtt{Split}(\mathbb{C})$ are pairs (A, e), where $e : A \to A$ is an idempotent in $\mathbb{C}$. A morphism is a triple $(e, f, e') : (A, e) \to (A', e')$, where $f : A \to A'$ satisfies $e' \circ f = f = f \circ e$. Composition and identities are then given by $(e'', g, e') \circ (e, f, e') = (e'', g \circ f, e)$ and $1_{(A,e)} = (e, 1_A, e)$.

An automorphism in $\mathtt{Split}(\mathbb{C})$ is then given by $(e, f, e) : (A, e) \to (A, e)$ with inverse $(e, g, e) : (A, e) \to (A, e)$ satisfying

$$\begin{aligned} g \circ f &= e = f \circ g \\ g \circ f \circ g &= g \\ f \circ g \circ f &= f \end{aligned}$$

In particular, the first equation tells us that e is recoverable from f, g.

On the intuition that a partial automorphism on an object A of $\mathbb{C}$ should be an automorphism acting on a subdomain of A specified by an idempotent, we define a *partial automorphism* f, with inverse g, to be morphisms $f, g : A \to A$ satisfying

$$\begin{aligned} g \circ f &= f \circ g \\ g \circ f \circ g &= g \\ f \circ g \circ f &= f \end{aligned}$$

The reader will enjoy proving that g is uniquely determined by f. A *partial involution* is then a self-inverse partial automorphism. Taking $f = g$ in the definition above, this reduces to the single equation

$$f^3 = f$$

We note a simple way of constructing (partial) involutions. Suppose $f : A \to B$ is an isomorphism, then $\hat{f} = A \times B \xrightarrow{f \times f^{-1}} B \times A \xrightarrow{\mathtt{symm}} A \times B$ is a partial involution.

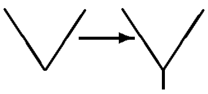
Domain theory

We review some basic definitions that we shall use. A poset P is *ω-complete* if every ω-chain in P has a least upper bound in P. A function between ω-complete posets is continuous if it preserves least upper bounds of ω-chains. We write **Predom** for the category of "predomains", *i.e.* ω-complete posets and continuous maps, and **Dom** for the full subcategory of *domains*, *i.e.* ω-complete posets with least elements.

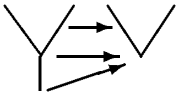
The category **Predom** is distributive [Wal89], *i.e.* it has finite products and coproducts, created by cartesian product and disjoint union of the underlying sets respectively and a natural isomorphism

$$\mathtt{dist}_{A,B,C}: \ A \times (B + C) \to A \times B + A \times C$$

There is also a strong monad [Mog91] of *lifting*, $((\cdot)_\perp, \mathtt{up}, \mu, t)$ where $P_\perp$ is the domain obtained by adjoining a bottom element to P; $\mathtt{up}_P : P \to P_\perp$ and $\mu_P : P_{\perp\perp} \to P_\perp$ are illustrated by:



up



μ

and $t_{AB} : A \times B_\perp \to (A \times B)_\perp$ is the tensorial strength [Mog91].

Domains arise precisely as the *algebras* of this monad. Given a domain D, we write its structure map as $\alpha_D : D_\perp \to D$. Since $0_\perp = 1$, we get a map

$$1 \xrightarrow{(0_D)_\perp} D_\perp \xrightarrow{\alpha_D} D$$

for each domain, and hence $\perp_{AD}$: $A \to 1 \to D$ for each predomain A.

We also recall that domains are closed under products (in **Predom**). For each endomorphism $f : D \to D$ in **Dom**, the least fixpoint of f is given by

$$Y(f) = \bigsqcup_{k \in \omega} f^k(\perp)$$

next, we need to consider for each domain D, the functor T_D : **Dom** $\to$ **Dom** defined by $T_D(E) = (D + 1 + (E \times E))_\perp$. For each D, the initial T_D algebra exists and is denoted by $(\mathcal{T}D, \mathtt{fold}_D)$, with $\mathtt{fold}_D : (D + 1 + (\mathcal{T}D \times \mathcal{T}D))_\perp \to \mathcal{T}D$ being an isomorphism. The inverse of $\mathtt{fold}_D$ is written as $\mathtt{unfold}_D$.

A $\mathcal{GI}$-category is a subcategory of **Predom** closed under all of the above constructions. Some examples:

- Any of the usual full subcategories of **Predom** (**Dom**) considered in denotational semantics, *e.g.* Scott domains, SFP, continuous cpo's.
- Any of the usual categories of domains and *stable* functions, *e.g.* dI domains, L-domains. However, coherence spaces are not an example, since they are not closed under lifting.
- The category of sequential functions on concrete domains [KP78]. In particular, note that $\mathcal{GI}$-categories are not required to be cartesian closed.

3.2 $\mathcal{GI}$-models

We will now describe the construction of a model $\mathcal{GI}(\mathbb{C})$ of Classical Linear Logic from any $\mathcal{GI}$-category $\mathbb{C}$. This construction contains all the essential ingredients of the Geometry of Interaction interpretation in a more synthetic form, and provides a good point of entry to the interpretation, which will be described in the next Section.

Firstly, we shall briefly recall the definition of CLL models; for more details see [See89]. Let $\mathbb{C}$ be a symmetric monoidal closed category with tensor $\otimes$, unit I , internal hom $\multimap$. An object $\perp$ is *dualizing* if for all A, the morphism $A \to (A \multimap \perp) \multimap \perp$, obtained by currying the evaluation map, is an isomorphism. A *$\star$-autonomous category* is a symmetric monoidal closed category with a dualizing object. A *CLL-model* is a $\star$-autonomous category with finite products and coproducts, and a comonad $(!, \epsilon, \delta)$ together with natural isomorphisms

$$\begin{array}{rcl} !A \otimes !B & \simeq & !(A \times B) \\ I & \simeq & !1 \end{array}$$

The models we shall construct will in fact satisfy a slightly weaker form of these axioms; the situation is analogous to giving a model of $\lambda\beta$ calculus rather than $\lambda\beta\eta$. Specifically, we only have *lax* products and coproducts, and the isomorphism described above are replaced by embedding-projection pairs. This description uses the fact that $\mathcal{GI}$-categories are poset-enriched, using the pointwise (or in the appropriate cases, the stable) ordering on homsets.

We shall now describe $\mathcal{GI}(\mathbb{C})$. Verification that these constructions work as advertised is omitted, since these arguments are very similar to establishing the soundness of the $\mathcal{GI}$-interpretation, which is proved in detail in Section 6.

Objects: Domains in $\mathbb{C}$

Morphisms: $f : A \rightsquigarrow B \in \mathcal{GI}(\mathbb{C})$ is $f : A \times B \to A \times B \in \mathbb{C}$.

Identities: $I_A : A \rightsquigarrow A \in \mathcal{GI}(\mathbb{C})$ is $\hat{1} : A \times A \to A \times A \in \mathbb{C}$.

Composition: If $f : A \rightsquigarrow B$, $g : B \rightsquigarrow C \in \mathcal{GI}(\mathbb{C})$, then $g \circ f : A \rightsquigarrow C$ is the projection onto (a', b') of the least solution of the equations:

$$\begin{array}{rcl} (a', x) & = & f(a, y) \\ (y, b') & = & g(x, b) \end{array}$$

Proposition 1 *$\mathcal{GI}(\mathbb{C})$ is a category.*

Involution

We define a duality $(\cdot)^{\perp} : \mathcal{GI}(\mathbb{C})^{op} \to \mathcal{GI}(\mathbb{C})$ by

$$\begin{aligned} A^{\perp} &= A \\ f^{\perp} &= B \times A \xrightarrow{\mathtt{symm}} A \times B \xrightarrow{f} A \times B \xrightarrow{\mathtt{symm}} B \times A \end{aligned}$$

where $f : A \rightsquigarrow B$. Clearly, this is a functor and $(\cdot)^{\perp\perp} = Id$.

Multiplicatives

Define

$$\begin{aligned} A \otimes B &\stackrel{\text{def}}{=} A \times B \\ I &\stackrel{\text{def}}{=} 1 \end{aligned}$$

To complete the definition of tensor as a functor, we define given $f : A \rightsquigarrow B$, $g : A' \rightsquigarrow B'$,

$$f \otimes g = (A \times A') \times (B \times B') \xrightarrow{\sigma^{-1}} (A \times B) \times (A' \times B') \xrightarrow{f \times g} (A \times B) \times (A' \times B') \xrightarrow{\sigma} (A \times A') \times (B \times B')$$

where $\sigma \langle \langle a, b \rangle, \langle a', b' \rangle \rangle = \langle \langle a, a' \rangle, \langle b, b' \rangle \rangle$.

The monoidal structure of product on $\mathbb{C}$ is transported to the tensor product on $\mathcal{GI}(\mathbb{C})$ via the embedding $f \mapsto \hat{f}$ of isomorphisms; note that $\widehat{f^{-1}} = \hat{f}^{\perp} = (\hat{f})^{-1}$. Thus, we define

$$\begin{aligned} \mathtt{Assoc} : &\quad (A \otimes B) \otimes C \rightsquigarrow A \otimes (B \otimes C) \\ \mathtt{Symm} : &\quad A \otimes B \rightsquigarrow B \otimes A \\ \mathtt{Unit} : &\quad A \otimes I \rightsquigarrow A \end{aligned}$$

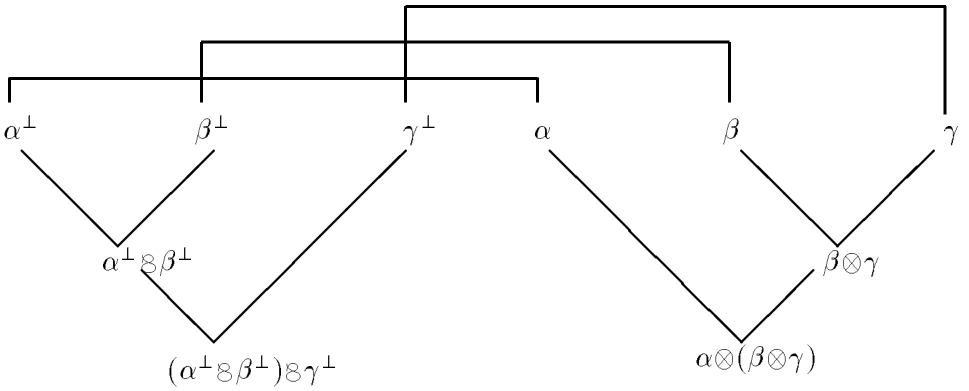
by $\mathtt{Assoc} = \widehat{\mathtt{assoc}}$, $\mathtt{Symm} = \widehat{\mathtt{symm}}$, $\mathtt{Unit} = \widehat{\mathtt{unit}}$, where

$$\begin{aligned} \mathtt{assoc} : &\quad (A \otimes B) \otimes C \simeq A \otimes (B \otimes C) \\ \mathtt{symm} : &\quad A \otimes B \simeq B \otimes A \\ \mathtt{unit} : &\quad A \otimes I \simeq A \end{aligned}$$

Thus, for example,

$$\mathtt{Assoc} \langle \langle \langle x, y \rangle, z \rangle, \langle \langle u, v \rangle, w \rangle \rangle = \langle \langle \langle u, v \rangle, w \rangle, \langle \langle x, y \rangle, z \rangle \rangle$$

Compare this with the proof net:



The application map $\mathtt{Ap} : (A \multimap B) \otimes A \rightsquigarrow B$ is defined by

$$\mathtt{Ap}\langle\langle\langle x, y\rangle, u\rangle, v\rangle = \langle\langle\langle u, v\rangle, x\rangle, y\rangle$$

Given $f : A \otimes B \rightsquigarrow C, \Lambda(f) : A \rightsquigarrow (B \multimap C)$ is defined as:

$$\Lambda(f) = A \times (B \times C) \xrightarrow{\mathtt{assoc}^{-1}} (A \times B) \times C \xrightarrow{f} (A \times B) \times C \xrightarrow{\mathtt{assoc}} A \times (B \times C)$$

Finally, the isomorphism $(A \multimap \bot) \multimap \bot \rightsquigarrow A$ is induced by $(A \times 1) \times 1 \simeq A$.

Proposition 2 *With these definitions, $\mathcal{GI}(\mathbb{C})$ is a $\star$-autonomous category.*

Additives

Define

$$\begin{aligned} A \& B &\stackrel{\text{def}}{=} (A + B)_\perp \\ A \oplus B &\stackrel{\text{def}}{=} (A + B)_\perp \end{aligned}$$

Since, $\mathcal{GI}(\mathbb{C})$ is self-dual under $(\cdot)^\perp$, it suffices to discuss products.

$$\begin{tikzcd} A & A \& B \arrow[l, "\mathtt{Fst}"'] \arrow[r, "\mathtt{Snd}"] & B \\ & C \arrow[ul, "f"] \arrow[u, "\langle f, g\rangle"'] \arrow[ur, "g"'] & \end{tikzcd}$$

Firstly, we define retracts

$$\begin{aligned} l : A &\lhd (A + B)_\perp : l^\star \\ r : A &\lhd (A + B)_\perp : r^\star \end{aligned}$$

as follows.

$$
\begin{array}{ll}
l = \texttt{inl}; \texttt{up} & l^\star = (A+B)_\perp \stackrel{[1,\perp]_\perp}{\longrightarrow} A_\perp \stackrel{\alpha_A}{\longrightarrow} A \\
r = \texttt{inr}; \texttt{up} & l^\star = (A+B)_\perp \stackrel{[\perp,1]_\perp}{\longrightarrow} A_\perp \stackrel{\alpha_B}{\longrightarrow} A
\end{array}
$$

Now, define

$$
\begin{array}{rcl}
\texttt{Fst} & = & (A+B)_\perp \times A \stackrel{l^\star\times 1}{\longrightarrow} A \times A \stackrel{\texttt{symm}}{\longrightarrow} A \times A \stackrel{l\times 1}{\longrightarrow} (A+B)_\perp \times A \\
\texttt{Snd} & = & (A+B)_\perp \times B \stackrel{r^\star\times 1}{\longrightarrow} B \times B \stackrel{\texttt{symm}}{\longrightarrow} B \times B \stackrel{r\times 1}{\longrightarrow} (A+B)_\perp \times B
\end{array}
$$

Note that Fst, Snd are both "$\perp$-raising", *i.e.* non-strict maps. For example,

$$
\begin{array}{rcl}
\texttt{Fst}\langle \texttt{inl}(x), y\rangle & = & \langle \texttt{inl}(y), x\rangle \\
\texttt{Fst}\langle \perp, y\rangle & = & \langle \texttt{inl}(y), \perp\rangle \\
\texttt{Fst}\langle \texttt{inr}(x), y\rangle & = & \langle \texttt{inl}(y), \perp\rangle
\end{array}
$$

Given $f : C \rightsquigarrow A$, $g : C \rightsquigarrow B$, define $\langle f, g\rangle : C \rightsquigarrow A\&B$ by

$$
\begin{array}{rcl}
\langle f, g\rangle & = & C \times (A+B)_\perp \stackrel{t}{\rightarrow} (C \times (A+B))_\perp \stackrel{\texttt{dist}_\perp}{\longrightarrow} (C \times A + C \times B)_\perp \\
& & \stackrel{[(1\times l)\circ f, (1\times r)\circ g]_\perp}{\longrightarrow} (C \times (A+B)_\perp)_\perp \stackrel{\alpha}{\rightarrow} C \times (A+B)_\perp
\end{array}
$$

Note that $\langle f, g\rangle$ is strict in its second argument; if no information is given as to how the choice between A and B is to resolved, no information can be produced.

$$
\begin{array}{rcl}
\langle f, g\rangle(u, \perp) & = & \perp \\
\langle f, g\rangle(u, \texttt{inl}(y)) & = & (u', \texttt{inl}(y')),\ f(u, y) = (u', y') \\
\langle f, g\rangle(u, \texttt{inr}(y)) & = & (u', \texttt{inl}(y')),\ g(u, y) = (u', y')
\end{array}
$$

The non-strictness of the projections is then essential in order that no "deadlocks" occur when a projection is composed with pairing. This construction is actually a *weak product* in the following sense.

Proposition 3

$$
\begin{array}{rcl}
f & = & \texttt{Fst} \circ \langle f, g\rangle \\
g & = & \texttt{Snd} \circ \langle f, g\rangle \\
h & \sqsubseteq & \langle \texttt{Fst} \circ h, \texttt{Snd} \circ h\rangle \\
y \neq \perp \Rightarrow h(x, y) & = & \langle \texttt{Fst} \circ h, \texttt{Snd} \circ h\rangle(x, y)
\end{array}
$$

Thus, we "almost" get a categorical product. Analogous equations hold in the lazy λ-calculus [Abr90] and the λ_p-calculus [Mog86].

Additive units

Define $0 \stackrel{\text{def}}{=} T \stackrel{\text{def}}{=} (0)_\perp = 1$. This is a lax zero object; for any domain A we define

$$T_A = A \times 1 \xrightarrow{\perp} A \times 1$$

Clearly $T_A \sqsubseteq f$ for any $f : A \rightsquigarrow T$, and trivially we have

$$(\forall y \neq \perp)\, [T_A(x,y) = f(x,y)]$$

Exponentials

Define

$$\begin{aligned} !A &\stackrel{\text{def}}{=} \mathcal{T}A \\ ?A &\stackrel{\text{def}}{=} \mathcal{T}A \end{aligned}$$

Again, we define retracts

$$\begin{aligned} d : A &\lhd \mathcal{T}A : d^\star \quad [\text{"dereliction"}] \\ w : 1 &\lhd \mathcal{T}A : w^\star \quad [\text{"weakening"}] \\ c : \mathcal{T}A \times \mathcal{T}A &\lhd \mathcal{T}A : c^\star \quad [\text{"contraction"}] \end{aligned}$$

as follows.

$$\begin{array}{ll} d = \mathtt{in}_1; \mathtt{up}; \mathtt{fold}_A & d^\star = \mathtt{unfold}_A; [1, \perp, \perp]_\perp; \alpha \\ w = \mathtt{in}_2; \mathtt{up}; \mathtt{fold}_A & w^\star = \mathtt{unfold}_A; [\perp, 1, \perp]_\perp; \alpha \\ c = \mathtt{in}_3; \mathtt{up}; \mathtt{fold}_A & c^\star = \mathtt{unfold}_A; [\perp, \perp, 1]_\perp; \alpha \end{array}$$

We define the counit $\epsilon_A : !A \rightsquigarrow A$ by

$$\epsilon_A = \mathcal{T}_A \times A \xrightarrow{d^\star \times 1} A \times A \xrightarrow{I_A} A \times A \xrightarrow{d \times 1} \mathcal{T}_A \times A$$

Given $f : !A \rightsquigarrow B$, define $f^\dagger : !A \rightsquigarrow !B$ by:

$$\begin{aligned} f^\dagger \;=\; & \mathcal{T}A \times \mathcal{T}B \xrightarrow{1 \times \mathtt{unfold}_B} \mathcal{T}A \times (B + 1 + \mathcal{T}B \times \mathcal{T}B)_\perp \xrightarrow{t} (\mathcal{T}A \times (B + 1 + \mathcal{T}B \times \mathcal{T}B))_\perp \\ & \xrightarrow{\mathtt{dist}_\perp} (\mathcal{T}A \times B + \mathcal{T} \times 1 + \mathcal{T}A \times (\mathcal{T}B \times \mathcal{T}B))_\perp \xrightarrow{[g,h,k]_\perp} (\mathcal{T}A \times \mathcal{T}B)_\perp \xrightarrow{\alpha} \mathcal{T}A \times \mathcal{T}B \end{aligned}$$

where

$$\begin{aligned} g \;=\; & \mathcal{T}A \times B \xrightarrow{f} \mathcal{T}A \times B \xrightarrow{1 \times d} \mathcal{T}A \times \mathcal{T}B \\ h \;=\; & \mathcal{T}A \times 1 \xrightarrow{w^\star \times 1} 1 \times 1 \xrightarrow{w \times w} \mathcal{T}A \times \mathcal{T}B \\ k \;=\; & \mathcal{T}A \times (\mathcal{T}B \times \mathcal{T}B) \xrightarrow{c^\star \times 1} (\mathcal{T}A \times \mathcal{T}A) \times (\mathcal{T}B \times \mathcal{T}B) \xrightarrow{\sigma^{-1}} (\mathcal{T}A \times \mathcal{T}B) \times (\mathcal{T}A \times \mathcal{T}B) \\ & \xrightarrow{f^\dagger \times f^\dagger} (\mathcal{T}A \times \mathcal{T}B) \times (\mathcal{T}A \times \mathcal{T}B) \xrightarrow{\sigma} (\mathcal{T}A \times \mathcal{T}A) \times (\mathcal{T}B \times \mathcal{T}B) \xrightarrow{c \times c} \mathcal{T}A \times \mathcal{T}B \end{aligned}$$

where $\sigma\langle\langle a,b\rangle,\langle c,d\rangle\rangle = \langle\langle a,c\rangle,\langle b,d\rangle\rangle$. For example,

$$\begin{array}{rcl} \epsilon_A\langle d(x),y\rangle & = & \langle d(y),x\rangle \\ f^\dagger\langle x,\bot\rangle & = & \langle\bot,\bot\rangle \\ f^\dagger\langle x,d(y)\rangle & = & \langle x',d(y')\rangle \\ f^\dagger\langle w,w\rangle & = & \langle w,w\rangle \\ f^\dagger\langle c(x,y),c(u,v)\rangle & = & \langle c(x',y'),c(u',v')\rangle \end{array}$$

where $f\langle x,y\rangle = \langle x',y'\rangle, f^\dagger\langle x,u\rangle = \langle x',u'\rangle, f^\dagger\langle y,v\rangle = \langle y',v'\rangle$. Note that $f^\dagger$ is "demand-driven" by its second argument.

Now, given $f : A \leadsto B$, we can define $!f = (f \circ \epsilon_A)^\dagger : !A \leadsto !B$, and $\delta_A = (I_{!A})^\dagger : !A \leadsto !!A$.

Proposition 4 $(!, \epsilon, \delta)$ *is a comonad.*

Also, define $\kappa : I \leadsto !1$ as:

$$\kappa = 1 \times \mathcal{T}1 \xrightarrow{1\times w^\star} 1 \times 1 \xrightarrow{1\times w} 1 \times \mathcal{T}1$$

Finally, define $\iota : !A \otimes !B \leadsto !(A\&B)$, as follows:

$$\begin{array}{rcl} \iota & = & (\mathcal{T}A \times \mathcal{T}B) \times \mathcal{T}(A+B)_\bot \xrightarrow{1\times c^\star} (\mathcal{T}A \times \mathcal{T}B) \times \mathcal{T}(A+B)_\bot \times \mathcal{T}(A+B)_\bot \\ & & \xrightarrow{\sigma^{-1}} (\mathcal{T}A \times \mathcal{T}(A+B)_\bot) \times (\mathcal{T}B \times \mathcal{T}(A+B)_\bot) \\ & & \xrightarrow{!(\mathtt{Fst}^\bot)\times !(\mathtt{Snd}^\bot)} (\mathcal{T}A \times \mathcal{T}(A+B)_\bot) \times (\mathcal{T}B \times \mathcal{T}(A+B)_\bot) \\ & & \xrightarrow{\sigma} (\mathcal{T}A \times \mathcal{T}B) \times \mathcal{T}(A+B)_\bot \times \mathcal{T}(A+B)_\bot \\ & & \xrightarrow{1\times c} (\mathcal{T}A \times \mathcal{T}B) \times \mathcal{T}(A+B)_\bot \end{array}$$

Taking $\iota^\star = \iota^\bot, \kappa^\star = \kappa^\bot$: we get the following proposition.

Proposition 5 *The following are embedding-projection pairs:*

$$\begin{array}{rcl} \iota : \mathcal{T}A \otimes \mathcal{T}B & \lhd & \mathcal{T}(A\&B) : \iota^\star \\ \kappa : I & \lhd & !1 : \kappa^\star \end{array}$$

4 $\mathcal{GI}$-interpretations

4.1 Type-free models

We will now present our version of the Geometry of Interaction interpretation of Classical Linear Logic. This can be seen as arising from the model $\mathcal{GI}(\mathbb{C})$ described in the previous section in two steps:

1. Moving from a typed to a type-free model.
2. Introducing a suitable normal form, so that the entire process of Cut-elimination is captured by a single fixpoint computation at the top level.

The first step is familiar from the semantics of the type-free λ-calculus [Sco80]. In that context, a model of the type-free λ-calculus can be defined as a reflexive object in a cartesian closed category, *i.e.* an object A equipped with a retraction $A^A \lhd A$. We shall define a *type-free $\mathcal{GI}$ model* to be an object D in a $\mathcal{GI}$ category such that D is a domain equipped with retractions:

$$
\begin{aligned}
u : 1 &\quad\lhd\quad D : u^\star \\
m : D^2 &\quad\lhd\quad D : m^\star \\
a : (D + D)_\perp &\quad\lhd\quad D : a^\star \\
e : \mathcal{T}D &\quad\lhd\quad D : e^\star
\end{aligned}
$$

Examples are not hard to find. We shall describe perhaps the simplest and most obvious, which also forms a natural basis for a concrete implementation.

Consider the (one-sorted) signature Σ specified by

$$\Sigma_0 = \{\mathtt{u}, \mathtt{w}\},\ \Sigma_1 = \{\mathtt{l}, \mathtt{r}, \mathtt{d}\},\ \Sigma_2 = \{\mathtt{m}, \mathtt{c}\}$$

We write $W(\Sigma, X)$ for the free Σ-algebra on a set of generators X. The free ordered Σ-algebra [GTWW77, Gue81] $W_\perp(\Sigma, X)$ can be constructed as follows: the Σ-algebra structure is that of $W(\Sigma, X \cup \{\perp\})$ (where $\perp \notin X$), and the order is generated by $\perp \sqsubseteq x$, subject to the condition that all operations are monotone. This is the "$\perp$-match" ordering: $t \sqsubseteq u$ just if u can be obtained from t by replacing the $\perp$-leaves of t by arbitrary terms. Finally, $W_\perp^\infty(\Sigma, X)$ is the free continuous Σ-algebra generated by X; this can be constructed as the ideal completion of $W_\perp(\Sigma, X)$.

We take $D = W_\perp^\infty(\Sigma, X)$. This is obviously a domain. Define u, m, a, e as follows:

$$
\begin{array}{ll}
u(\cdot) = \mathtt{u} & u^\star(z) = \cdot \\
m(t,u) = \mathtt{m}(t,u) & m^\star(z) = \begin{cases} \langle t, u\rangle, z = \mathtt{m}(t,u) \\ \langle \perp, \perp\rangle, \text{otherwise} \end{cases}
\end{array}
$$

$$
\begin{array}{ll}
\begin{array}{l} a(\perp) = \perp \\ a(\mathtt{inl}(t)) = \mathtt{l}(t) \\ a(\mathtt{inr}(t)) = \mathtt{r}(t) \end{array} & a^\star(z) = \begin{cases} \mathtt{inl}(t), z = \mathtt{l}(t) \\ \mathtt{inr}(t), z = \mathtt{r}(t) \\ \perp, \text{otherwise} \end{cases}
\end{array}
$$

$$
\begin{array}{ll}
\begin{array}{l} e(\perp) = \perp \\ e(\mathtt{in}_1(t)) = \mathtt{d}(t) \\ e(\mathtt{in}_2(\cdot)) = \mathtt{w} \\ e(\mathtt{in}_3(t,u)) = \mathtt{c}(t', u') \\ \quad t' = e(t), u' = e(u) \end{array} & e^\star(z) = \begin{cases} \mathtt{in}_1(t), z = \mathtt{d}(t) \\ \mathtt{in}_2(\cdot), z = \mathtt{w} \\ \mathtt{in}_3(e^\star(t), e^\star(u)), \\ \quad z = \mathtt{c}(t,u) \\ \perp, \text{otherwise} \end{cases}
\end{array}
$$

The recursion in the definition of $e, e^\star$ is interpreted as a least fixpoint.

4.2 Interpreting $\mathtt{CLL}_2$ in type-free models

Having specified a type-free $\mathcal{GI}$ model D, the idea is to use this as a "universal domain", using the retractions to internalize the definitions of the Linear proof combinators, with all types denoting D. In particular, the second order quantifiers can be interpreted trivially, in the "Curry-style" [BH90].

But this is not quite the end of the story. The final ingredient is the normal form, which pushes all the fixpoints to the top level. This idea is also quite well known, at least as folklore in the dataflow literature, and more generally in Domain theory.

The idea is to carry the information about the cuts used in the proof in the object assigned to the proof. Thus, if Π is a proof of a sequent $\vdash A_1, \ldots A_m$ in which the Cut rule has been applied m times, to formulas $B_1, B_1^\perp, \ldots B_m, B_m^\perp$, then the object assigned to Π by the interpretation will be a function $f : D^{2m+n} \to D^{2m+n}$.

If Π' is a proof of $\vdash \Gamma, A$ interpreted by a function $f' : D^{2m'+n'+1} \to D^{2m'+n'+1}$, and Π'' is a proof of $\vdash \Delta, A^\perp$ interpreted by a function $f'' : D^{2m''+n''+1} \to D^{2m''+n''+1}$, then $\mathrm{Cut}(\Pi', \Pi'')$ will be interpreted by

$$\tau \circ (f' \times f'') \circ \tau^{-1} : D^{2(m+1)+n} \to D^{2(m+1)+n}$$

where $\tau\langle \vec{u}, \vec{x}, x, \vec{v}, \vec{y}, y\rangle = \langle \vec{u}, \vec{v}, x, y, \vec{x}, \vec{y}\rangle$, $m = m' + m''$ and $n = n' + n''$.

Given $f : D^{2m+n} \to D^{2m+n}$, define the *feedback formula*, $FB(f, \sigma) : D^n \to D^n$:

$$\begin{array}{rcl} f'(\vec{x}) & = & Y[\lambda \vec{u}.\pi' \circ (\sigma \times 1) \circ f(\vec{u}, \vec{x})] \\ FB(f,\sigma)(\vec{x}) & = & \pi \circ f(f'(\vec{x}), \vec{x}) \end{array}$$

where $\pi(\vec{u}, \vec{x}) = \vec{x}$, $\pi'(\vec{u}, \vec{x}) = \vec{u}$. The permutation $\sigma(x_1, x_2 \ldots x_{2m-1}, x_{2m}) = (x_2, x_1 \ldots x_{2m}, x_{2m-1})$ is used to represent the flow of information through the cuts. We write $FB(f, \sigma) = \pi(\bigsqcup_k f^{(k)})$, where $f^{(k)} : D^n \to D^{2m+n}$ is defined inductively by

$$\begin{array}{rcl} f^{(0)} & = & \perp \\ f^{(k+1)} & = & (\sigma \times 1) \circ f \circ \langle \pi' \circ f^{(k)}, 1\rangle \end{array}$$

The intention is that if (f, σ) is the interpretation of a proof Π in $\mathtt{CLL}_2$ (second order linear logic), then $FB(f, \sigma)$ will be the interpretation of any cut free proof Π' which can be obtained by performing cut-elimination on Π. The dynamics of the cut-elimination process itself is modelled by the sequence of iterations to the fixpoint: $f^{(0)}, f^{(1)} \ldots$, and strong normalisation will be mirrored by a *finite convergence* property: $f^{(k)} = f^{(k+1)} = \bigsqcup_k f^{(k)}$ for some $k \in \omega$.

We will now proceed to specify the interpretation of proofs in Classical Linear Logic. We write $f \vdash [\Delta], \Gamma$ to denote that f is the interpretation of a proof Π of the sequent Γ with cuts Δ. We use $f; g$ for diagram order composition.

Axiom: $\dfrac{}{I \vdash \alpha, \alpha^\perp}$ where $I = \mathtt{symm}$, with $\mathtt{symm}(x, y) = (y, x)$.

Exchange : $\dfrac{f \vdash [\Delta], \Gamma}{Ex_\sigma(f) \vdash [\Delta], \sigma\Gamma}$ where $Ex_\sigma(f) = 1 \times \sigma^{-1}; f; 1 \times \sigma$.

Cut : $\dfrac{f \vdash [\Delta'], \Gamma', A \qquad g \vdash [\Delta''], \Gamma'', A^\perp}{f \cdot g \vdash [\Delta', \Delta'', A, A^\perp], \Gamma', \Gamma''}$ where $f \cdot g = \sigma^{-1}; f \times g; \sigma$, with $\sigma(\vec{u}, \vec{x}, x, \vec{v}, \vec{y}, y) = (\vec{u}, \vec{v}, x, y, \vec{x}, \vec{y})$

Multiplicatives: Let σ be defined by $\sigma(\vec{u}, \vec{x}, x, \vec{v}, \vec{y}, y) = (\vec{u}, \vec{v}, \vec{x}, \vec{y}, x, y)$.

Tensor $\dfrac{f \vdash [\Delta'], \Gamma', A \qquad g \vdash [\Delta''], \Gamma'', B}{\otimes(f, g) \vdash [\Delta', \Delta''], \Gamma', \Gamma'', A \otimes B}$ $\quad \otimes(f, g) = 1 \times m^\star; \sigma^{-1}; f \times g; \sigma; 1 \times m$

Par $\dfrac{f \vdash [\Delta], \Gamma, A, B}{⅋(f) \vdash [\Delta], \Gamma, A ⅋ B}$ $\quad ⅋(f) = 1 \times m^\star; f; 1 \times m$

Unit $\dfrac{}{U \vdash I}$ $\quad U = u^\star; u$

Perp $\dfrac{f \vdash [\Delta], \Gamma}{\perp(f) \vdash [\Delta], \Gamma, \perp}$ $\quad \perp(f) = 1 \times u^\star; f \times 1; 1 \times u$

Additives: For the additives,we need some auxiliary definitions: We define retracts

$$l : D \ \lhd \ D : l^\star$$
$$r : D \ \lhd \ D : r^\star$$

as follows.

$$l = \mathtt{inl}; \mathtt{up}; a \qquad l^\star = a^\star; [1, \perp]_\perp; \alpha$$
$$r = \mathtt{inr}; \mathtt{up}; a \qquad r^\star = a^\star; [\perp, 1]_\perp; \alpha$$

Plus left: $\dfrac{f \vdash [\Delta], \Gamma, A}{L(f) \vdash [\Delta], \Gamma, A \oplus B}$ where $L(f) = 1 \times l^\star; f; 1 \times l$.

Plus right: $\dfrac{f \vdash [\Delta], \Gamma, B}{R(f) \vdash [\Delta], \Gamma, A \oplus B}$ where $R(f) = 1 \times r^\star; f; 1 \times r$

With: Given $f : D^{2m'+n+1} \to D^{2m'+n+1}$, $g : D^{2m''+n+1} \to D^{2m''+n+1}$, let $m = m' + m''$.

$$\frac{f \vdash [\Delta'], \Gamma, A \qquad g \vdash [\Delta''], \Gamma, B}{\&(f, g) \vdash [\Delta', \Delta''], \Gamma, A \& B}$$

where

$$\begin{aligned} \&(f, g) \quad = \quad & D^{2m+n+1} \xrightarrow{1 \times a^\star} D^{2m+n} \times (D + D)_\perp \xrightarrow{t} (D^{2m+n} \times (D + D))_\perp \\ & \xrightarrow{dist_\perp} (D^{2m+n} \times D + D^{2m+n} \times D)_\perp \xrightarrow{[h,k]_\perp} (D^{2m+n} \times D)_\perp \xrightarrow{\alpha} D^{2m+n+1} \end{aligned}$$

where $\pi', \pi'', \rho', \rho''$ are defined as:

$$\pi'(\vec{u}, \vec{v}, \vec{x}) = (\vec{u}, \vec{x}) \qquad \rho'(\vec{u}, \vec{x}) = (\vec{u}, \vec{\perp}, \vec{x})$$
$$\pi''(\vec{u}, \vec{v}, \vec{x}) = (\vec{v}, \vec{x}) \qquad \rho'(\vec{v}, \vec{x}) = (\vec{\perp}, \vec{v}, \vec{x})$$

in the definition of h, k:

$$\begin{aligned} h \quad &= \quad D^{2m+n+1} \xrightarrow{\pi'} D^{2m'+n+1} \xrightarrow{f} D^{2m'+n+1} \xrightarrow{\rho'} D^{2m+n+1} \\ k \quad &= \quad D^{2m+n+1} \xrightarrow{\pi''} D^{2m''+n+1} \xrightarrow{g} D^{2m''+n+1} \xrightarrow{\rho''} D^{2m+n+1} \end{aligned}$$

Top: $\dfrac{}{T \vdash \Gamma, \top}$, where $T = \perp$

Exponentials: For the exponentials, we again need some auxiliary definitions: We define retracts

$$\begin{array}{rcl} d : D & \lhd & D : d^\star \\ w : 1 & \lhd & D : w^\star \\ c : \mathcal{T}D \times \mathcal{T}D & \lhd & D : c^\star \end{array}$$

as follows:

$$\begin{array}{ll} d = \mathtt{in}_1; \mathtt{up}; \mathtt{fold}; e & d^\star = e^\star;\ \mathtt{unfold}; [1, \perp, \perp]_\perp; \alpha \\ w = \mathtt{in}_2; \mathtt{up}; \mathtt{fold}; e & w^\star = e^\star;\ \mathtt{unfold}; [\perp, 1, \perp]_\perp; \alpha \\ c = \mathtt{in}_3; \mathtt{up}; \mathtt{fold}; e & c^\star = e^\star;\ \mathtt{unfold}; [\perp, \perp, 1]_\perp; \alpha \end{array}$$

Dereliction: $\dfrac{f \vdash [\Delta], \Gamma, A}{D(f) \vdash [\Delta], \Gamma, ?A}$ where $D(f) = 1 \times d^\star; f; 1 \times d$

Weakening: $\dfrac{f \vdash [\Delta], \Gamma}{W(f) \vdash [\Delta], \Gamma, ?A}$ where $W(f) = 1 \times w^\star; f \times 1; 1 \times w$

Contraction: $\dfrac{f \vdash [\Delta], \Gamma, ?A, ?A}{C(f) \vdash [\Delta], \Gamma, ?A}$ where $C(f) = 1 \times c^\star; f; 1 \times c$

Of course: $\dfrac{f \vdash [\Delta], ?\Gamma, A}{!(f) \vdash [\Delta], ?\Gamma, !A}$

Define

$$\begin{array}{rcl} !(f) & = & D^{2m+n} \times D \xrightarrow{1 \times (e^\star;\ \mathtt{unfold})} D^{2m+n} \times (D + 1 + \mathcal{T}D \times \mathcal{T}D)_\perp \\ & & \xrightarrow{t} (D^{2m+n} \times (D + 1 + \mathcal{T}D \times \mathcal{T}D))_\perp \\ & & \xrightarrow{dist_\perp} (D^{2m+n} \times D + D^{2m+n} \times 1 + D^{2m+n} \times \mathcal{T}D \times \mathcal{T}D)_\perp \\ & & \xrightarrow{[g,h,k]_\perp} (D^{2m+n} \times D)_\perp \\ & & \xrightarrow{\alpha} D^{2m+n} \times D \end{array}$$

where $\sigma(x_1 \dots x_{2m+n+1}, y_1 \dots y_{2m+n+1}) = (x_1, y_1 \dots x_{2m+n+1}, y_{2m+n+1})$ in

$$\begin{array}{rcl} g & = & D^{2m+n} \times D \xrightarrow{f} D^{2m+n} \times D \xrightarrow{1 \times d} D^{2m+n} \times D \\ h & = & D^{2m+n} \times 1 \xrightarrow{\vec{w}^\star \times 1} 1^{2m+n} \times 1 \xrightarrow{\vec{w} \times w} D^{2m+n} \times D \\ k & = & D^{2m+n} \times (\mathcal{T}D \times \mathcal{T}D) \xrightarrow{\vec{c}^\star \times 1} (\mathcal{T}D \times \mathcal{T}D)^{2m+n} \times (\mathcal{T}D \times \mathcal{T}D) \\ & & \xrightarrow{\sigma^{-1}} \mathcal{T}D^{2m+n+1} \times \mathcal{T}D^{2m+n+1} \\ & & \xrightarrow{\vec{e} \times \vec{e}} D^{2m+n+1} \times D^{2m+n+1} \\ & & \xrightarrow{!(f) \times !(f)} D^{2m+n+1} \times D^{2m+n+1} \\ & & \xrightarrow{\vec{e}^\star \times \vec{e}^\star} \mathcal{T}D^{2m+n+1} \times \mathcal{T}D^{2m+n+1} \xrightarrow{\sigma} (\mathcal{T}D \times \mathcal{T}D)^{2m+n} \times (\mathcal{T}D \times \mathcal{T}D) \\ & & \xrightarrow{\vec{c} \times c} D^{2m+n} \times D \end{array}$$

Quantifiers: The quantifiers are interpreted trivially.

All: $\dfrac{f \vdash [\Delta], \Gamma, A}{\forall(f) \vdash [\Delta], \Gamma, \forall\alpha.A}$, if α not free in Γ. Define $\forall(f) = f$.

Exists: $\dfrac{f \vdash [\Delta], \Gamma, A[B/\alpha]}{\exists(f) \vdash [\Delta], \Gamma, \exists\alpha.A}$ where $\exists(f) = f$.

Definition by elements

The definitions given above are written in an element free style, using the categorical combinators from the underlying category of predomains. This style has a number of advantages: it is rather concise and forms the basis for an axiomatic approach; and it admits a very natural and direct translation into dataflow graphs. However, the reader may appreciate a more traditional definition using variables ranging over elements. We give a few sample cases in this style and the reader should have no problem in translating the remaining cases similarly.

Tensor: Let $m^\star(z) = \langle x, y\rangle, f\langle \vec{u}, \vec{x}, x\rangle = \langle \vec{u}', \vec{x}', x'\rangle, g\langle \vec{v}, \vec{y}, y\rangle = \langle \vec{v}', \vec{y}', y'\rangle$ in

$$\otimes(f, g)\langle \vec{u}, \vec{v}, \vec{x}, \vec{y}, z\rangle\rangle = \langle \vec{u}', \vec{v}', \vec{x}', \vec{y}', m(x', y')\rangle$$

With: This requires a **case** statement.

$$\&(f,g)\langle \vec{u}, \vec{v}, \vec{x}, z\rangle = \begin{cases} \textbf{case } a^\star(z) \textbf{ of} \\ \quad \texttt{inl}(x) : \langle \vec{u}', \vec{\bot}, \vec{x}', l(x')\rangle, \text{ where } f\langle \vec{u}, \vec{x}, x\rangle = \langle \vec{u}', \vec{x}', x'\rangle \\ \quad \texttt{inr}(y) : \langle \vec{\bot}, \vec{v}', \vec{y}', r(y')\rangle, \text{ where } g\langle \vec{v}, \vec{y}, y\rangle = \langle \vec{v}', \vec{y}', y'\rangle \\ \quad \textbf{otherwise} : \vec{\bot} \\ \textbf{endcase} \end{cases}$$

Of course: This requires a **case** statement and recursion.

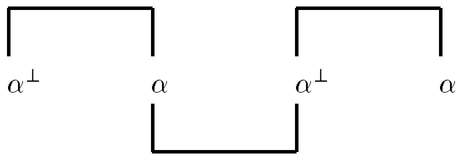
$$!(f)\langle \vec{u}, \vec{z}, z\rangle = \begin{cases} \textbf{case } e^\star(z) \textbf{ of} \\ \quad \texttt{in}_1(x) : \langle \vec{u}', \vec{z}', d(x')\rangle, \text{ where } f\langle \vec{u}, \vec{z}, x\rangle = \langle \vec{u}', \vec{z}', x'\rangle \\ \quad \texttt{in}_2 : \langle \vec{w}, \vec{w}, w\rangle \\ \quad \texttt{in}_3(x,y) : \langle c(\vec{s}', \vec{t}'), c(\vec{x}', \vec{y}'), c(x', y')\rangle \\ \quad \textbf{otherwise} : \vec{\bot} \\ \textbf{endcase} \end{cases}$$

where $!(f)\langle \vec{s}, \vec{x}, x\rangle = \langle \vec{s}', \vec{x}', x'\rangle,\ !(f)\langle \vec{t}, \vec{y}, y\rangle = \langle \vec{t}', \vec{y}', y'\rangle,\ \vec{c}^\star(\vec{u}) = \langle \vec{s}, \vec{t}\rangle,\ \vec{c}^\star(\vec{z}) = \langle \vec{x}, \vec{y}\rangle$.

4.3 Examples of Computations

As a prelude to the detailed verification of the soundness of the interpretation in Section 6, we will illustrate how it works in a number of key cases. We will perform our calculations in $W^\infty_\bot(\Sigma, X)$ defined earlier. The advantage of being able to trace the dynamics through specific calculations with concrete data structures, both as regards comprehensibility and as a basis for implementations, should be apparent.

Axiom contraction



The fixpoint computation is as follows.

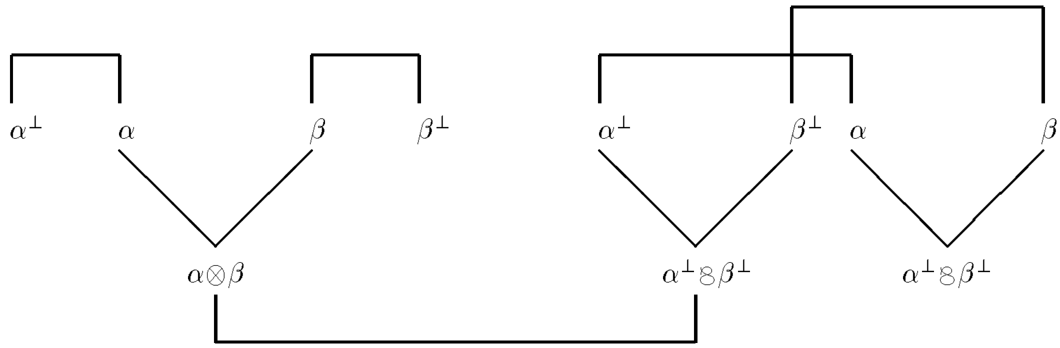
Iteration	Values
0	$\langle \perp, \perp, x, y \rangle$
1	$\langle y, x, \perp, \perp \rangle$
2	$\langle y, x, y, x \rangle$

So, $\langle x, y \rangle \mapsto \langle y, x \rangle$. In more detail, we have

$$\begin{aligned} \sigma \circ f\langle \perp, \perp, x, y \rangle &= \langle y, x, \perp, \perp \rangle \\ \sigma \circ f\langle y, x, x, y \rangle &= \langle y, x, y, x \rangle \end{aligned}$$

Note that a “fresh copy” of the parameters x, y is used at each iteration. This is in fact the key difference between Girard’s execution formula and our feedback formula.

Multiplicative contraction

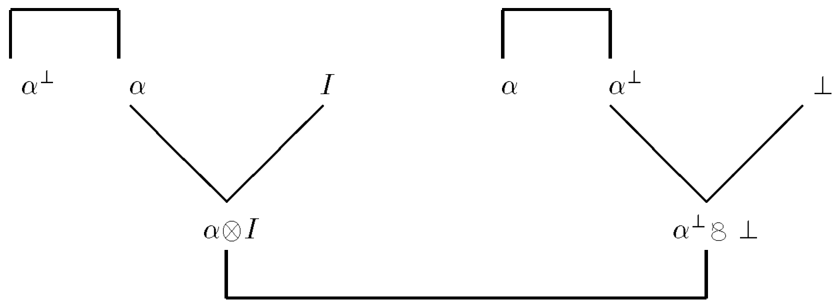


The fixpoint computation is as follows.

Iteration	Values
0	$\langle \perp, \perp, x, y, m(u, v) \rangle$
1	$\langle m(\perp, \perp), m(\perp, \perp), \perp, \perp, m(\perp, \perp) \rangle$
2	$\langle m(u, v), m(x, y), \perp, \perp, m(\perp, \perp) \rangle$
3	$\langle m(u, v), m(x, y), u, v, m(x, y) \rangle$

So, $\langle x, y, m(u, v) \rangle \mapsto \langle u, v, m(x, y) \rangle$.

Multiplicative units

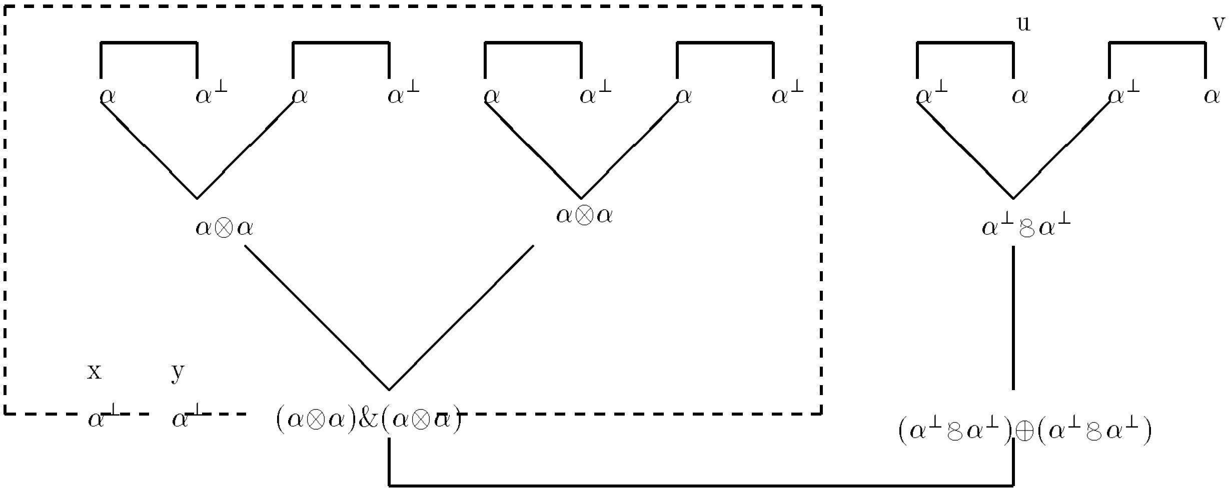


The fixpoint computation is as follows.

Iteration	Values
0	$\langle \perp, \perp, x, y \rangle$
1	$\langle m(\perp, u), m(\perp, u), \perp, \perp \rangle$
2	$\langle m(y, u), m(x, u), \perp, \perp \rangle$
3	$\langle m(y, u), m(x, u), y, x \rangle$

So, $\langle x, y \rangle \mapsto \langle y, x \rangle$.

Additives



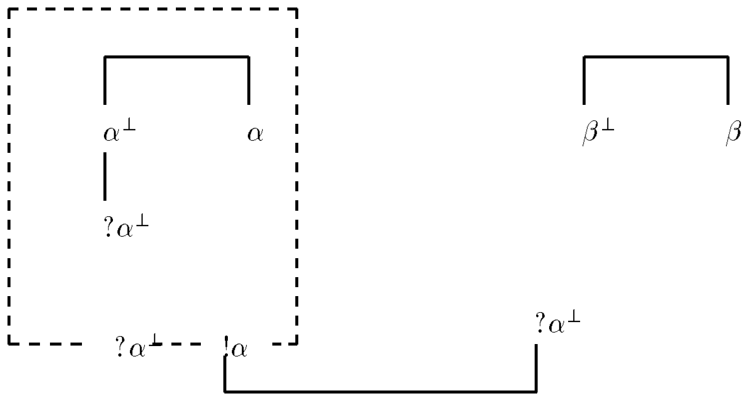
The fixpoint computation is as follows.

Iteration	Values
0	$\langle \perp, \perp, x, y, u, v \rangle$
1	$\langle l(m(\perp, \perp)), \perp, \perp, \perp, \perp, \perp \rangle$
2	$\langle l(m(\perp, \perp)), l(m(x, y), \perp, \perp, \perp, \perp \rangle$
3	$\langle l(m(u, v)), l(m(x, y), \perp, \perp, x, y \rangle$
4	$\langle l(m(u, v)), l(m(x, y), u, , v, x, y \rangle$

So, $\langle x, y, u, v\rangle \mapsto \langle u, v, x, y\rangle$.

Note how synchronisation occurs in steps 1 and 2; firstly the **Plus Left** side of the communication produces some information: the partial tree $l(m(\perp, \perp))$. This is transmitted by the permutation σ to the **With** side, which is able to proceed on the next iteration.

Exponentials



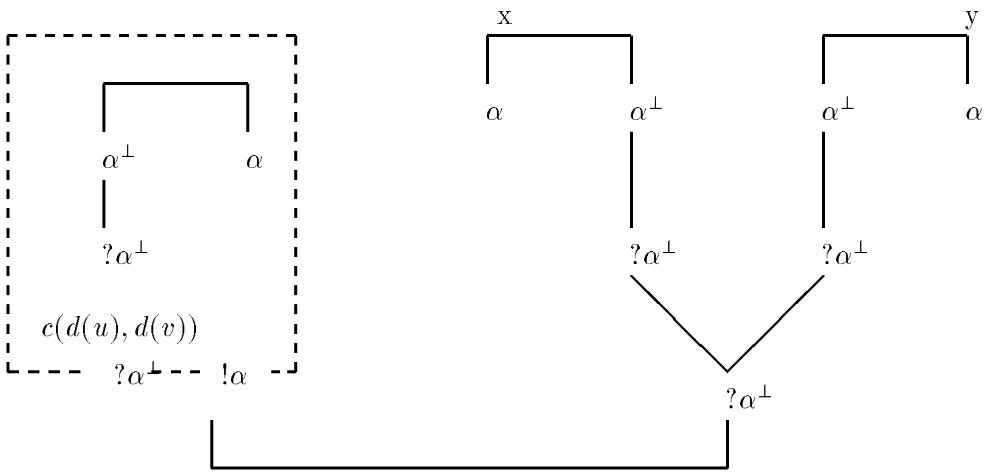
Weakening:

The fixpoint computation is as follows.

Iteration	Values
0	$\langle \perp, \perp, d(x), y, z\rangle$
1	$\langle w, \perp, d(\perp), z, y\rangle$
2	$\langle w, w, w, z, y\rangle$

So, $\langle d(x), y, z\rangle \mapsto \langle w, z, y\rangle$. This example illustrates how synchronisation affects the context of an `Ofcourse`. This is a typical case where Girard's interpretation does not fit the proof theory.

Contraction and Dereliction: The fixpoint computation is as follows.



Iteration	Values
0	$\langle \perp, \perp, c(d(u), d(v)), x, y \rangle$
1	$\langle c(d(\perp), d(\perp)), \perp, c(d(\perp), d(\perp)), \perp, \perp \rangle$
2	$\langle c(d(\perp), d(\perp)), c(d(u), d(v)), c(d(\perp), d(\perp), \perp, \perp \rangle$
3	$\langle c(d(x), d(y)), c(d(u), d(v)), c(d(\perp), d(\perp), u, v \rangle$
4	$\langle c(d(x), d(y)), c(d(u), d(v)), c(d(x), d(y), u, v \rangle$

So $\langle c(d(u), d(v)), x, y \rangle \mapsto \langle c(d(x), d(y)), u, v \rangle$.

5 $\mathcal{GI}$ as Dataflow

In this section, we outline a computational interpretation of the Geometry of Interaction in terms of the dataflow model of computation [Den74, AKP80]. We include this material to reinforce the reader's computational intuitions.

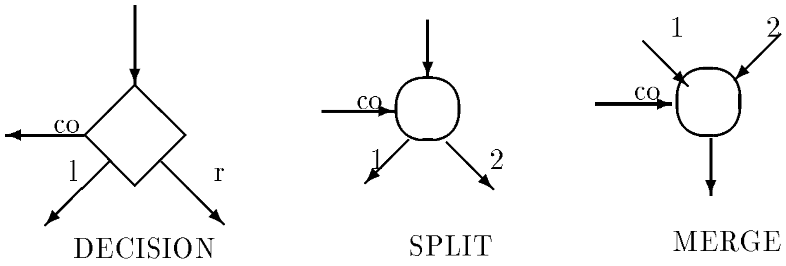
Dataflow graphs consist of *directed graphs*. The structure of these graphs does not change during the course of computation. Computation consists of the circulation and transformation of information tokens around the graph. The nodes are labelled by instructions. They do not have any memory associated with them; they are history insensitive. The behaviour of a node is given by a firing rule, which describes how nodes can "fire", removing some tokens from the input arcs and generating tokens on the output arcs. The firing of nodes proceeds asynchronously and in parallel. Output tokens are generated *asynchronously*, and queue up at their destination nodes. There is no need to assume that these queues work in any particular order. The interested reader is referred to [Den74, AKP80, Kah77, KM77] for a more detailed description of dataflow networks.

Executing the Geometry of Interaction

The tokens that we will use are finite elements of $W_\perp^\infty(\Sigma, \varnothing)$. The interpretation of the axiom is simple: there are a pair of nodes that merely transfer tokens from their input arcs to the output arcs. Cut is interpreted by feedback as indicated earlier.

We have a pair of nodes $r+, r-$for every operator r in Σ. $r+$ has k inputs and one output, and $r-$ has one input and k outputs, where k is the arity of r. This $r+$ node fires whenever it receives a token on any channel. For example, on receipt of a token v_s on input arc s, the node outputs the token $r(\perp, \ldots, v_s, \ldots, \perp)$. The node $r-$ fires when it receives a token on the input. If the input is of form $r(t_1, \ldots, t_k)$, for each output arc i, it outputs t_i if $t_i \neq \perp$.

Now, we have enough to interpret all the combinators except $\&, !$. Implementing the $\mathcal{GI}$ interpretation of $\&, !$ requires some mechanisms for *local synchronisation*. We describe a set of three nodes each for $a^\star, e^\star$. Various forms of these nodes have been used in dataflow graphs used as intermediate representations for compilers. We first describe the set of three nodes for $a^\star$. The *decision* node fires on receipt of an input token. If this token has the form $l(t)$ it sends t to its left output, and the "control" token 1 on the arc labelled `co`; if this token has form $r(t)$ it sends t to its right output, and the token 2 on the arc labelled `co`. For any other token, it sends 0 on `co`.



The *split* node fires on receipt of a tokens on both input lines. The horizontal line carries a control token generated by a decision node, and selects one of the k output lines for despatching the token received on the vertical input line. If the control line carries a 0, no token is despatched.

The firing rules of the *merge* node is dual to the Split node. The horizontal line carries a control token generated by a decision node, and selects one of the k input lines. If the token is non-zero, the node waits for a token on the selected input line, and despatches the token on the sole output line.

The set of three nodes for $e^{\star}$ behaves similarly with respect to the tokens of form $\mathtt{d}(t), \mathtt{w}, \mathtt{c}(t_1, t_2)$. Roughly speaking, the proof boxes for $\&, !$ gets converted to a collection of decision-switch-merge combinations in the dataflow interpretation.

6 Soundness of the interpretation

Our objective in this section is to prove the soundness of the Geometry of Interaction interpretation.

6.1 Overview of results

Firstly, some notation. We fix a $\mathcal{GI}$ interpretation based on a model D. If Π is a proof of a sequent $\vdash [\Delta],\ \Gamma$ in $\mathtt{CLL}_2$, we write $[\![\Pi]\!] = f$ for the function $f : D^{2m+n} \to D^{2m+n}$ assigned to Π in the $\mathcal{GI}$ interpretation, and $\sigma_f : D^{2m} \to D^{2m}$ for the corresponding "message exchange" function.

Soundness Theorem: Let Π be a proof of a sequent $\vdash [\Delta],\ \Gamma$ in $\mathtt{CLL}_2$, with $[\![\Pi]\!] = f$. Then:

1. If there are no occurrences of & either explicitly in Γ or in any of the witnessing formulas used in the *Exists* rule to introduce occurrences of $\exists$ in Γ, then if Π reduces to Π' by any sequence of contractions, with $[\![\Pi']\!] = g$, then $FB(f, \sigma_f) = FB(g, \sigma_g)$. In particular, if Π' is any cut-free proof obtained from Π by cut-elimination, then $FB(f, \sigma_f) = g$.

2. f has the *finite convergence* property: $(\exists k)\,[f^{(k)} = f^{(k+1)} = FB(f, \sigma_f)]$.

Comparing our results with Girard's, we note that for the fragment he considers, or the larger fragment with multiplicative units included, we *do* get full correspondence with cut-elimination.

In terms of the dataflow interpretation, finite convergence means that for any input, the network is guaranteed to eventually become *quiescent*, with no tokens remaining in circulation. The value computed on an output line will then be given by the join of all the tokens which have been despatched on that output line.

The proof in fact establishes tight connections between reduction steps on proofs and iterations to the fixpoint in the $\mathcal{GI}$ interpretation. Call the the number of iterations required to reach the fixpoint the *index of convergence*. If $\Pi \rightarrow \Pi'$ by the commutative conversion $!(f) \cdot !(h) = !(f \cdot !(h))$ (a cut with a formula in the context of the *Of Course* is intended), the index of convergence of Π will be (at most) one more than that of Π. If $\Pi \rightarrow \Pi'$ by any other commutative conversion, the index of convergence of Π will equal that of Π. If $\Pi \rightarrow \Pi'$ by a symmetric contraction, the index of convergence for Π will be (at most) one or two more than that of Π': one in the case of the multiplicatives, where both sides of the the cut can proceed without waiting for information from the environment; two for the additives and the exponentials, where synchronisation does take place, one side $(\&, !)$ waiting for information; its partner $(\oplus, ?)$ generating information immediately without waiting. Two steps are also needed for axiom contraction, one for information to flow (bidirectionally) into the buffer, one for it to flow out.

The reason for the restriction on part 1 is that the following commutative conversion is not valid under our interpretation: $\&(f, g) \cdot h = \&(f \cdot h, g \cdot h)$ (we are omitting exchanges here; a cut with a formula in the context of the *With* is intended). In fact, these two functions are *equal* for all arguments where the component corresponding to the *With* is of the form $\mathtt{l}(t)$ or $\mathtt{r}(u)$. Intuitively, for arguments of the right "shape", the interpretation is well behaved. These ideas are developed in section 7.

6.2 Linear Realizability Algebras

As a preliminary to proving soundness, we review the formalism of Linear Realizability Algebras, introduced by the first author [Abr91b]. This provides a very convenient framework for proving soundness and allows some general lemmas to be factored out.

Syntax

We assume an infinite set of *names* $\mathcal{N}$ ranged over by α, β, γ. Names can be thought of as ports or channels as in various process formalisms; the closest analogy is in fact with names as used in the π-calculus [MPW89, Mil91]. A *sort* is a finite subset of $\mathcal{N}$; we use $X, Y, \ldots$ to range over sorts. A *renaming* is a bijection between sorts.

Next, we introduce the idea of *located sequents*, of the form

$$\vdash \alpha_1 : A_1, \ldots, \alpha_k : A_k$$

Proof Rule	Operation	Constraint	Sort
Axiom	$I_{\alpha,\beta}$		$\{\alpha, \beta\}$
Cut	$P \cdot_{\alpha} Q$	$\mathtt{FN}(P) \cap \mathtt{FN}(Q) = \{\alpha\}$	$\mathtt{FN}(P) \cup \mathtt{FN}(Q) \setminus \{\alpha\}$
Tensor Unit	U_{α}		$\{\alpha\}$
Perp	$\perp_{\alpha} (P)$	$\alpha \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \cup \{\alpha\}$
Times	$\otimes_{\gamma}^{\alpha,\beta}(P, Q)$	$\alpha \in \mathtt{FN}(P), \beta \in \mathtt{FN}(Q)$ $\mathtt{FN}(P) \cap \mathtt{FN}(Q) = \varnothing$ $\gamma \notin \mathtt{FN}(P) \cup \mathtt{FN}(Q)$	$\mathtt{FN}(P) \cup \mathtt{FN}(Q) \setminus \{\alpha, \beta\} \cup \{\gamma\}$
Par	$⅋_{\gamma}^{\alpha,\beta}(P)$	$\alpha, \beta \in \mathtt{FN}(P)$ $\alpha \neq \beta$ $\gamma \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \setminus \{\alpha, \beta\} \cup \{\gamma\}$
Plus Left	$L_{\gamma}^{\alpha}(P)$	$\alpha \in \mathtt{FN}(P), \gamma \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \setminus \{\alpha\} \cup \{\gamma\}$
Plus Right	$R_{\gamma}^{\alpha}(P)$	$\alpha \in \mathtt{FN}(P), \gamma \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \setminus \{\alpha\} \cup \{\gamma\}$
Top	$\top_{\vec{\alpha},\alpha}$		$\{\vec{\alpha}, \alpha\}$
With	$\&_{\gamma}^{\alpha,\beta}(P, Q)$	$\alpha \in \mathtt{FN}(P), \beta \in \mathtt{FN}(Q)$ $\mathtt{FN}(P) \setminus \{\alpha, \beta\} = \mathtt{FN}(Q) \setminus \{\beta\}$	$\mathtt{FN}(P) \setminus \{\alpha\} \cup \{\gamma\}$
Dereliction	$D_{\gamma}^{\alpha}(P)$	$\alpha \in \mathtt{FN}(P), \gamma \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \setminus \{\alpha\} \cup \{\gamma\}$
Weakening	$W_{\gamma}(P)$	$\gamma \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \cup \{\gamma\}$
Contraction	$C_{\gamma}^{\alpha,\beta}(P)$	$\alpha, \beta \in \mathtt{FN}(P)$ $\alpha \neq \beta$ $\gamma \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \setminus \{\alpha, \beta\} \cup \{\gamma\}$
Of course	$!_{\gamma}^{\alpha}(P)$	$\alpha \in \mathtt{FN}(P), \gamma \notin \mathtt{FN}(P)$	$\mathtt{FN}(P) \setminus \{\alpha\} \cup \{\gamma\}$

Figure 1: Syntax: Linear Realizability Algebra

where the α_i are distinct names, and the A_i are formulas of $\mathtt{CLL}_2$. These sequents are to understood as *unordered*, *i.e.* as functions from $\{\alpha_1, \ldots, \alpha_k\}$—the sort of the sequent—to the set of $\mathtt{CLL}_2$ formulae.

The terms are described in Figure 1. We use P, Q, R to range over these terms, and write $\mathtt{FN}(P)$ for the set of names occurring freely in P—its *sort*. With each term-forming operation we give a linearity constraint on how it can be applied, and specify its sort. There is an evident notion of α-conversion $P \equiv_{\alpha} Q$, and renaming is written $P[\beta/\alpha]$.

Terms are assigned to sequent proofs in $\mathtt{CLL}_2$ as in Figure 2.

Dynamics

We now describe the "dynamics" of terms, corresponding to cut-elimination of proofs. We factor this into two parts, in the style of [Mil90, Abr93]: a structural congruence $\equiv$ and a reduction relation $\rightarrow$.

Identity Group	$\dfrac{}{I_{\alpha,\beta} \vdash \alpha : A^{\perp}, \beta : A}$	$\dfrac{P \vdash \Gamma, \alpha : A \qquad Q \vdash \Delta, \alpha : A^{\perp}}{P \cdot_{\alpha} Q \vdash \Gamma, \Delta}$
Multiplicative Units	$\dfrac{}{U_{\alpha} \vdash \alpha : I}$	$\dfrac{P \vdash \Gamma}{\perp_{\alpha} \ \vdash \alpha : \perp, \Gamma}$
Multiplicatives	$\dfrac{P \vdash \Gamma, \alpha : A \qquad Q \vdash \Delta, \beta : B}{\otimes^{\alpha,\beta}_{\gamma}(P, Q) \vdash \Gamma, \Delta, \gamma : A \otimes B}$	$\dfrac{P \vdash \Gamma, \alpha : A, \beta : B}{⅋^{\alpha,\beta}_{\gamma}(P) \vdash \Gamma, \gamma : A ⅋ B}$
Additive Unit	$\dfrac{}{\top_{\vec{\alpha},\alpha} \vdash \vec{\alpha} : \Gamma, \alpha : \top}$	
Additives	$\dfrac{P \vdash \Gamma, \alpha : A}{L^{\alpha}_{\gamma}(P) \vdash \Gamma, \gamma : A \oplus B}$ $\dfrac{P \vdash \Gamma, \alpha : B}{R^{\alpha}_{\gamma}(P) \vdash \Gamma, \gamma : A \oplus B}$	$\dfrac{P \vdash \Gamma, \alpha : A \qquad Q \vdash \Gamma, \beta : B}{\&(P, Q) \vdash \Gamma, \gamma : A \& B}$
Exponentials	$\dfrac{P \vdash \Gamma, \alpha : A}{D^{\alpha}_{\gamma}(P) \vdash \Gamma, \gamma : ?A}$ $\dfrac{P \vdash \Gamma}{W_{\gamma}(P) \vdash \Gamma, \gamma : ?A}$ $\dfrac{P \vdash \Gamma, \alpha : ?A, \beta : ?A}{C^{\alpha,\beta}_{\gamma}(P) \vdash \Gamma, \gamma : ?A}$	$\dfrac{P \vdash ?\Gamma, \alpha : A}{!^{\alpha}_{\gamma}(P) \vdash ?\Gamma, \gamma : !A}$
Quantifiers	$\dfrac{P \vdash \Gamma, \alpha : A}{P \vdash \Gamma, \alpha : \forall X.A}$, X not free in Γ	$\dfrac{P \vdash \Gamma, \alpha : A[B/X]}{P \vdash \Gamma, \alpha : \exists X.A}$

Figure 2: Realizability semantics

The structural congruence is the least congruence $\equiv$ on terms such that:

(SC1) $P \equiv_\alpha Q \Rightarrow P \equiv Q$

(SC2) $P \cdot_\alpha Q \equiv Q \cdot_\alpha P$

(SC3) $\omega(P_1, \ldots, P_k) \equiv \omega(P_1, \ldots, P_i \cdot_\alpha Q, \ldots, P_k)$, if $\alpha \in \mathtt{FN}(P_i)$, $\omega \notin \{\&, !\}$.

We give some examples to illustrate the third rule (SC3):

1. $(P \cdot_\alpha Q) \cdot_\beta R \equiv P \cdot_\alpha (Q \cdot_\beta R)$, if $\beta \in \mathtt{FN}(Q)$.
2. $L^\alpha_\gamma(P) \cdot_\beta Q \equiv L^\alpha_\gamma(P \cdot_\beta Q)$, if $\beta \in \mathtt{FN}(P)$.
3. $\otimes^{\alpha,\beta}_\gamma(P, Q) \cdot_\delta R \equiv \otimes^{\alpha,\beta}_\gamma(P, Q \cdot_\delta R)$, if $\delta \in \mathtt{FN}(Q)$.

Note that by linearity constraints, for example $L^\alpha_\gamma(P) \cdot_\beta Q$ well-formed; thus $\beta \in \mathtt{FN}(L^\alpha_\gamma(P))$, and hence $\beta \neq \alpha$; also, $\beta \in \mathtt{FN}(P)$ implies that $\beta \neq \gamma$.

The reductions are as follows:

(R1) $P \cdot_\alpha I_{\alpha,\beta} \rightarrow P[\beta/\alpha]$.

(R2) $\perp_\alpha (P) \cdot_\alpha U_\alpha \rightarrow P$.

(R3) $⅋^{\alpha,\beta}_\gamma(P) \cdot_\gamma \otimes^{\alpha,\beta}_\gamma(Q, R) \rightarrow P \cdot_\alpha Q \cdot_\beta R$.

(R4) $L^\alpha_\gamma(P) \cdot_\gamma \&^{\alpha,\beta}_\gamma(Q, R) \rightarrow P \cdot_\alpha Q$.

(R5) $R^\alpha_\gamma(P) \cdot_\gamma \&^{\alpha,\beta}_\gamma(Q, R) \rightarrow P \cdot_\alpha R$.

(R6) $D^\alpha_\gamma(P) \cdot_\gamma !^\alpha_\gamma(Q) \rightarrow P \cdot_\alpha Q$.

(R7) $W_\gamma(P) \cdot_\gamma !^\alpha_\gamma(Q) \rightarrow W_{\vec{\alpha}}(P)$, where $\mathtt{FN}(Q) \setminus \{\alpha\} = \vec{\alpha}$.

(R8) $C^{\gamma',\gamma''}_\gamma(P) \cdot_\gamma !^\alpha_\gamma(Q) \rightarrow C^{\vec{\alpha}',\vec{\alpha}''}_{\vec{\alpha}}(P \cdot_\gamma {}'!^\alpha_{\gamma'}(Q[\vec{\alpha'}/\vec{\alpha}]) \cdot_\gamma {}''!^\alpha_{\gamma''}(Q[\vec{\alpha}''/\vec{\alpha}]))$, where $\mathtt{FN}(Q) \setminus \{\alpha\} = \vec{\alpha}$.

(R9) $!^\alpha_\gamma(P) \cdot_\delta !^\beta_\delta(Q) \rightarrow !^\alpha_\gamma(P \cdot_\delta !^\beta_\delta(Q))$, if $\delta \in \mathtt{FN}(P)$.

These reductions can be applied in any context.

$$\frac{P \rightarrow Q}{C[P] \rightarrow C[Q]}$$

and are performed modulo structural congruence.

$$\frac{P' \equiv P \quad P \rightarrow Q \quad Q' \equiv Q}{P \rightarrow Q}$$

Note that we do *not* have the reduction rules

(R10) $\&_{\gamma}^{\alpha,\beta}(P,Q) \cdot_{\delta} R \rightarrow \&_{\gamma}^{\alpha,\beta}(P \cdot_{\delta} R, Q \cdot_{\delta} R)$ $[\delta \in \mathtt{FN}(P) \cap \mathtt{FN}(Q)]$

(R11) $T_{\vec{\alpha},\alpha} \cdot_{\beta} P \rightarrow T_{\vec{\alpha},\vec{\beta},\alpha}$ $[\beta \in \vec{\alpha}, \mathtt{FN}(P) = \vec{\beta},\beta]$

which would allow us to reduce cuts on formulae occurring in the context of a *With*. The reason for omitting these rules is that they are not sound in many useful interpretations, including the Geometry of Interaction. In fact for the Geometry of Interaction, this is exactly the point already made in Section 3, that the interpretation of & in $\mathcal{GI}(\mathbb{C})$ does not quite yield the categorical product. In general, normal forms for terms under reduction will correspond to *canonical* rather than cut-free proofs; *cf* [Abr93].

Algebraic presentation

We now introduce the notion of Linear Realizability Algebra LRA, *i.e.* the structure which interprets this term calculus.

An LRA provides, for each sort $\vec{\alpha}$, a set $A_{\vec{\alpha}}$, together with, for each $1-1$ renaming $[\vec{\beta}/\vec{\alpha}]$, a bijection: $=_{[\vec{\beta}/\vec{\alpha}]}$ subject to the obvious functorial conditions.

For each syntactic term-forming operation, there is a corresponding family of functions, *e.g.* $\otimes_{\gamma}^{\alpha,\beta}$ is interpreted by a family of functions:

$$A_{\vec{\alpha},\alpha} \times A_{\vec{\beta},\beta} \rightarrow A_{\vec{\alpha},\vec{\beta},\gamma}$$

for each $\vec{\alpha},\vec{\beta},\alpha,\beta,\gamma$ satisfying the linearity constraints as in the formation rule for $\otimes_{\gamma}^{\alpha,\beta}(P,Q)$, where $\vec{\alpha},\alpha = \mathtt{FN}(P)$ and $\vec{\beta},\beta = \mathtt{FN}(Q)$. Moreover, this family satisfies naturality conditions which ensures that it behaves smoothly with respect to renaming.

For a more precise definition, it will be convenient to describe Linear Realizability Algebras as structures in $\mathtt{Set}^{\mathtt{G}}$, where $\mathtt{G}$ is the groupoid of sorts and renamings. The *carrier* of the set will be a "set" in $\mathtt{Set}^{\mathtt{G}}$, *i.e.* a functor $A : \mathtt{G} \rightarrow \mathtt{Set}$. The operations are certain natural transformations, described as follows.

Let B be the category with objects (X,Y,α) where $X \cap Y = \emptyset$, $\alpha \notin X \cup Y$, and morphisms $(f,g) : (X,Y,\alpha) \rightarrow (X',Y',\alpha)$, where $f : X \xrightarrow{\sim} X'$, $g : Y \xrightarrow{\sim} Y'$. Define functors $F,G : B \rightarrow \mathtt{Set}$ by:

$$\begin{aligned} F(X,Y,\alpha) &= A(X,\alpha) \times A(Y,\alpha) \\ G(X,Y,\alpha) &= A(X,Y) \end{aligned}$$

where we write X,Y for $X \cup Y$. Then, the interpretation of the composition is a natural transformations $F \rightarrow G$. More succinctly, we can write

$$\cdot_{\alpha} \; : X,\alpha \times Y,\alpha \rightarrow X,Y$$

Continuing this succinct notation, we write

$$
\begin{array}{rcl}
I_{\alpha,\beta} & : & \rightarrow \alpha, \beta \\
\otimes_{\gamma}^{\alpha,\beta} & : & X, \alpha \times Y, \beta \rightarrow X, Y, \gamma \\
⅋_{\gamma}^{\alpha,\beta} & : & X, \alpha, \beta \rightarrow X, \gamma \\
1_{\alpha} & : & \rightarrow \alpha \\
\bot_{\alpha} & : & X \rightarrow X, \alpha \\
\&_{\gamma}^{\alpha,\beta} & : & X, \alpha \times X, \beta \rightarrow X, \gamma \\
L_{\gamma}^{\alpha} & : & X, \alpha \rightarrow X, \gamma \\
R_{\gamma}^{\alpha} & : & X, \alpha \rightarrow X, \gamma \\
\top_{X,\gamma} & : & \rightarrow X, \gamma \\
!_{\gamma}^{\alpha} & : & X, \alpha \rightarrow X, \gamma \\
D_{\gamma}^{\alpha} & : & X, \alpha \rightarrow X, \gamma \\
W_{\gamma} & : & X \rightarrow X, \gamma \\
C_{\gamma}^{\alpha,\beta} & : & X, \alpha, \beta \rightarrow X, \gamma
\end{array}
$$

to indicate the "types" of the corresponding natural transformations, which will interpret these operations. Note that the operations are parametrised by the names they bind and introduce.

Thus, $\mathcal{A}$ induces a semantic function $[\![\cdot]\!]$ which maps for each term P of sort $\vec{\alpha}$ to $[\![P]\!] \in A_{\vec{\alpha}}$. The following further data must be provided:

- A family $\approx = \{\approx_{\alpha}\}$ of equivalence relations on $A_{\vec{\alpha}}$ for each sort $\vec{\alpha}$. This relation should be thought of as "observational congruence" or "extensional equality".
- A family $\Downarrow = \{\Downarrow_{\alpha}\}$ of predicates (subsets) for each sort $\vec{\alpha}$. This predicate should be thought of as "convergence".

The data are required to satisfy the following conditions:

	Hypothesis	Conditions	
P0	$[\![P]\!] \approx [\![Q]\!]$	$[\![C[P]]\!] \approx [\![C[Q]]\!]$	
P1	$P \equiv Q$	$[\![P]\!] = [\![Q]\!]$	
P2	$P \rightarrow Q$	$[\![P]\!] \approx [\![Q]\!]$	$Q\Downarrow \Rightarrow P\Downarrow$
P3	ω a constructor (any operator except Cut)	$\dfrac{P_1\Downarrow, \ldots, P_k\Downarrow}{\omega(P_1, \ldots, P_k)\Downarrow}$	
P4	$P \cdot_{\alpha} I_{\alpha,\beta}\Downarrow$	$P\Downarrow$	

Realizability

This section shows that LRA's isolate the essence of what is required to provide a sound interpretation of cut-elimination in Linear Logic, including the essence of Strong Normalisation. In particular, the second part of Theorem 2 proved below specialises to yield Cut-elimination for System F, via the interpretation into $\mathtt{CLL}_2$, and hence is by no means a trivial result. Our proof follows much the same lines as [Abr93].

Firstly, some notation. Given terms $P, Q \in A$, we define

$$\mathtt{Cut}(P, Q) = P \cdot_\alpha Q$$

More precisely, we choose renamed versions P' of P, Q' of Q such that $\mathtt{FN}(P') \cap \mathtt{FN}(Q') = \{\alpha\}$, and form $P' \cdot_\alpha Q'$ (compare the definition of substitution in [Bar84]); we will generally take this renaming for granted, and not refer to it explicitly. Now we define

$$P \perp Q \Longleftrightarrow \mathtt{Cut}(P, Q) \Downarrow .$$

(This definition is easily seen to be independent of the choice of P', Q'.) Given $U \subseteq A$, we can define

$$U^\perp = \{P \in A \mid \forall Q \in U. (P \perp Q)\}.$$

Now by standard facts about Galois connections [Coh81], we have:

Proposition 6

(i) *The operator* $(\cdot)^{\perp\perp}$ *is monotone, inflationary and idempotent.*

(ii) $U^{\perp\perp\perp} = U^\perp$.

(iii) $\forall P \in A, U \subseteq A. (P \perp U \Longleftrightarrow P \perp U^{\perp\perp})$.

A *semantic type* is a subset $U \subseteq A$ satisfying:

- $I_{\alpha,\beta} \in U$.
- $U \Downarrow$ *i.e.* $\forall P \in U. (P \Downarrow)$.
- $U = U^{\perp\perp}$

We write $\mathcal{U}$ for the set of all semantic types.

Lemma 1 *For all* $U \subseteq A$:

(i) $[\![I_{\alpha,\beta}]\!] \in U \Rightarrow U^{\perp} \Downarrow$

(ii) $U \Downarrow \Rightarrow [\![I_{\alpha,\beta}]\!] \in U^{\perp}$.

(iii) U is closed under renaming.

Proof:

(iii) For any $U \subseteq A$, it is immediate from definitions that $U^{\perp}$ is closed under renaming.

(i) If $Q \in U^{\perp}$, then $\texttt{Cut}(I_{\alpha,\beta}, Q) \rightarrow Q[\beta/\alpha]$ by **(R1)**. Since $Q[\beta/\alpha] \Downarrow$, Using **(P2)**, $\texttt{Cut}(I_{\alpha,\beta}, Q) \Downarrow$.

(ii) Let $P \in U$. Then $\texttt{Cut}(P, I_{\alpha,\beta}) \rightarrow P$. Since $P \Downarrow$, using **(P2)**, $\texttt{Cut}(P, I_{\alpha,\beta}) \Downarrow$ as required. ∎

Proposition 7 *If $I_{\alpha,\beta} \in U \subseteq A$ satisfies $U \Downarrow$, then $U^{\perp} \in \mathcal{U}$.*

Proof: By Lemma 1 and Proposition 6(ii). ∎

We will now give a realizability interpretation of the Linear types as elements of $\mathcal{U}$.

$$\begin{array}{rcl}
\perp & = & \{U_{\alpha}\}^{\perp} \\
U \parr V & = & \{\otimes_{\gamma}^{\alpha,\beta}(P,Q) \in A \mid P \in U^{\perp}, Q \in V^{\perp}\}^{\perp} \\
U \,\&\, V & = & (\{L_{\gamma}^{\alpha}(P) \in A \mid P \in U^{\perp}\} \cup \{R_{\gamma}^{\alpha}(P) \in A \mid Q \in V^{\perp}\})^{\perp} \\
?U & = & \{!_{\gamma}^{\alpha}(P) \in A \mid P \in U^{\perp}\}^{\perp}
\end{array}$$

and for $F : \mathcal{U} \rightarrow \mathcal{U}$

$$\forall(F) = (\bigcup\{F(U)^{\perp} \mid U \in \mathcal{U}\})^{\perp}$$

By Propositions 7 and 6 and the remarks immediately preceding these definitions, they do yield semantic types. The remaining connectives are defined by duality.

$$\begin{array}{rcl}
\mathbf{1} & = & \perp^{\perp} \\
U \otimes V & = & (U^{\perp} \parr V^{\perp})^{\perp} \\
U \oplus V & = & (U^{\perp} \,\&\, V^{\perp})^{\perp} \\
!U & = & (\,?U^{\perp})^{\perp} \\
\exists(F) & = & \forall(\lambda U.F(U)^{\perp})^{\perp}
\end{array}$$

These definitions induce a semantic function

$$[\![\cdot]\!] : \mathsf{TExp} \rightarrow \mathsf{TEnv} \rightarrow \mathcal{U}$$

where TExp is the set of Linear type expressions, *i.e.* formulae of CLL, and $\mathsf{TEnv} = \mathsf{TVar} \rightarrow \mathcal{U}$ is the set of type environments, ranged over by η.

Lemma 2 *For all $A \in \mathsf{TExp}$, $\eta \in \mathsf{TEnv}$: $([\![A]\!]\eta)^{\perp} = [\![A^{\perp}]\!]\eta$.*

We can now give a realizability interpretation of $\mathtt{CLL}_2$ sequents. We write $PQ_1 \cdots Q_k$ to abbreviate $\mathtt{Cut}(\ldots\mathtt{Cut}((\mathtt{Cut}(P, Q_1)), P_2)\ldots, Q_k)$. Define

$$P \models \Gamma \Longleftrightarrow \forall \eta \in \mathsf{TEnv}, \bar{P} \in [\![\Gamma^\perp]\!]\eta.\,(P\bar{P}\Downarrow)$$

where $\Gamma = A_1, \ldots, A_k$, $[\![\Gamma^\perp]\!]\eta = [\![{A_1}^\perp]\!]\eta, \ldots, [\![{A_k}^\perp]\!]\eta$.

Theorem 1 (Realizability) $P \vdash \Gamma \Rightarrow P \models \Gamma$.

Proof: By induction on derivations in $\mathtt{CLL}_2$.

(1) Axiom:

$$\frac{}{I_{\alpha,\beta} \vdash \alpha : A^\perp, \beta : A}$$

Fix $\eta \in \mathsf{TEnv}$, $P \in [\![A^{\perp\perp}]\!]\eta = ([\![A^\perp]\!]\eta)^\perp$, $Q \in [\![A^\perp]\!]\eta$. We must show that $I_{\alpha,\beta}PQ\Downarrow$. But $I_{\alpha,\beta}PQ \xrightarrow{*} \mathtt{Cut}(P, Q)$, and $\mathtt{Cut}(P, Q)\Downarrow$, since $P \perp Q$ by assumption. Hence by **(P2)**, $I_{\alpha,\beta}PQ\Downarrow$.

(2) Exchange: immediate.

(3) Cut:

$$\frac{P \vdash \Gamma, \alpha : A \quad Q \vdash \Delta, \alpha : A^\perp}{P \cdot_\alpha Q \vdash \Gamma, \Delta}$$

Fix $\eta \in \mathsf{TEnv}$, $\bar{P} \in [\![\Gamma^\perp]\!]\eta$, $\bar{Q} \in [\![\Delta^\perp]\!]\eta$. We must show that $\mathtt{Cut}(P, Q)\bar{P}\bar{Q}\Downarrow$. By induction hypothesis, for all $R \in [\![A^\perp]\!]\eta$, $P\bar{P}R\Downarrow$, and for all $S \in [\![A]\!]\eta$, $Q\bar{Q}S\Downarrow$. Hence $P\bar{P} \in ([\![A^\perp]\!]\eta)^\perp = [\![A]\!]\eta$, and $Q\bar{Q} \in ([\![A]\!]\eta)^\perp$, so $\mathtt{Cut}(P\bar{P}, Q\bar{Q})\Downarrow$. But $\mathtt{Cut}(P\bar{P}, Q\bar{Q}) \equiv \mathtt{Cut}(P, Q)\bar{P}\bar{Q}$, so $\mathtt{Cut}(P, Q)\bar{P}\bar{Q}\Downarrow$.

(4) Perp:

$$\frac{P \vdash \Gamma}{\perp_\alpha(P) \vdash \Gamma, \alpha : \perp}$$

Fix $\eta \in \mathsf{TEnv}$, $\bar{Q} \in [\![\Gamma^\perp]\!]\eta$. We must show that for all $Q \in [\![\perp^\perp]\!]\eta$, $\perp_\alpha(P)\bar{Q}Q\Downarrow$, *i.e.* that $\perp_\alpha(P)\bar{Q}) \perp \{U_\alpha\}^{\perp\perp}$. By Proposition 6(iii), it suffices to show that $\sigma(\perp_\alpha(P)\bar{Q}) \perp \{U_\alpha\}$, *i.e.* that $\perp_\alpha(P)\bar{Q}U_\alpha\Downarrow$. But $\perp_\alpha(P)\bar{Q}U_\alpha \rightarrow P\bar{Q}$, and by induction hypothesis $P\bar{Q}\Downarrow$, so by **(P2)**, $\perp_\alpha(P)\bar{Q}Q\Downarrow$.

(5) Unit:

$$\frac{}{U_\alpha \vdash \alpha : \mathbf{1}}$$

We must show that for all η, $P \in [\![\mathbf{1}^\perp]\!]\eta$, $U_\alpha P\Downarrow$. By Proposition 6(i), $(\cdot)^{\perp\perp}$ is inflationary, so $U_\alpha \in [\![\mathbf{1}]\!]\eta$, and $U_\alpha P = \mathtt{Cut}(U_\alpha, P)\Downarrow$.

(6) Par:

$$\frac{P \vdash \Gamma, \alpha : A, \beta : B}{\mathbin{\bindnasrepma}_{\gamma}^{\alpha,\beta}(P) \vdash \Gamma, \gamma : A \mathbin{\bindnasrepma} B}$$

Fix $\eta \in \mathsf{TEnv}$, $\bar{Q} \in [\![\Gamma^{\perp}]\!]\eta$. We must show that for all $Q \in [\![(A \mathbin{\bindnasrepma} B)^{\perp}]\!]\eta$, $\mathbin{\bindnasrepma}_{\gamma}^{\alpha,\beta}(P)\bar{Q}Q \Downarrow$, *i.e.* that $\mathbin{\bindnasrepma}_{\gamma}^{\alpha,\beta}(P)\bar{Q} \perp [\![(A \mathbin{\bindnasrepma} B)^{\perp}]\!]\eta$. Applying Proposition 6(iii) to the definition of $[\![(A \mathbin{\bindnasrepma} B)^{\perp}]\!]\eta$, we see that it is sufficient to consider Q of the form $\otimes_{\gamma}^{\alpha,\beta}(R, S)$, where $R \in [\![A^{\perp}]\!]\eta$, $S \in [\![B^{\perp}]\!]\eta$. But $\mathbin{\bindnasrepma}_{\gamma}^{\alpha,\beta}(P)\bar{Q}\otimes_{\gamma}^{\alpha,\beta}(R, S) \rightarrow P\bar{Q}RS$, and by the induction hypothesis $P\bar{Q}RS \Downarrow$, so by **(P2)**, $\mathbin{\bindnasrepma}_{\gamma}^{\alpha,\beta}(P)\bar{Q}\otimes_{\gamma}^{\alpha,\beta}(R, S) \Downarrow$.

(7) Times:

$$\frac{P \vdash \Gamma, \alpha : A \quad Q \vdash \Delta, \beta : B}{\otimes_{\gamma}^{\alpha,\beta}(P, Q) \vdash \Gamma, \Delta, \gamma : A \otimes B}$$

Fix $\eta \in \mathsf{TEnv}$,$\bar{P} \in [\![\Gamma^{\perp}]\!]\eta$, $\bar{Q} \in [\![\Delta^{\perp}]\!]\eta$. We must show that for all $R \in [\![(A \otimes B)^{\perp}]\!]\eta$, $\otimes_{\gamma}^{\alpha,\beta}(P, Q)\bar{P}\bar{Q}R \Downarrow$. By induction hypothesis, for all $S \in [\![A^{\perp}]\!]\eta$, $P\bar{P}S \Downarrow$, and for all $T \in [\![B^{\perp}]\!]\eta$, $Q\bar{Q}T \Downarrow$. Hence $P\bar{P} \in ([\![A^{\perp}]\!]\eta)^{\perp} = [\![A]\!]\eta$, and $Q\bar{Q} \in ([\![B^{\perp}]\!]\eta)^{\perp}$. Applying Proposition 6(i) (specifically, the fact that $(\cdot)^{\perp\perp}$ is inflationary) to the definition of $[\![(A \mathbin{\bindnasrepma} B)^{\perp}]\!]\eta$, we see that

$$\otimes_{\gamma}^{\alpha,\beta}(P\bar{P}, Q\bar{Q}) \in ([\![A^{\perp} \mathbin{\bindnasrepma} B^{\perp}]\!]\eta)^{\perp},$$

and hence that $\mathtt{Cut}(\otimes_{\gamma}^{\alpha,\beta}(P\bar{P}, Q\bar{Q}), R)$. But

$$\mathtt{Cut}(\otimes_{\gamma}^{\alpha,\beta}(P\bar{P}, Q\bar{Q}), R) \Downarrow \equiv \otimes_{\gamma}^{\alpha,\beta}(P, Q)\bar{P}\bar{Q}R,$$

so $\otimes_{\gamma}^{\alpha,\beta}(P, Q)\bar{P}\bar{Q}R \Downarrow$.

(8) With:

$$\frac{P \vdash \Gamma, \alpha : A \quad Q \vdash \Gamma, \beta : B}{\&_{\gamma}^{\alpha,\beta}(P, Q) \vdash \Gamma, \gamma : A \mathbin{\&} B}$$

Fix $\eta \in \mathsf{TEnv}$, $\bar{P} \in [\![\Gamma^{\perp}]\!]\eta$. We must show that for all $R \in [\![(A \mathbin{\&} B)^{\perp}]\!]\eta$, $\&_{\gamma}^{\alpha,\beta}(P, Q)\bar{P}R \Downarrow$. Reasoning as in the case for Par, it suffices to consider Q of the form *either* $L_{\gamma}^{\alpha}(S)$, $S \in [\![A^{\perp}]\!]\eta$, *or* $R_{\gamma}^{\alpha}(T)$, $T \in [\![B^{\perp}]\!]\eta$. In the first case, $\&_{\gamma}^{\alpha,\beta}(P, Q)\bar{P}L_{\gamma}^{\alpha}(S) \xrightarrow{*} P\bar{P}S$, and by induction hypothesis $P\bar{P}S \Downarrow$, so by **(P2)**, $\&_{\gamma}^{\alpha,\beta}(P, Q)\bar{P}R \Downarrow$. The second case is similar.

(9) Plus Left:

$$\frac{P \vdash \Gamma, \alpha : A}{L_{\gamma}^{\alpha}(P) \vdash \Gamma, \gamma : A \oplus B}$$

Fix $\eta \in \mathsf{TEnv}$, $\bar{Q} \in [\![\Gamma^{\perp}]\!]\eta$. We must show that for all $Q \in [\![(A \oplus B)^{\perp}]\!]\eta$, $L_{\gamma}^{\alpha}(P)\bar{Q}Q \Downarrow$. By induction hypothesis, for all $R \in [\![A^{\perp}]\!]\eta$, $P\bar{Q}R \Downarrow$, so $P\bar{Q} \in ([\![A^{\perp}]\!]\eta)^{\perp}$, and

$$L_{\gamma}^{\alpha}(P\bar{Q}) \in ([\![A^{\perp} \mathbin{\&} B^{\perp}]\!]\eta)^{\perp} = [\![A \oplus B]\!]\eta,$$

so $\mathtt{Cut}(L_\gamma^\alpha(P\bar{Q}),Q)\Downarrow$. But

$$\mathtt{Cut}(L_\gamma^\alpha(P\bar{Q}),Q)\equiv L_\gamma^\alpha(P)\bar{Q}Q,$$

so $L_\gamma^\alpha(P)\bar{Q}Q\Downarrow$.

(10) Plus Right: similar to Plus Left.

(11) Dereliction:

$$\frac{P\vdash\Gamma,\alpha:A}{D_\gamma^\alpha(P)\vdash\Gamma,\gamma:?A}$$

Fix $\eta\in\mathsf{TEnv}$, $\bar{Q}\in[\![\Gamma^\perp]\!]\eta$. We must show that for all $Q\in[\![(?A)^\perp]\!]\eta$, $D_\gamma^\alpha(P)\bar{Q}Q\Downarrow$. Reasoning as in the case for Par, it suffices to consider Q of the form $!_\gamma^\alpha(R)$, $R\in[\![A^\perp]\!]\eta$. But $D_\gamma^\alpha(P)\bar{Q}!_\gamma^\alpha(R)\xrightarrow{*}P\bar{Q}R$, and by the induction hypothesis $P\bar{Q}R\Downarrow$. Hence by **(P2)**, $D_\gamma^\alpha(P)\bar{Q}Q\Downarrow$.

(12) Contraction:

$$\frac{P\vdash\Gamma,\alpha:?A,\beta:?B}{C_\gamma^{\alpha,\beta}(P)\vdash\Gamma,\gamma:?A}$$

Fix $\eta\in\mathsf{TEnv}$, $\bar{Q}\in[\![\Gamma^\perp]\!]\eta$. We must show that for all $Q\in[\![(?A)^\perp]\!]\eta$, $C_\gamma^{\alpha,\beta}(P)\bar{Q}Q\Downarrow$. Reasoning as in the case for Par, it suffices to consider Q of the form $!_\gamma^\alpha(R)$, $R\in[\![A^\perp]\!]\eta$. But $C_\gamma^{\alpha,\beta}(P)\bar{Q}!_\gamma^\alpha(R)\xrightarrow{*}C_{\vec{\alpha}}^{\vec{\alpha}',\vec{\alpha}''}(P\bar{Q}\ !_{\gamma'}^\alpha(R[\vec{\alpha'}/\vec{\alpha}])\ !_{\gamma''}^\alpha(R[\vec{\alpha}''/\vec{\alpha}]))$, where $\mathtt{FN}(R)\setminus\{\alpha\}=\vec{\alpha}$. Now

$$C_{\vec{\alpha}}^{\vec{\alpha}',\vec{\alpha}''}(P\bar{Q}\ !_{\gamma'}^\alpha R[\vec{\alpha'}/\vec{\alpha}]\ !_{\gamma''}^\alpha R[\vec{\alpha}''/\vec{\alpha}])\Downarrow\Longleftrightarrow P\bar{Q}\ !_{\gamma'}^\alpha R[\vec{\alpha'}/\vec{\alpha}]\ !_{\gamma''}^\alpha R[\vec{\alpha}''/\vec{\alpha}]\Downarrow$$

But by induction hypothesis $P\bar{Q}\ !_{\gamma'}^\alpha R[\vec{\alpha'}/\vec{\alpha}]\ !_{\gamma''}^\alpha R[\vec{\alpha}''/\vec{\alpha}]\Downarrow$, hence by **(P2)**, $C_\gamma^{\alpha,\beta}(P)\bar{Q}Q\Downarrow$.

(13) Weakening:

$$\frac{P\vdash\Gamma}{W_\gamma(P)\vdash\Gamma,\gamma:?A}$$

Fix $\eta\in\mathsf{TEnv}$, $\bar{Q}\in[\![\Gamma^\perp]\!]\eta$. We must show that for all $Q\in[\![(?A)^\perp]\!]\eta$, $W_\gamma(P)\bar{Q}Q\Downarrow$. Once again, it suffices to consider Q of the form $!_\gamma^\alpha(R)$, $R\in[\![A^\perp]\!]\eta$. But $W_\gamma(P)\bar{Q}\ !_\gamma^\alpha(R)\xrightarrow{*}W_{\vec{\alpha}}(P\bar{Q})$, where $\mathtt{FN}(R)\setminus\{\alpha\}=\vec{\alpha}$. Also, $W_{\vec{\alpha}}(P\bar{Q})\Downarrow\Longleftrightarrow P\bar{Q}\Downarrow$. By induction hypothesis $P\bar{Q}\Downarrow$, hence by **(P2)**, $W_\gamma(P)\bar{Q}Q\Downarrow$.

(14) Of Course:

$$\frac{P\vdash ?\Gamma,\alpha:A}{!_\gamma^\alpha(P)\vdash ?\Gamma,\gamma:!A}$$

Fix $\eta\in\mathsf{TEnv}$, $\bar{Q}\in[\![?\Gamma^\perp]\!]\eta$. We must show that for all $Q\in[\![(!A)^\perp]\!]\eta$, $!_\gamma^\alpha(P)\bar{Q}Q\Downarrow$. By induction hypothesis, for all $R\in[\![A^\perp]\!]\eta$, $P\bar{Q}R\Downarrow$, hence $P\bar{Q}\in([\![A^\perp]\!]\eta)^\perp$, so $!_\gamma^\alpha((P\bar{Q}))\in([\![?(A^\perp)]\!]\eta)^\perp$, and $\mathtt{Cut}(!_\gamma^\alpha(P\bar{Q}),Q)\Downarrow$. We must show that this implies that $!_\gamma^\alpha(P)\bar{Q}Q\Downarrow$.

Firstly, we claim that it is sufficient to prove that $!^{\alpha}_{\gamma}(P)\bar{Q}Q \Downarrow$ for $\bar{Q}$ of the form $!^{\alpha_1}_{\gamma_1}(R_1), \ldots, !^{\alpha_k}_{\gamma_k}(R_k)$. To see this, note that

$$\begin{array}{ll} !^{\alpha}_{\gamma}(P)\bar{Q}Q \Downarrow & \Longleftrightarrow \\ ⅋^{\alpha_1,\ldots,\alpha_k}_{\gamma}(!^{\alpha}_{\gamma}(P)Q) \otimes^{\alpha_1,\ldots,\alpha_k}_{\gamma}(\bar{Q}) \Downarrow & \Longleftrightarrow \\ (⅋^{\alpha_1,\ldots,\alpha_k}_{\gamma}(!^{\alpha}_{\gamma}(P)Q) \perp \otimes^{\alpha_1,\ldots,\alpha_k}_{\gamma}(\bar{Q}), & \end{array}$$

and that

$$\begin{array}{rcl} ⅋^{\alpha_1,\ldots,\alpha_k}_{\gamma}(!^{\alpha}_{\gamma}(P)Q) & \perp & [\![\bigotimes(\,?\Gamma^{\perp})]\!]\eta \\ & = & \{\otimes^{\alpha_1,\ldots,\alpha_k}_{\gamma}(!^{\alpha}_{\gamma}(\bar{R})) \mid \bar{R} \in [\![\Gamma^{\perp}]\!]\eta\}^{\perp\perp} \end{array}$$

if and only if

$$⅋^{\alpha_1,\ldots,\alpha_k}_{\gamma}(!^{\alpha}_{\gamma}(P)Q) \perp \{\otimes^{\alpha_1,\ldots,\alpha_k}_{\gamma}(!^{\alpha}_{\gamma}(\bar{R})) \mid \bar{R} \in [\![\Gamma^{\perp}]\!]\eta\}$$

by Proposition 6(iii). However, by **(R9)**, $!^{\alpha}_{\gamma}(P)\bar{Q}Q \xrightarrow{*} \mathtt{Cut}(!^{\alpha}_{\gamma}(P\bar{Q}), Q)$,

$$\mathtt{Cut}(!^{\alpha}_{\gamma}((P\bar{Q})), Q) \Downarrow \Rightarrow !^{\alpha}_{\gamma}(P)\bar{Q}Q \Downarrow$$

(15) All:

$$\frac{P \vdash \Gamma, A}{P \vdash \Gamma, \forall\alpha.A} \quad (*)$$

Fix $\eta \in \mathsf{TEnv}$, $\bar{Q} \in [\![\Gamma^{\perp}]\!]\eta$. We must show that for all $Q \in [\![(\forall\alpha.A)^{\perp}]\!]\eta$, $P\bar{Q}Q \Downarrow$. Reasoning as in the case for Par, it suffices to consider $Q \in F(U)^{\perp}$ for some $U \in \mathcal{U}$, where $F = \lambda U.[\![A]\!]\eta[\alpha \mapsto U]$. By the eigenvariable condition $[\![\Gamma^{\perp}]\!]\eta = [\![\Gamma^{\perp}]\!]\eta[\alpha \mapsto U]$, so by the induction hypothesis (with respect to $\eta[\alpha \mapsto U]$), $P\bar{Q}Q \Downarrow$.

(16) Exists:

$$\frac{P \vdash \Gamma, A[B/\alpha]}{P \vdash \Gamma, \exists\alpha.A}$$

Fix $\eta \in \mathsf{TEnv}$, $\bar{Q} \in [\![\Gamma^{\perp}]\!]\eta$. We must show that for all $Q \in [\![(\exists\alpha.A)^{\perp}]\!]\eta$, $P\bar{Q}Q \Downarrow$. By induction hypothesis, for all $R \in [\![A[B/\alpha]^{\perp}]\!]\eta$, $P\bar{Q}R \Downarrow$. Now

$$[\![A[B/\alpha]^{\perp}]\!]\eta = [\![A^{\perp}]\!]\eta[\alpha \mapsto [\![B]\!]\eta] = F(U)^{\perp},$$

where $F = \lambda U.[\![A^{\perp}]\!]\eta[\alpha \mapsto U]$, $U = [\![B]\!]\eta$. (It is just at this point in the proof that second-order comprehension is used.) Hence $P\bar{Q} \in \forall(F)^{\perp} = ([\![(\exists\alpha.A)^{\perp}]\!]\eta)^{\perp}$, so $\mathtt{Cut}(P\bar{Q}, Q) \Downarrow$. But $\mathtt{Cut}(P\bar{Q}, Q) \equiv P\bar{Q}Q$, so $P\bar{Q}Q \Downarrow$. ∎

As an immediate consequence of the Realizability Theorem, we get

Theorem 2 (Convergence) *Let $\mathcal{A}$ be an LRA, and let $P \vdash \Gamma$ be the realizer for a sequent proof in $\mathtt{CLL}_2$. Then,*

1. *If* $P \rightarrow Q$*, then* $[\![P]\!] \approx [\![Q]\!]$*. In particular, if* Q *is any cut-free proof obtainable from* P *under LRA-reductions, then* $[\![P]\!] \approx [\![Q]\!]$*.*

2. $P\Downarrow$*.*

Proof: The proof of the first part follows from **(P0), (P2)** by a straightforward induction on the length of reduction.

For the second part, by the Realizability Theorem,

$$P \vdash \Gamma \Rightarrow P \models \Gamma.$$

Now choose $\eta \in \mathsf{TEnv}$, $\bar{I}_{\bar{\alpha},\bar{\beta}} \in [\![\Gamma^{\perp}]\!]\eta$, and conclude that $P\bar{I}_{\bar{\alpha},\bar{\beta}} \Downarrow$. Hence, by **(P4)**, $P \Downarrow$. ∎

Note that the proof of the Realizability Theorem uses the hypothesis **(P2)** only for "outermost" reductions. Thus, it suffices to assume **(P2)** only for "outermost" reductions.

6.3 Geometry of Interaction as an LRA

We can now rephrase our goal of proving the soundness of the $\mathcal{GI}$ interpretation more precisely; we want to show that the $\mathcal{GI}$ interpretation— or a reformulation of it—is an LRA.

The trivial reformulation involves replacing functions

$$f : D^{2m+n} \rightarrow D^{2m+n}$$

by

$$f : D^{2\vec{\beta}+\vec{\alpha}} \rightarrow D^{2\vec{\beta}+\vec{\alpha}}$$

where $\vec{\alpha}, \vec{\beta}$ are sorts with $\mathtt{card}(\vec{\alpha}) = n, \mathtt{card}(\vec{\beta}) = m$ and $2\vec{\beta} = \vec{\beta} + \vec{\beta}$ (disjoint union). In other words, we replace vectors $(x_1 \ldots x_k)$ by records $[\alpha_1 \mapsto x_1, \ldots, \alpha_k \mapsto x_k]$.

It is straightforward to rewrite the definition of the $\mathcal{GI}$ interpretation, as given in section 4, in this style. An example should suffice: we describe the case of *Times* below: Let

$$\begin{array}{rcl} f & : & D^{2\vec{\beta}'+\vec{\alpha}',\alpha'} \rightarrow D^{2\vec{\beta}'+\vec{\alpha}',\alpha'} \\ g & : & D^{2\vec{\beta}''+\vec{\alpha}'',\alpha''} \rightarrow D^{2\vec{\beta}''+\vec{\alpha}'',\alpha''} \\ \vec{\beta} & = & \vec{\beta}' + \vec{\beta}'' \\ \vec{\alpha} & = & \vec{\alpha} + \vec{\alpha} \end{array}$$

Then,

$$\begin{array}{rcl} \otimes_{\alpha}^{\alpha',\alpha''}(f,g) & = & D^{2\vec{\beta}+\vec{\alpha},\alpha} \\ & \xrightarrow{1\times m^{\star}} & D^{2\vec{\beta}+\vec{\alpha},\alpha',\alpha''} \cong D^{2\vec{\beta}'+\vec{\alpha}',\alpha'} \times D^{2\vec{\beta}''+\vec{\alpha}'',\alpha''} \\ & \xrightarrow{f\times g} & D^{2\vec{\beta}'+\vec{\alpha}',\alpha'} \times D^{2\vec{\beta}''+\vec{\alpha}'',\alpha''} \cong D^{2\vec{\beta}+\vec{\alpha},\alpha',\alpha''} \\ & \xrightarrow{1\times m} & D^{2\vec{\beta}+\vec{\alpha},\alpha} \end{array}$$

In terms of elements, $\otimes_{\alpha}^{\alpha',\alpha''}(f,g)([2\vec{\beta}' \mapsto \vec{u}', 2\vec{\beta}'' \mapsto \vec{u}'', \vec{\alpha}' \mapsto \vec{x}', \vec{\alpha}'' \mapsto \vec{x}'', \alpha \mapsto x]) =$ $([2\vec{\beta}' \mapsto \vec{v}', 2\vec{\beta}'' \mapsto \vec{v}'', \vec{\alpha}' \mapsto \vec{y}', \vec{\alpha}'' \mapsto \vec{y}'', \alpha \mapsto m(y', y'')])$, where

$$\begin{array}{rcl} f([2\vec{\beta}' \mapsto \vec{u}', \vec{\alpha}' \mapsto \vec{x}', \alpha' \mapsto x']) & = & [2\vec{\beta}' \mapsto \vec{v}', \vec{\alpha}' \mapsto \vec{y}', \alpha' \mapsto y'] \\ g([2\vec{\beta}'' \mapsto \vec{u}'', \vec{\alpha}'' \mapsto \vec{x}'', \alpha'' \mapsto x'']) & = & [2\vec{\beta}'' \mapsto \vec{v}'', \vec{\alpha}'' \mapsto \vec{y}'', \alpha'' \mapsto y''] \\ m^{\star}(x) & = & (x', x'') \end{array}$$

Now, we fix a $\mathcal{GI}$ interpretation based on a type free model D, and define the corresponding LRA $\mathcal{A}$ as follows. Firstly, for each sort $\vec{\alpha}$, $A_{\vec{\alpha}}$ is the set of all endomorphisms

$$f : D^{2\vec{\beta}+\vec{\alpha}} \to D^{2\vec{\beta}+\vec{\alpha}}$$

where $\vec{\beta}$ is a sort. Renaming is interpreted in the obvious way, by composition: let $[\vec{\gamma}/\vec{\alpha}]$ be bijective renaming; then, $f[\vec{\gamma}/\vec{\alpha}] \in A_{\vec{\gamma}}]$, where $f : D^{2\vec{\beta}+\vec{\gamma}} \to D^{2\vec{\beta}+\vec{\gamma}}$ is defined as

$$f[\vec{\gamma}/\vec{\alpha}]([2\vec{\beta} \mapsto \vec{u}, \vec{\gamma} \mapsto \vec{x}]) = [2\vec{\beta} \mapsto \vec{v}, \vec{\gamma} \mapsto \vec{y}]$$

where $f([2\vec{\beta} \mapsto \vec{u}, \vec{\alpha} \mapsto \vec{x}]) = [2\vec{\beta} \mapsto \vec{v}, \vec{\alpha} \mapsto \vec{y}]$.

There is also an evident notion of α-conversion between functions of the same sort.

$$f \equiv_{\alpha} g \text{ if } \begin{array}{l} f : D^{2\vec{\beta}+\vec{\alpha}} \to D^{2\vec{\beta}+\vec{\alpha}} \\ g : D^{2\vec{\gamma}+\vec{\gamma}} \to D^{2\vec{\beta}+\vec{\gamma}} \end{array}$$

and there is a bijective renaming $[\vec{\gamma}/\vec{\beta}]$ such that

$$f([2\vec{\beta} \mapsto \vec{u}, \vec{\alpha} \mapsto \vec{x}]) = g([2\vec{\beta} \mapsto \vec{u}, \vec{\gamma} \mapsto \vec{x}])$$

for all $\vec{u}, \vec{x}$. We shall take functions modulo α-conversion; that is we identify α-convertible functions.

The definition of the LRA equations is precisely the content of the $\mathcal{GI}$ interpretation, once reformulated as explained above. Given $f : D^{2\vec{\beta}+\vec{\alpha}} \to D^{2\vec{\beta}+\vec{\alpha}}$, define the *message exchange* function $\sigma_f : D^{2\vec{\beta}} \to D^{2\vec{\beta}}$ by:

$$\sigma_f([\beta_1^0 \mapsto x_1, \beta_1^1 \mapsto y_1, \ldots, \beta_k^0 \mapsto x_k, \beta_k^1 \mapsto y_k]) = [\beta_1^0 \mapsto y_1, \beta_1^1 \mapsto x_1, \ldots, \beta_k^0 \mapsto y_k, \beta_k^1 \mapsto x_k]$$

We can now define: $FB(f, \sigma_f) = \pi(\bigsqcup_k f^{(k)})$, just as before, where $\pi(\vec{u}, \vec{x}) = \vec{x}$, with $f^{(k)} : D^{\vec{\alpha}} \to D^{2\vec{\beta}+\vec{\alpha}}$ defined inductively by:

$$\begin{array}{rcl} f^{(0)} & = & \perp \\ f^{(k+1)} & = & (\sigma_f \times 1) \circ f \circ \langle \pi' \circ f^{(k)}, 1 \rangle \end{array}$$

where $\pi'(\vec{u}, \vec{x}) = \vec{u}$.

We now complete the definition of $\mathcal{A}$ by:

$$\begin{array}{rcl} f \approx g & \stackrel{\text{def}}{=} & FB(f, \sigma_f) = FB(g, \sigma_g) \\ f\Downarrow & \stackrel{\text{def}}{=} & \exists k, f\Downarrow_k \end{array}$$

where $f\Downarrow_k \stackrel{\text{def}}{=} f^{(k)} = f^{(k+1)}$.

The remainder of this section is devoted to proving that $\mathcal{A}$ is an LRA. Thus, we have to prove that the conditions of the following table (reproduced from section 6.2) are satisfied:

	Hypothesis	Conditions	
P0	$[\![P]\!] \approx [\![Q]\!]$	$[\![C[P]]\!] \approx [\![C[Q]]\!]$	
P1	$P \equiv Q$	$[\![P]\!] = [\![Q]\!]$	
P2	$P \rightarrow Q$	$[\![P]\!] \approx [\![Q]\!]$	$Q\Downarrow \Rightarrow P\Downarrow$
P3	ω a constructor (any operator except Cut)	$\dfrac{P_1\Downarrow, \ldots, P_k\Downarrow}{\omega(P_1, \ldots, P_k)\Downarrow}$	
P4	$P \cdot_\alpha I_{\alpha,\beta}\Downarrow$	$P\Downarrow$	

The first step of the proof is checking property P1. The proof is immediate and is omitted.

Proposition 8 $P \equiv Q \Rightarrow [\![P]\!] = [\![Q]\!]$.

Proving P0, P3

The second step of the proof is to set up the machinery to prove P0 and P3.

Given $f : D^{2\vec{\beta}+\vec{\alpha}} \rightarrow D^{2\vec{\beta}+\vec{\alpha}}$, $g : D^{\vec{\alpha}} \rightarrow D^{2\vec{\beta}+\vec{\alpha}}$, define

$$\theta(f,g) = (\sigma_f \times 1) \circ f \circ \langle \pi' \circ g, i \rangle$$

Thus, $f^{(k+1)} = \theta(f, f^{(k)})$.

Lemma 3 *For each constructor* ω, $\theta(\omega(f_1, \ldots, f_k), \omega(g_1, \ldots, g_k)) = \omega(\theta(f_1, g_1), \ldots, \theta(f_k, g_k))$.

Proof: Note that the definitions of each of the constructors in $\{U, \perp, \top, \otimes, \invamp, L, R, D, W, C\}$ has the form:

$$\omega(f_1, \ldots f_k) = (1 \times r) \circ (f_1 \times \ldots \times f_k) \circ (1 \times r^\star)$$

for some retractions r and for some $1 \leq k \leq 2$. Let $f = f_1 \times \ldots \times f_k$ and $g = g_1 \times \ldots \times g_k$

$$\begin{aligned}
\theta(\omega(f_1, \ldots, f_k), \omega(g_1, \ldots, g_k)) &= (\sigma \times 1) \circ (1 \times r) \circ f \circ (1 \times r^\star) \circ \langle \pi' \circ (1 \times r) \circ g \circ (1 \times r^\star), 1 \rangle \\
&= (1 \times r) \circ (\sigma \times 1) \circ f \circ \langle \pi' \circ (1 \times r) \circ g, 1 \rangle \circ (1 \times r^\star) \\
&= (1 \times r) \circ (\sigma \times 1) \circ f \circ \langle \pi' \circ g, 1 \rangle \circ (1 \times r^\star) \\
&= (1 \times r) \circ [\theta(f_1, g_1) \times \ldots \times \theta(f_k, g_k)] \circ (1 \times r^\star) \\
&= \omega(\theta(f_1, g_1), \ldots, \theta(f_k, g_k)]
\end{aligned}$$

Next, we prove the result for the constructor &.

$$\begin{aligned}
\theta(\&(f_1,f_2),\&(g_1,g_2)) &= \alpha\circ[f_1,f_2]_\perp\circ dist_\perp\circ t\circ(1\times a^\star)\circ\langle\pi'\circ\alpha\circ[g_1,g_2]_\perp\circ dist_\perp\circ t\circ(1\times a^\star),1\rangle\\
&= \alpha\circ([f_1,f_2]_\perp\circ\langle\pi'\circ[g_1,g_2]_\perp,1\rangle\circ dist_\perp\circ t\circ(1\times a^\star)\\
&= \alpha\circ([f_1\circ\langle\pi'\circ g_1,1\rangle,f_2\circ\langle\pi'\circ g_2,1\rangle]_\perp)\circ dist_\perp\circ t\circ(1\times a^\star)\\
&= \&(\theta(f_1,g_1),\theta(f_2,g_2))
\end{aligned}$$

The proof for the constructor ! is similar and is omitted. ∎

Lemma 4 *For each constructor ω and $k\geq 1$, $[\omega(\vec{f})]^{(n)}=\omega(\vec{f}^{(n)})$.*

Proof: By induction on n. For $n=1$:

$$\begin{aligned}
[\omega(\vec{f})]^{(1)} &= \theta(\omega(\vec{f}),\vec{\perp})\\
&= \theta(\omega(\vec{f}),\omega(\vec{\perp}))\quad[\pi'\circ\omega(\vec{\perp})=\vec{\perp}]\\
&= \omega(\theta(\vec{f},\vec{\perp}))\ [lemma\ 3]\\
&= \omega(\vec{f}^{(1)})
\end{aligned}$$

For the induction step,

$$\begin{aligned}
[\omega(\vec{f})]^{(k+1)} &= \theta(\omega(\vec{f}),[\omega(\vec{f})]^{(k)})\\
&= \theta(\omega(\vec{f}),\omega(\vec{f}^{(k)})\quad[Induction]\\
&= \omega(\theta(\vec{f},\vec{f}^{(k)}))\ [lemma\ 3]\\
&= \omega(\vec{f}^{(k+1)})
\end{aligned}$$

∎

Corollary 1 *For all constructors ω,*

1. $FB(\omega(f_1,\ldots f_k),\sigma_{\omega(f_1,\ldots,f_k)})=\omega(FB(f_1,\sigma_{f_1}),\ldots FB(f_k,\sigma_{f_k}))$.
2. $f_1\Downarrow_{i_1},\ldots,f_k\Downarrow_{i_k}\Rightarrow\omega(f_1,\ldots,f_k)\Downarrow_{max(i_1,\ldots,i_k)}$.

Cut preserves FB

The aim of this subsection is to prove that

$$FB(f\cdot g,\sigma_{f\,\cdot\,g})=FB(FB(f,\sigma_f)\cdot FB(g,\sigma_g),\sigma_{FB(f)\,\cdot\,FB(g)})$$

Lemma 5 *Let* $f : D^{2\vec{\beta}'+\vec{\alpha}} \to D^{2\vec{\beta}+\vec{\alpha}}$. *Let* $2\vec{\beta}' = \vec{\beta}$. *Let* $H = FB(f, \sigma_F)$. *Then,*

$$H[\vec{\alpha} \mapsto \vec{a}] = [\vec{\alpha} \mapsto \vec{b}]$$

if and only if $(\exists \vec{b})$ *such that:*

$$\begin{array}{rcl} f[\vec{\beta} \mapsto \vec{b}, \vec{\alpha} \mapsto \vec{a}] & = & [\vec{\beta} \mapsto \sigma_f(\vec{b}), \vec{\alpha} \mapsto \vec{b}] \\ \vec{b} \sqsubseteq \vec{x} & \Leftarrow & f[\vec{\beta} \mapsto \vec{x}, \vec{\alpha} \mapsto \vec{a}] = [\vec{\beta} \mapsto \sigma(\vec{x}), \vec{\alpha} \mapsto \vec{b}'] \end{array}$$

Proof: Define $\vec{b}$ by:

$$\bigsqcup_k (f^{(k)})[\vec{\alpha} \mapsto \vec{a}] = [\vec{\beta} \mapsto \sigma_f(\vec{b}), \vec{\alpha} \mapsto \vec{a}]$$

Then, it is immediate from definitions that

$$f[\vec{\beta} \mapsto \vec{b}, \vec{\alpha} \mapsto \vec{a}] = [\vec{\beta} \mapsto \sigma_f(\vec{b}), \vec{\alpha} \mapsto \vec{a}]$$

Let $\vec{x}, \vec{b}'$ satisfy

$$f[\vec{\beta} \mapsto \vec{x}, \vec{\alpha} \mapsto \vec{a}] = [\vec{\beta} \mapsto \sigma(\vec{x}), \vec{\alpha} \mapsto \vec{b}']$$

We prove that $\vec{b} \sqsubseteq \vec{x}$. This is done by induction on the iterates $f^{(k)}$. Denote the iterates approximating $\vec{b}$ by $\vec{b}_k$. Base case is immediate. For the inductive case:

$$\begin{array}{rcl} f^{(k+1)}[\vec{\alpha} \mapsto \vec{a}] & = & (\sigma_f \times 1) \circ f[\vec{\beta} \mapsto \vec{b}_k, \vec{\alpha} \mapsto \vec{a}] \\ & \sqsubseteq & (\sigma_f \times 1) f[\vec{\alpha} \mapsto \vec{x}, \vec{\alpha} \mapsto \vec{a}] \end{array}$$

∎

Lemma 6 *Let* $h = FB(f, \sigma_f), k = FB(g, \sigma_g)$. *Then,*

$$FB(f \cdot g, \sigma_{f \cdot g}) = FB(h \cdot k, \sigma_{h \cdot k})$$

Proof: Using lemma 5,

$$FB(h \cdot k, \sigma_{h \cdot k})[\vec{\alpha}_f \mapsto \vec{a}_f, \vec{\alpha}_g \mapsto \vec{a}_g] = [\vec{\alpha}_f \mapsto \vec{b}'_f, \vec{\alpha}_g \mapsto \vec{b}'_g]$$

if and only if $(\exists b_f, b_g)$ such that:

$$\begin{array}{rcl} h[\vec{\alpha}_f \mapsto \vec{a}_f, \beta_f \mapsto b_g] & = & [\vec{\alpha}_f \mapsto \vec{b}_f, \beta_f \mapsto b_f] \\ k[\vec{\alpha}_g \mapsto \vec{a}_g, \beta_g \mapsto b_f] & = & [\vec{\alpha}_g \mapsto \vec{b}_g, \beta_g \mapsto b_g] \\ (b_f, b_g) \sqsubseteq (x_f, x_g) & \Leftarrow & \left\{ \begin{array}{rcl} h[\vec{\alpha}_f \mapsto \vec{a}_f, \beta_f \mapsto x_g] & = & [\vec{\alpha}_f \mapsto \vec{b}_f, \beta_f \mapsto x_f] \\ k[\vec{\alpha}_g \mapsto \vec{a}_g, \beta_g \mapsto x_f] & = & [\vec{\alpha}_g \mapsto \vec{b}_g, \beta_g \mapsto x_g] \end{array} \right. \end{array}$$

Using lemma 5, we deduce that

$$h[\vec{\alpha}_f \mapsto \vec{a}_f, \beta_f \mapsto b_g] = [\vec{\alpha}_f \mapsto \vec{b}_f, \beta_f \mapsto b_f]$$

if and only if $(\exists \vec{b}_f)$ such that:

$$\begin{array}{rcl} f[\vec{\beta}_f \mapsto \vec{b}_f, \vec{\alpha}_f \mapsto \vec{a}_f, \beta_f \mapsto b_g] & = & [\vec{\beta}_f \mapsto \sigma(\vec{b}_f), \vec{\alpha}_f \mapsto \vec{b}_f, \beta_f \mapsto b_f] \\ \vec{b}_f \sqsubseteq \vec{x}_f & \Leftarrow & f[\vec{\beta} \mapsto \vec{x}, \vec{\alpha} \mapsto \vec{a}, \beta_f \mapsto b_g] = [\vec{\beta} \mapsto \sigma(\vec{x}), \vec{\alpha} \mapsto \vec{b}', \beta_f \mapsto b'_f] \end{array}$$

Similarly, using lemma 5,

$$k[\vec{\alpha}_g \mapsto \vec{a}_g, \beta_g \mapsto b_f] = [\vec{\alpha}_g \mapsto \vec{b}_g, \beta_g \mapsto b_g]$$

if and only if $(\exists \vec{b}_g)$ such that:

$$\begin{array}{rcl} g[\vec{\beta}_g \mapsto \vec{b}_g, \vec{\alpha}_g \mapsto \vec{a}_g, \beta_g \mapsto b_g] & = & [\vec{\beta}_g \mapsto \sigma(\vec{b}_g), \vec{\alpha}_g \mapsto \vec{b}_g, \beta_g \mapsto b_g] \\ \vec{b}_g \sqsubseteq \vec{x}_g & \Leftarrow & g[\vec{\beta} \mapsto \vec{x}, \vec{\alpha} \mapsto \vec{a}, \beta_g \mapsto b_g] = [\vec{\beta} \mapsto \sigma(\vec{x}), \vec{\alpha} \mapsto \vec{b}', \beta_g \mapsto b'_g] \end{array}$$

Putting the above equivalences together,

$$FB(h \cdot k, \sigma_{h \cdot k})[\vec{\alpha}_f \mapsto \vec{a}_f, \vec{\alpha}_g \mapsto \vec{a}_g] = [\vec{\alpha}_f \mapsto \vec{b}'_f, \vec{\alpha}_g \mapsto \vec{b}'_g]$$

if and only if $(\exists \vec{b}_f, b_f, b_g, \vec{b}_g)$ such that:

$$\begin{array}{rcl} f[\vec{\beta}_f \mapsto \vec{b}_f, \vec{\alpha}_f \mapsto \vec{a}_f, \beta_f \mapsto b_g] & = & [\vec{\beta}_f \mapsto \sigma(\vec{b}_f), \vec{\alpha}_f \mapsto \vec{b}_f, \beta_f \mapsto b_f] \\ g[\vec{\beta}_g \mapsto \vec{b}_g, \vec{\alpha}_g \mapsto \vec{a}_g, \beta_g \mapsto b_g] & = & [\vec{\beta}_g \mapsto \sigma(\vec{b}_g), \vec{\alpha}_g \mapsto \vec{b}_g, \beta_g \mapsto b_g] \\ (b_f, \vec{b}_f, b_g, \vec{b}_g) \sqsubseteq (x_g, \vec{x}_f, b_g, \vec{b}_g) & \Leftarrow & \left\{ \begin{array}{rcl} f[\vec{\beta} \mapsto \vec{x}_f, \vec{\alpha} \mapsto \vec{a}, \beta_f \mapsto b_g] & = & [\vec{\beta}_f \mapsto \sigma(\vec{x}_f), \vec{\alpha}_f \mapsto \vec{b}', \beta_f \mapsto b'_f] \\ g[\vec{\beta}_g \mapsto \vec{x}_g, \vec{\alpha}_g \mapsto \vec{a}, \beta_g \mapsto b_g] & = & [\vec{\beta}_g \mapsto \sigma(\vec{x}), \vec{\alpha}_g \mapsto \vec{b}', \beta_g \mapsto b'_g] \end{array} \right. \end{array}$$

Using lemma 5 again, we note that

$$FB(f \cdot g, \sigma_{f \cdot g})[\vec{\alpha}_f \mapsto \vec{a}_f, \vec{\alpha}_g \mapsto \vec{a}_g] = [\vec{\alpha}_f \mapsto \vec{b}'_f, \vec{\alpha}_g \mapsto \vec{b}'_g]$$

if and only if $(\exists \vec{b}_f, b_f, b_g, \vec{b}_g)$ satisfying the above conditions. Hence, the result. ∎

Proposition 9 *[Proof of P0,P3]*

1. *$\approx$ is a congruence.*
2. *$f_1\Downarrow, \ldots, f_k\Downarrow \Rightarrow \omega(f_1, \ldots f_k)\Downarrow$, where ω is a constructor.*

Proof: The proof of the second part is immediate from Corollary 1(2). The proof of the first part is by structural induction. The induction steps are proved below: Let $f_1 \approx g_1, \ldots, f_k \ldots g_k$. Thus, $FB(f_1, \sigma_{f_1}) = FB(g_1, \sigma_{g_1}), \ldots, FB(f_k, \sigma_{f_k}) = FB(g_k, \sigma_{g_k})$.

Constructors: Let ω be a constructor. We use Corollary 1(1).

$$\begin{array}{rcl} FB(\omega(f_1, \ldots, f_k), \sigma_{\omega(f_1,\ldots,f_k)}) & = & \omega(FB(f_1, \sigma_{f_1}), \ldots, FB(f_k, \sigma_{f_k})) \\ & = & \omega(FB(g_1, \sigma_{g_1}), \ldots, FB(g_k, \sigma_{g_k})) \\ & = & FB(\omega(g_1, \ldots, g_k), \sigma_{\omega(g_1,\ldots,g_k)}) \end{array}$$

Cut: In this case, we use Lemma 6.

$$\begin{aligned} FB(f_1 \cdot f_2, \sigma_{f_1 \cdot f_2}) &= FB(f_1, \sigma_{f_1}) \cdot FB(f_2, \sigma_{f_2}) \\ &= FB(g_1, \sigma_{g_1}) \cdot FB(g_2, \sigma_{g_2}) \\ &= FB(g_1 \cdot g_2, \sigma_{(g_1 \cdot g_2)}) \end{aligned}$$

∎

Proving (P2): Reduction Rules: (R2)–(R8)

First, we set up the machinery that factors out the details of the *single* proof that works for all cases.

Lemma 7 *Suppose* $f : D^{2\vec{\beta}+\vec{\alpha}} \to D^{2\vec{\beta}+\vec{\alpha}}$ *and* $g : D^{2\vec{\gamma}+\vec{\alpha}} \to D^{2\vec{\gamma}+\vec{\alpha}}$*. Let there be a function* $r : D^{2\vec{\beta}} \to D^{2\vec{\gamma}}$ *such that:*

1. $\sigma_f \circ r = r \circ \sigma_g$.
2. $f \circ (r \times 1) = (r \times 1) \circ g$.
3. $f^{(i)} \sqsupseteq (r \times 1) \circ \perp$.

Then, for all $k \geq 0$*,*

$$(r \times 1) \circ g^{(k)} \sqsubseteq f^{(k+i)} \sqsubseteq (r \times 1) \circ g^{(k+i)}$$

Proof: We first prove the following.

$$f \circ (r \times 1) = (r \times 1) \circ g \Rightarrow \begin{cases} F \sqsubseteq (r \times 1) \circ G \Rightarrow \theta(f, F) \sqsubseteq (e \times 1) \circ \theta(g, G) \\ F \sqsupseteq (r \times 1) \circ G \Rightarrow \theta(f, F) \sqsupseteq (e \times 1) \circ \theta(g, G) \end{cases}$$

Assume $F \sqsubseteq (r \times 1) \circ G$. Then,

$$\begin{aligned} \theta(f, F) &= (\sigma_f \times 1) \circ f \circ \langle \pi' \circ F, 1 \rangle \\ &\sqsubseteq (\sigma_f \times 1) \circ f \circ \langle \pi' \circ (r \times 1) \circ G, 1 \rangle \\ &= (\sigma_f \times 1) \circ f \circ (r \times 1) \circ \langle \pi' \circ G, 1 \rangle \\ &= (\sigma_f \times 1) \circ (r \times 1) \circ g \circ \langle \pi' \circ G, 1 \rangle \\ &= (r \times 1) \circ (\sigma_g \times 1) \circ g \circ \langle \pi' \circ G, 1 \rangle \\ &= (r \times 1) \circ \theta(g, G) \end{aligned}$$

The proof of the other implication is symmetric and is omitted.

We first prove that $f^{(k+i)} \sqsupseteq g^{(k)}$, for all k by induction on k. From hypothesis, $f^{(i)} \sqsupseteq (r \times 1) \circ \perp = (r \times 1) \circ g^{(0)}$. For the induction step, note that

$$\begin{aligned} f^{(k+1+i)} &= \theta(f, f^{(k+i)}) \\ &\sqsupseteq (r \times 1) \circ \theta(g, g^{(k)}) \\ &= (r \times 1) \circ g^{(k+1)}) \end{aligned}$$

The proof of the other implication is symmetric and is omitted. ∎

Corollary 2 *With notation as in Lemma 7:*

1. $f \approx g$
2. $g \Downarrow_k \Rightarrow f \Downarrow_{k+i}$
3. *if* r *reflects order* $(r(x) \sqsubseteq r(y) \Rightarrow x \sqsubseteq y)$, *then* $f \Downarrow_{k+i} \Rightarrow g \Downarrow_{k+i}$.

Proof:

$$\begin{aligned} FB(f, \sigma_f) &= \pi_f(\bigsqcup f^{(k)}) \\ &= \bigsqcup \pi_f(f^{(k)}) \\ &= \bigsqcup \pi_f((r \times 1) \circ g^{(k)}) \\ &= \bigsqcup \pi_g(g^{(k)}) \\ &= FB(f, \sigma_f) \end{aligned}$$

For the second part

$$\begin{aligned} g \Downarrow_k &\Rightarrow g^{(k)} = g^{(k+2i)} \\ &\Rightarrow f^{(k+i)} = f^{(k+2i)} \\ &\Rightarrow f \Downarrow_{(k+i)} \end{aligned}$$

For the third part, if r reflects order,

$$\begin{aligned} f \Downarrow_{k+i} &\Rightarrow g^{(k+i)} = g^{(k+2i)} \\ &\Rightarrow g \Downarrow_{(k+i)} \end{aligned}$$

∎

Corollary 3 *Suppose* $f : D^{2\vec{\beta}+\vec{\alpha}} \to D^{2\vec{\beta}+\vec{\alpha}}$ *and* $g : D^{2\vec{\gamma}+\vec{\alpha}} \to D^{2\vec{\gamma}+\vec{\alpha}}$. *Let*

1. $2\beta = \vec{\beta}_1 + \vec{\beta}_2 + \vec{\beta}_3$, $2\vec{\gamma} = \vec{\gamma}_1 + \vec{\gamma}_2 + \vec{\gamma}_3$.
2. *Let* $r_1 : D^{\vec{\beta}_1} \to D^{\vec{\gamma}_1}$, $r_2 : D^{\vec{\beta}_2} \to D^{\vec{\gamma}_2}$, $r_3 : D^{\vec{\beta}_3} \to D^{\vec{\gamma}_3}$ *be retractions.*

3. $r : D^{2\vec{\beta}} \to D^{2\vec{\gamma}} = r_1 \times r_2 \times r_3$.

satisfy the following conditions:

1. $\sigma_f \circ r = r \circ \sigma_g$.
2. $\sigma_f \circ (1 \times 1 \times r_3) = (1 \times r_2 \times 1)$.
3. $f \circ (1 \times r_2 \times 1 \times 1) = (r \times 1) \circ g \circ (r_1^\star \times 1 \times r_3^\star \times 1)$.
4. $f \circ \vec{\bot} \sqsupseteq (1 \times 1 \times r_3 \times 1) \circ \vec{\bot}$.

The, the hypothesis of Corollary 2 are satisfied with $i = 2$.

Proof: Note that any retraction is an order monomorphism.

We now show that $f \circ (r \times 1) = (r \times 1) \circ g$.

$$\begin{aligned} f \circ (r \times 1) &= f \circ (1 \times r_2 \times 1 \times 1) \circ (r_1 \times 1 \times r_3 \times 1) \\ &= (r \times 1) \circ g \circ (r_1^\star \times 1 \times r_3^\star \times 1) \circ (r_1 \times 1 \times r_3 \times 1) \\ &= (r \times 1) \circ g \end{aligned}$$

Next, we show that $f^{(2)} \sqsupseteq (r \times 1) \circ \bot$.

$$\begin{aligned} f^{(1)} &= (\sigma_f \times 1) \circ f \circ \langle \pi' \circ f^{(0)}, 1 \rangle \\ &\sqsupseteq (\sigma_f \times 1) \circ f \circ \vec{\bot} \\ &\sqsupseteq (\sigma_f \times 1) \circ \vec{\bot} \times (r_3 \circ \vec{\bot}) \times \vec{\bot} \\ &= \vec{\bot} \times (r_2 \circ \vec{\bot}) \times \vec{\bot} \end{aligned}$$

Next,

$$\begin{aligned} f^{(2)} &= (\sigma_f \times 1) \circ f \circ \langle \pi' \circ f^{(1)}, 1 \rangle \\ &\sqsupseteq (\sigma_f \times 1) \circ f \circ (1 \times r_2 \times 1 \times 1) \circ (\vec{\bot} \times 1) \\ &\sqsupseteq (\sigma_f \times 1) \circ (r \times 1) \circ g \circ (r_1^\star \times 1 \times r_3^\star \times 1) \circ (\vec{\bot} \times 1) \\ &\sqsupseteq (r \times 1) \circ (\sigma_g \times 1) \circ g \circ (\vec{\bot} \times 1) \\ &= g^{(1)} \end{aligned}$$

∎

Rule	r			i
	Domain Indices	Function	Range Indices	
$\bot_\gamma\ (P) \cdot_\gamma U_\gamma$ $\to P$	$\left[\ \vec{\gamma}_P\ \right]$	$\left[\ (1, \bot_{\gamma_P}, \bot_{\gamma_U})\ \right]$	$\left[\ \vec{\gamma}_P + \vec{\gamma}_P + \vec{\gamma}_U\ \right]$	1
$⅋_\gamma^{\alpha,\beta}(P) \cdot_\gamma \otimes_\gamma^{\alpha,\beta}(Q,R)$ $\to P \cdot_\alpha Q \cdot_\beta R$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q + \vec{\gamma}_R \\ \alpha_P + \beta_P \\ \alpha_Q + \beta_R \end{bmatrix}$	$\begin{bmatrix} 1 \\ m \\ m \end{bmatrix}$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q + \vec{\gamma}_R \\ \gamma_P \\ \gamma_{QR} \end{bmatrix}$	1
$L_\gamma^\alpha(P) \cdot_\gamma \&_\gamma^{\alpha,\beta}(Q,R)$ $\to P \cdot_\alpha Q$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q \\ \alpha_P \\ \alpha_Q \end{bmatrix}$	$\begin{bmatrix} (1_P, 1_Q, \bot_R) \\ l \\ l \end{bmatrix}$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q + \vec{\gamma}_R \\ \gamma_P \\ \gamma_{QR} \end{bmatrix}$	2
$R_\gamma^\alpha(P) \cdot_\gamma \&_\gamma^{\alpha,\beta}(Q,R)$ $\to P \cdot_\alpha R$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_R \\ \alpha_P \\ \beta_R \end{bmatrix}$	$\begin{bmatrix} (1_P, \bot_Q, 1_R) \\ r \\ r \end{bmatrix}$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q + \vec{\gamma}_R \\ \gamma_P \\ \gamma_{QR} \end{bmatrix}$	2
$D_\gamma^\alpha(P) \cdot_\gamma !_\gamma^\alpha(Q)$ $\to P \cdot_\alpha Q$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q \\ \alpha_P \\ \alpha_Q \end{bmatrix}$	$\begin{bmatrix} 1 \\ d \\ d \end{bmatrix}$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q \\ \gamma_P \\ \gamma_Q \end{bmatrix}$	2
$W_\gamma(P) \cdot_\gamma !_\gamma^\alpha(Q)$ $\to W_{\vec{\alpha}}(P)$ where $\mathtt{FN}(Q) \setminus \{\alpha\} = \vec{\alpha}$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q \\ \gamma_P \\ \alpha_Q \end{bmatrix}$	$\begin{bmatrix} (1, \vec{w}) \\ w \\ w \end{bmatrix}$	$\begin{bmatrix} \vec{\gamma}_P + \vec{\gamma}_Q \\ \gamma_P \\ \gamma_Q \end{bmatrix}$	2
$C_\gamma^{\gamma',\gamma''}(P) \cdot_\gamma !_\gamma^\alpha(Q)$ $\to C_{\vec{\alpha}}^{\vec{\alpha'},\vec{\alpha''}}(P \cdot_{\gamma'} Q' \cdot_{\gamma''} Q'')$ where $\mathtt{FN}(Q) \setminus \{\alpha\} = \vec{\alpha}$ $Q' = !_{\gamma'}^\alpha(Q[\vec{\alpha'}/\vec{\alpha}])$ $Q'' = !_{\gamma''}^\alpha(Q[\vec{\alpha'}/\vec{\alpha}])$	$\begin{bmatrix} \vec{\gamma}_P \\ \vec{\gamma}_{Q'} + \vec{\gamma}_{Q''} \\ \gamma'_P + \gamma''_P \\ \gamma'_{Q'} + \gamma''_{Q''} \end{bmatrix}$	$\begin{bmatrix} 1 \\ \vec{c} \\ c \\ c \end{bmatrix}$	$\begin{bmatrix} \vec{\gamma}_P \\ \vec{\gamma}_Q \\ \gamma_P \\ \gamma_Q \end{bmatrix}$	2

Lemma 8 *For each instance $P \to Q$ of a reduction rule $R_2 \ldots R_8$, with $[\![P]\!] = f, [\![Q]\!] = g$, the conditions of Lemma 7 are satisfied with e, i defined as in above table.*

Proof: We present the proofs of two sample cases, below. These cases suffice to illustrate the use of lemma 7 and corollary 3 in the proofs.

(R3) For this case, $r = 1 \times m \times m$. It is immediate that $\sigma_f \circ r = r \circ \sigma_g$. Furthermore, r is a retraction. Note that

$$\begin{aligned} f \circ (r \times 1) &= (r \times 1) \circ ([\![P]\!] \times [\![Q]\!]) \circ (r^\star \times 1) \circ (r \times 1) \\ &= (r \times 1) \circ ([\![P]\!] \times [\![Q]\!]) \\ &= (r \times 1) \circ g \end{aligned}$$

Furthermore,

$$\begin{aligned} f^{(1)} &= (\sigma_f \times 1) \circ f \circ \langle \pi' \circ \perp, 1 \rangle \\ &\sqsupseteq (\sigma_f \times 1) \circ (r \times 1) \circ ([\![P]\!] \times [\![Q]\!]) \circ (r^\star \times 1) \circ \perp \\ &\sqsupseteq (r \times 1) \circ \perp \end{aligned}$$

(R4) For this case, we check the hypothesis of corollary 3. Let

$$\begin{aligned} r_1 &= (1_P, 1_Q, \perp_R) \\ r_3 &= D^{\alpha_P} \xrightarrow{l} D^{\gamma_P} \\ r_2 &= D^{\alpha_Q} \xrightarrow{l} D^{\gamma_Q} \end{aligned}$$

Note that r_1, r_2, r_3, e and σ_f, σ_g satisfy the conditions relating them in the hypothesis of corollary 3.

$$\begin{aligned} f \circ \vec{\perp} &\sqsupseteq ([\![L^\alpha_\gamma(P)]\!] \times \perp) \circ \vec{\perp} \\ &= ((1 \times l) \circ [\![P]\!] \circ (1 \times l^\star) \times \perp) \circ \vec{\perp} \\ &= ((1 \times r_3) \circ [\![P]\!] \circ (1 \times l^\star) \times \perp) \circ \vec{\perp} \\ &\sqsupseteq (1 \times 1 \times r_3 \times 1) \circ [\![P]\!] \circ \perp \\ &\sqsupseteq (1 \times 1 \times r_3 \times 1) \circ \perp \end{aligned}$$

Finally, we prove the sole remaining hypothesis of Corollary 3.

$$f \circ (1 \times r_2 \times 1 \times 1) = (r \times 1) \circ g \circ (r_1^\star \times 1 \times r_3^\star \times 1)$$

$$\begin{aligned} f \circ (1 \times r_2 \times 1 \times 1) &= [\![L^\alpha_\gamma(P)]\!] \times ([\![\&^{\alpha,\beta}_\gamma(Q, R)]\!] \circ (r_2 \times 1)) \\ &= ([\![L^\alpha_\gamma(P)]\!] \times ([\![\&^{\alpha,\beta}_\gamma(Q, R)]\!] \circ (1 \times r_2 \times 1)) \\ &= (1 \times r_3 \times 1) \circ [\![P]\!] \circ (1 \times r_3^\star \times 1) \times ([1_Q, \perp_R] \times r_2 \times 1) \circ [\![Q]\!] \circ (\pi_Q \times r_2^\star \times 1) \\ &= (r \times 1) \circ [\![g]\!] \circ (r_1^\star \times 1 \times r_3^\star \times 1) \end{aligned}$$

The proofs for cases (R5) through (R8) follow the proof described for the case (R4). ∎

Proving P4, R1

Reduction **(R1)** is reproduced below.

$$P \cdot_\alpha I_{\alpha,\beta} \to P[\beta/\alpha]$$

Lemma 9 *For all f,*

1. $f \cdot_\alpha I_{\alpha,\beta} \approx f[\beta/\alpha]$
2. $f[\beta/\alpha]\Downarrow_k \Longleftrightarrow f \cdot_\alpha I_{\alpha,\beta}\Downarrow_{k+2}$

Proof: Let $F = f \cdot_\alpha I_{\alpha,\beta}$ and $G = f$. We first set up notation to describe the iterates of $FB(F, \sigma_F)$ and $FB(G, \sigma_G)$. Note that we can write

$$\begin{aligned} F^{(k)}(\vec{x}, x) &= (a^{(k)}, b^{(k)}, \vec{v}^{(k)}, \vec{y}^{(k)}, y^{(k)}) \\ G^{(k)}(\vec{x}, x) &= (\vec{u}^{(k)}, \vec{x}^{(k)}, x^{(k)}) \end{aligned}$$

where the inductive definitions of $(a^{(k)}, b^{(k)}, \vec{v}^{(k)}, \vec{y}^{(k)}, y^{(k)})$ and $(\vec{u}^{(k)}, \vec{x}^{(k)}, x^{(k)})$ is as follows:

$$\begin{array}{ll} a^{(0)} = \bot,\ b^{(0)} = \bot,\ \vec{v}^{(0)} = \bot & \vec{u}^{(0)} = \bot \\ y^{(k+1)} = a^{(k)},\ b^{(k+1)} = x & (\vec{u}^{(k+1)}, \vec{x}^{(k+1)}, x^{(k+1)}) = f(\vec{u}^{(k)}, \vec{x}, x) \\ (\vec{v}^{(k+1)}, \vec{y}^{(k+1)}, a^{(k+1)}) = f(\vec{v}^{(k)}, \vec{x}, b^{(k)}) & \end{array}$$

We prove

$$(\forall k)\ [\vec{u}^{(k)}, \vec{x}^{(k)}, x^{(k)}) \sqsubseteq \vec{v}^{(k+1)}, \vec{y}^{(k+1)}, a^{(k+1)}) \sqsubseteq \vec{u}^{(k+1)}, \vec{x}^{(k+1)}, x^{(k+1)})$$

Note that this immediately implies that $G\Downarrow_k \Rightarrow F\Downarrow_{k+2}$. We prove the first inequality first. The base case is immediate. For the induction step:

$$\begin{aligned} (\vec{u}^{(k+1)}, \vec{x}^{(k+1)}, x^{(k+1)}) &= f(\vec{u}^{(k)}, \vec{x}, x) \\ &\sqsubseteq f(\vec{v}^{(k+1)}, \vec{x}, b_{k+1}) \\ &= (\vec{v}^{(k+2)}, x^{(\vec{k+2})}, a_{k+2}) \end{aligned}$$

Next, we prove the second inequality. We prove the result for all k by induction. The base case $(k = 0)$ is immediate. For the induction step,

$$\begin{aligned} (\vec{v}^{(k+1)}, \vec{y}^{(k+1)}, a^{(k+1)}) &= f(\vec{v}^{(k)}, \vec{x}, b^{(k)} \\ &\sqsubseteq f(\vec{v}^{(k)}, \vec{x}, x) \\ &\sqsubseteq f(\vec{u}^{(k)}, \vec{x}, x) \\ &= (\vec{u}^{(k+1)}, \vec{x}^{(k+1)}, x^{(k+1)}) \end{aligned}$$

Note that

$$\begin{aligned} FB(F,\sigma_F) &= \pi_F(\bigsqcup_k(a^{(k)}, b^{(k)}, \vec{v}^{(k)}, \vec{y}^{(k)}, y^{(k)})) \\ &= \bigsqcup_k \pi_F((a^{(k)}, b^{(k)}, \vec{v}^{(k)}, \vec{y}^{(k)}, y^{(k)}) \\ &= \bigsqcup_k \pi_G((\vec{v}^{(k)}, \vec{y}^{(k)}, y^{(k),a^{(k)}}) \\ &= \bigsqcup_k \pi_G((\vec{u}^{(k)}, \vec{x}^{(k)}, y^{(k),x^{(k)}}) \\ &= FB(G,\sigma_G) \end{aligned}$$

∎

Reduction rule (R9)

Reduction rule **(R9)** is reproduced below.

$$!^{\alpha}_{\gamma}(P) \cdot_{\delta} !^{\beta}_{\delta}(Q) {\rightarrow} !^{\alpha}_{\gamma}(P \cdot_{\delta} !^{\beta}_{\delta}(Q)), \text{ if } \delta \in \mathtt{FN}(P)$$

Let $f = [\![!^{\alpha}_{\gamma}(P) \cdot_{\delta} !^{\beta}_{\delta}(Q)]\!]$, $g = [\![!^{\alpha}_{\gamma}(P \cdot_{\delta} !^{\beta}_{\delta}(Q))]\!]$.

Lemma 10 *Let* $\pi_{\delta_P}([\![!^{\alpha}_{\gamma}(P)]\!](\vec{u}, \gamma_P \mapsto x, \delta_P \mapsto y)) = y'$. *Then,*

$$(\forall \vec{u}_P, \vec{v}_Q)\ f(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto y', \vec{v}_Q) = g(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto y', \vec{v}_Q)$$

Proof: Let the iterates obtained by unwinding the recursion in the definition of $[\![!^{\alpha}_{\gamma}(\cdot)]\!]$ in $[\![!^{\alpha}_{\gamma}(P)]\!]$ and $[\![!^{\alpha}_{\gamma}(P \cdot_{\delta} !^{\beta}_{\delta}(Q))]\!]$ be $\{h_i\}$ and $\{g_i\}$ respectively. Furthermore, let $f_i = h_i \cdot_{\delta} [\![!^{\beta}_{\delta}(Q)]\!]$.

From the continuity of all functions, it suffices to show that:

$$(\forall \vec{u}_P, \vec{v}_Q)\ f_i(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto y', \vec{v}_Q) = g_i(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto y', \vec{v}_Q)$$

where $\pi_{\delta_P}(h_i(\vec{u}, \gamma_P \mapsto x, \delta_P \mapsto y)) = y'$.

Proof proceeds by induction on i. Note that $h_0 = \perp$ and $g_0 = \perp$. Required result follows since: $(\forall \vec{u}_P, \vec{v}_Q)\ f_0(\vec{u}_P, \gamma_P \mapsto \perp, \delta_Q \mapsto \perp, \vec{v}_Q) = \vec{\perp}$.

Proof for the inductive step proceeds by case analysis on $e(x)$. If $e(x) = \perp$ or $e(x) = (\mathtt{in}_2(\perp))_{\perp}$, from definitions, we have $y' = x$ and

$$(\forall \vec{u}_P, \vec{v}_Q)\ f_{i+1}(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto x, \vec{v}_Q) = g_{i+1}(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto x, \vec{v}_Q) = \vec{x}$$

If $e(x) = (\mathtt{in}_1(x'))_{\perp}$, from definitions,

$$(\forall \vec{u})\ f_{i+1}(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto y', \vec{v}_Q) = g_{i+1}(\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto y', \vec{v}_Q) = \vec{z}$$

where $[\![P \cdot_{\delta} !^{\beta}_{\delta}(Q)]\!](\vec{u}_P, \gamma_P \mapsto x, \delta_Q \mapsto y', \vec{v}_Q) = \vec{z}$.

If $e(x) = \mathtt{in}_3(x_1, x_2)$, note that $\pi_{\delta_P}(h_{i+1}(\vec{u}, \gamma_P \mapsto x, \delta_P \mapsto u)) = e^\star(\mathtt{in}_3(y_1, y_2))_\perp$, where $\vec{c^\star}(\vec{u}) = (\vec{u}_1, \vec{u}_2), c^\star(u) = (u_1, u_2)$ and for $i = 1, 2$, $\pi_{\delta_P}(h_i(\vec{u}_i, \gamma_P \mapsto x_i, \delta_P \mapsto u_i)) = y_i$. Result follows from induction hypothesis on h_i, g_i, f_i and definitions. ∎

Recall the definition of θ from section 6.3:

$$\theta(h_1, h_2) = (\sigma_{h_1} \times 1) \circ h_1 \circ \langle \pi' \circ h_2, 1 \rangle$$

Thus, $f^{(k+1)} = \theta(f, f^{(k)})$, $g^{(k+1)} = \theta(g, g^{(k)})$. From Lemma 10:

Corollary 4 $g^{(k+2)} = \theta(f, g^{(k+1)})$

Lemma 11

1. $(\forall 1 \leq k)\ [g^{(k)} \sqsubseteq f^{(k+1)} \sqsubseteq g^{(k+1)}]$
2. $f \approx g$
3. $g \Downarrow_k \Rightarrow f \Downarrow_{k+1}$

Proof: From corollary 2, it suffices to prove the first part. Proof proceeds by induction on k. Simple computation using definitions verifies the base case. For the induction:

$$\begin{array}{rcl} f^{(k+2)} & = & \theta(f, f^{(k+1)}) \\ & \sqsubseteq & \theta(f, g^{(k+1)}) \\ & = & g^{(k+2)} \end{array}$$

The proof of the other implication is symmetric and is omitted. ∎

Combining Propositions 8, 9 and Lemmas 8, 9 and 6.3, we obtain our main result.

Theorem 3 (Soundness) *$\mathcal{A}$ is an LRA.*

7 The structure of Cut-free proofs

Throughout this section, we will work exclusively with the $\mathcal{GI}$ interpretation based on $D = W_\perp^\infty(\Sigma, X)$, where X is some infinite set. We will give striking description of the interpretation of cut-free proofs, as certain very special kinds of partial involutions. This will clarify how our interpretation generalises the treatment of the multiplicatives described in Section 2. It will also be used to make a connection between the $\mathcal{GI}$ interpretation and a denotational semantics based on coherence spaces.

We begin with some preliminary notions. If D is an algebraic domain, we write $K(D)$ for the set of compact elements of D. We write

$$\uparrow(b) = \{d \in D \mid b \sqsubseteq d\}$$

and S_k for the set of all permutations on $\{1, \ldots, k\}$. Now, given a continuous function $f : D \to D$, we define

$$\mathtt{dom}(f) = \mathtt{Int}\{d \in D \mid f^2 d = d\}$$

Here $\mathtt{Int}$ is the topological interior operator, defined with respect to the Scott topology. Concretely,

$$\mathtt{Int}(U) = \bigcup\{\uparrow(b) \mid \uparrow(b) \subseteq U, b \in K(D)\}$$

Thus, $\mathtt{dom}(f)$ is the largest open subset of D^n on which f is an involution. We write $\mathcal{E}(f) : D^n \rightharpoonup D^n$ for the partial function on D^n obtained by restricting f to $\mathtt{dom}(f)$.

$$\mathcal{E}(f) = f \restriction \mathtt{dom}(f)$$

If f is a partial involution on D ($f^3 = f$), then $\mathcal{E}(f)$ (the "extension of f") is the restriction of f to the observable part of its subdomain of definition.

Now, consider $t \in K(W^{\infty}_{\perp}(\Sigma, X))$, *i.e.* a finite tree. We fix some standard way of enumerating the $\perp$-leaves of such a tree (say, left-to-right). Then, we can define an operation $t[t_1, \ldots, t_k]$ of grafting trees $t_1, \ldots, t_k$ on the k $\perp$-leaves of t. We recall that, for any $u \in W^{\infty}_{\perp}(\Sigma, X)$,

$$t \sqsubseteq u \iff \exists! t_1, \ldots, t_k . (u = t[t_1, \ldots, t_k]).$$

Now, given $\sigma \in S_k$, we define $p_{t,\sigma} : W^{\infty}_{\perp}(\Sigma, X) \rightharpoonup W^{\infty}_{\perp}(\Sigma, X)$ by

$$p_{t,\sigma}(u) = \begin{cases} t[t_{\sigma(1)}, \ldots, t_{\sigma(k)}], & u = t[t_1, \ldots t_k] \\ \text{undefined}, & t \not\sqsubseteq u \end{cases}$$

Note that $p_{t,\sigma}$ is an automorphism on its domain of definition $\uparrow t$, with inverse $p_{t,\sigma^{-1}}$; it is an involution if σ is.

These notions extend in an obvious way to $\vec{t} \in W^{\infty}_{\perp}(\Sigma, X)^n$; so that we can define $p_{\vec{t},\sigma}$. Now, suppose we are given a family $\{(\vec{t_i}, \sigma_i)\}_{i \in I}$, where $\vec{t_i} \in K(D^n)$, for some fixed n, for all $i \in I$. Such a family is *pairwise disjoint* if

$$i \neq j \Rightarrow (\uparrow\vec{t_i} \cap \uparrow\vec{t_j} = \varnothing).$$

Then, we can define $\sum_{i \in I} p_{\vec{t}_i, \sigma_i}$ as the partial function on D^n with domain $\bigcup \uparrow\vec{t_i}$ obtained by glueing all the $p_{\vec{t}_i, \sigma_i}$ together.

7.1 Shape semantics

Fix an infinite set of generators X. We say that $\vec{t} \in W(\Sigma, X)^n$ is *linear* if every "variable" = generator from X occurring in $\vec{t}$ does so exactly twice. We will interpret proofs in $\mathtt{CLL}_2$ as sets of linear term tuples; we call this the "shape semantics". It can also be seen as related to the notion of *slices* of a proof net introduced in [Gir87]; think of the tuple of terms as a set of trees growing back from the conclusions of a proof net, with leaves joined up in pairs by axiom links as specified by the occurrences of variables. Our notion has two important differences as compared with the slices in [Gir87]; we slice !-boxes, and we interpret cut "compositionally", in a denotational style. Note that we are reverting to the positional notation of section 3.

$$
\begin{array}{rcl}
\mathcal{S}[\![I]\!] & = & \{x, x \mid x \in X\} \\
\mathcal{S}[\![Ex_\sigma(P)]\!] & = & \{t_{\sigma(1)}, \ldots, t_{\sigma(k)} \mid t_1, \ldots, t_k \in \mathcal{S}[\![P]\!]\} \\
\mathcal{S}[\![U]\!] & = & \{\mathtt{u}\} \\
\mathcal{S}[\![\bot(P)]\!] & = & \{\vec{t}, \mathtt{u} \mid \vec{t} \in \mathcal{S}[\![P]\!]\} \\
\mathcal{S}[\![\otimes(P,Q)]\!] & = & \{\vec{t}, \vec{u}, \mathtt{m}(t,u) \mid \vec{t}, t \in \mathcal{S}[\![P]\!], \vec{u}, u \in \mathcal{S}[\![Q]\!]\} \\
\mathcal{S}[\![\wp(P)]\!] & = & \{\vec{t}, \mathtt{m}(t,u) \mid \vec{t}, t, u \in \mathcal{S}[\![P]\!]\} \\
\mathcal{S}[\![T]\!] & = & \varnothing \\
\mathcal{S}[\![L(P)]\!] & = & \{\vec{t}, \mathtt{l}(t) \mid \vec{t}, t \in \mathcal{S}[\![P]\!]\} \\
\mathcal{S}[\![R(Q)]\!] & = & \{\vec{t}, \mathtt{r}(t) \mid \vec{t}, t \in \mathcal{S}[\![Q]\!]\} \\
\mathcal{S}[\![\&(P,Q)]\!] & = & \{\vec{t}, \mathtt{l}(t) \mid \vec{t}, t \in \mathcal{S}[\![P]\!]\} \cup \{\vec{t}, \mathtt{r}(t) \mid \vec{t}, t \in \mathcal{S}[\![Q]\!]\} \\
\mathcal{S}[\![D(P)]\!] & = & \{\vec{t}, \mathtt{d}(t) \mid \vec{t}, t \in \mathcal{S}[\![P]\!]\} \\
\mathcal{S}[\![W(P)]\!] & = & \{\vec{t}, \mathtt{w} \mid \vec{t} \in \mathcal{S}[\![P]\!]\} \\
\mathcal{S}[\![C(P)]\!] & = & \{\vec{t}, \mathtt{c}(t,u) \mid \vec{t}, t, u \in \mathcal{S}[\![P]\!]\} \\
\mathcal{S}[\![!(P)]\!] & = & \{\vec{t}, \mathtt{d}(t) \mid \vec{t}, t \in \mathcal{S}[\![P]\!]\} \cup \{\vec{\mathtt{w}}, \mathtt{w}\} \cup \{\mathtt{c}(\vec{t}, \vec{u}), \mathtt{c}(t,u) \mid \vec{t}, t \in \mathcal{S}[\![!(P)]\!] \wedge \vec{u}, u \in \mathcal{S}[\![!(P)]\!]\} \\
\mathcal{S}[\![\forall(P)]\!] & = & \mathcal{S}[\![P]\!] \\
\mathcal{S}[\![\exists(P)]\!] & = & \mathcal{S}[\![P]\!]
\end{array}
$$

The definition of $\mathcal{S}[\![!(P)]\!]$ is a (monotone) induction; of course the least fixpoint is intended. Note that linearity constraints are imposed tacitly in the definition of $\otimes(P,Q)$; the variables occurring in $\vec{t}, t$ and $\vec{u}, u$ must be disjoint.

We complete the definition with the semantics of Cut which uses *unification*. We write $S = \mathcal{U}(t, t')$ if t, t' are unifiable, with most general unifying substitution S, and $S(\vec{t})$ for the application of a substitution to a tuple of terms.

$$\mathcal{S}[\![P \cdot Q]\!] = \{S(\vec{t}, \vec{u}) \mid \exists t, u.\ \vec{t}, t \in \mathcal{S}[\![P]\!], \vec{u}, u \in \mathcal{S}[\![Q]\!], S = \mathcal{U}(t,u)\}$$

Note that here again we are implicitly requiring the variables in $\vec{t}, t$ and $\vec{u}, u$ to be disjoint. To understand why we need the full power of unification, the reader should consider axiom contractions with non-atomic axioms; or, more essentially, second-order quantifier contractions.

Now, given a linear term tuple $\vec{t}$, we define

$$p_{\vec{t}} = p_{\vec{u},\sigma}$$

where $\vec{u}$ is obtained by replacing all variables in $\vec{t}$ by $\perp$, and σ is the fixpoint-free involution on the $2k$ $\perp$-leaves of $\vec{u}$ (where the variables occurring in $\vec{t}$ are $x_1, \ldots, x_k$) obtained by transposing each pair of $\perp$-leaves in $\vec{u}$ corresponding to the two occurrences of x_i in $\vec{t}$, $i = 1, \ldots, k$. Consider the following example.

$$\text{twist} = \cfrac{\cfrac{\cfrac{\vdash \alpha^{\perp}, \alpha \qquad \vdash \alpha^{\perp}, \alpha}{\vdash \alpha^{\perp}, \alpha^{\perp}, \alpha \otimes \alpha}}{Ex\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \vdash \alpha^{\perp}, \alpha^{\perp}, \alpha \otimes \alpha}}{\vdash \alpha^{\perp} \mathbin{⅋} \alpha^{\perp}, \alpha \otimes \alpha}$$

Then, we have

$$\begin{array}{rcl} \mathcal{S}[\![\text{twist}]\!] & = & \{\mathtt{m}(x, y), \mathtt{m}(y, x) \mid x, y \in X\} \\ p_{\mathtt{m}(x,y),\mathtt{m}(y,x)} & = & p_{b,\sigma} \\ b & = & \mathtt{m}(\perp_1, \perp_2), \mathtt{m}(\perp_3, \perp_4) \\ \sigma & = & \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix} \\ p_{b,\sigma}(\mathtt{m}(a, b), \mathtt{m}(c, d)) & = & \mathtt{m}(d, c), \mathtt{m}(b, a) \end{array}$$

We can now state the main result of this section. This result can be seen as giving a precise formulation of the idea of "genericity" or "communication without understanding" discussed at the end of [Gir89a]. A proof must analyze the structure of the data through which it communicates with its environment up to a fixed depth determined by its type; beyond that, it merely permutes data to achieve a certain flow of information, without any regard as to its structure.

Theorem 4 *Let f be the $\mathcal{GI}$ interpretation of a* $\mathtt{CLL}_2$ *proof P in $W^{\infty}_{\perp}(\Sigma, \varnothing)$. Then*

$$\mathcal{E}(FB(f, \sigma_f)) = \sum \{p_{\vec{t}} \mid \vec{t} \in \mathcal{S}[\![P]\!]\}$$

Proof: We now embark on the proof of this theorem. Our first step is to reformulate the semantics in terms of records rather than sets, just as we did for the $\mathcal{GI}$ interpretation in Section 6. Thus, for example the equation defining $\mathcal{S}[\![\otimes(P, Q)]\!]$ is replaced by

$$\mathcal{S}[\![\otimes^{\alpha,\beta}_{\gamma}(P, Q)]\!] = \{[\vec{\alpha} \mapsto \vec{t}, \vec{\beta} \mapsto \vec{u}, \gamma \mapsto \mathtt{m}(t, u) \mid [\vec{\alpha} \mapsto \vec{t}, \alpha \mapsto t] \in \mathcal{S}[\![P]\!], [\vec{\beta} \mapsto \vec{u}, \beta \mapsto u] \in \mathcal{S}[\![Q]\!]\}$$

We rely on the reader to supply the remaining details.

Lemma 12 *Let $P \rightarrow Q$ be an instance of the reductions* **(R1)** – **(R9)**. *Then*

$$\mathcal{S}[\![P]\!] = \mathcal{S}[\![Q]\!]$$

Proof: First, we verify **(R1)**. Let x does not occur in $\vec{t}$. Then, $S = \mathcal{U}(t,x) = \{(x,t)\}$.

$$\begin{aligned}
\mathcal{S}[\![P \cdot_\alpha I_{\alpha,\beta}]\!] &= \{S(\vec{t}, \beta \mapsto x) \mid \exists t.\ [\vec{t}, \alpha \mapsto t] \in \mathcal{S}[\![P]\!], [\alpha \mapsto x, \beta \mapsto x] \in \mathcal{S}[\![Q]\!], S = \mathcal{U}(t,x)\} \\
&= \{\vec{t}, \beta \mapsto t \mid \ \vec{t}, t \in \mathcal{S}[\![P]\!]\} \\
&= \mathcal{S}[\![P[\beta/\alpha]]\!]
\end{aligned}$$

Next, we verify **(R4)**. Let $\mathtt{h} \in \{\mathtt{l}, \mathtt{r}\}$. Let $[\vec{t}, \gamma \mapsto \mathtt{l}(t)]$ and $[\vec{u}, \gamma \mapsto \mathtt{h}(u)]$ satisfy the linearity constraint. If $\mathtt{h} = \mathtt{r}$ the two terms are not unifiable. If $\mathtt{h} = \mathtt{l}$, $S = \mathcal{U}(\mathtt{l}(t), \mathtt{h}(u)) = \mathcal{U}(t,u)$.

$$\begin{aligned}
\mathcal{S}[\![L^\alpha_\gamma(P) \cdot_\gamma \&^{\alpha,\beta}_\gamma(Q,R)]\!] &= \{S(\vec{t}, \vec{u}) \mid \exists t, u.\ [\vec{t}, \mathtt{l}(t)] \in \mathcal{S}[\![L^\alpha_\gamma(P)]\!], [\vec{u}, \mathtt{h}(u)] \in \mathcal{S}[\![\&^{\alpha,\beta}_\gamma(Q,R)]\!], S = \mathcal{U}(\mathtt{l}(t), \mathtt{h}(u))\} \\
&= \{S(\vec{t}, \vec{u}) \mid \exists t, u.\ [\vec{t}, \alpha \mapsto t] \in \mathcal{S}[\![P]\!], [\vec{u}, \alpha \mapsto u] \in \mathcal{S}[\![Q]\!], S = \mathcal{U}(t,u)\} \\
&= \mathcal{S}[\![P \cdot_\alpha Q]\!]
\end{aligned}$$

The other cases are similar and are omitted. ∎

Now we define a predicate l on LRA terms P:

$$l\ P \overset{\text{def}}{\iff} \mathcal{E}(FB(f, \sigma_f)) = \sum\{p_{\vec{t}} \mid \vec{t} \in \mathcal{S}[\![P]\!]\}$$

where f is the $\mathcal{GI}$ interpretation of P.

Lemma 13 *l is closed under constructors,* i.e. *for each k-ary constructor ω, $l\ P_1, \ldots, l\ P_k \Rightarrow l\ \omega(P_1, \ldots, P_k)$.*

Proof: Let f_i be the $\mathcal{GI}$ interpretation of P and let $g_i = FB(f_i, \sigma_{f_i})$. From Corollary 1 in Section 6,

$$FB(\omega(f_1, \ldots f_k), \sigma_{\omega(f_1,\ldots,f_k)}) = \omega(FB(f_1, \sigma_{f_1}), \ldots FB(f_k, \sigma_{f_k}))$$

Thus, it suffices to prove $\mathcal{E}(\omega(g_1, \ldots, g_k)) = \sum\{p_{\vec{t}} \mid \vec{t} \in \mathcal{S}[\![\omega(P_1, \ldots, P_k)]\!]\}$.

Consider $\omega \in \{U, \bot, \otimes, ⅋, L, R, D, W, C\}$. Then, there is a $\mathtt{r} \in \Sigma$ and an associated retraction r such that: $[\![\omega(g_1, \ldots g_k)]\!] = (1 \times r) \circ (g_1 \times \ldots \times g_k) \circ (1 \times r^\star)$. Thus,

$$\mathtt{dom}[\![(\omega(g_1, \ldots g_k))]\!] = \{\uparrow(\vec{u}_1, \ldots, \vec{u}_k, r(u_1, \ldots, u_k)) \mid \forall 1 \le i \le n. \uparrow(\vec{u}_i, u_i) \subseteq \mathtt{dom}(g_i)\}$$

Furthermore, from definition of $\mathcal{S}[\![\cdot]\!]$,

$$\mathcal{S}[\![\omega(P_1, \ldots P_k)]\!] = \{\vec{t}_1, \ldots, \vec{t}_k, \mathtt{r}(t_1, \ldots, t_k) \mid \forall 1 \le i \le n. \vec{t}_i, t_i \in \mathcal{S}(g_i)\}$$

Let $\vec{t}_1, \ldots, \vec{t}_k, r(t_1, \ldots, t_k) \in \mathcal{S}[\![\omega(P_1, \ldots P_k)]\!]$. Note that $\vec{t}_i, t_i$ have pairwise disjoint set of variables. Thus, the domain of definition of the partial function $p_{\vec{t}_1, \ldots, \vec{t}_k, r(t_1, \ldots, t_k)}$ is

$$\{\vec{u}_1, \ldots, \vec{u}_k, r(u_1, \ldots, u_k) \mid \forall 1 \le i \le n.\ p_{\vec{t}_i, t_i} \text{ is defined on } \vec{u}_i, u_i\}$$

Thus the domains of definition of the partial functions $\mathcal{E}(g)$ and $\sum\{p_{\vec{t}} \mid \vec{t} \in \mathcal{S}[\![\omega(P_1,\ldots,P_k)]\!]\}$ agree. If $p_{\vec{t}_1,\ldots,\vec{t}_k,r(t_1,\ldots,t_k)}$ is defined on $\vec{u}_1,\ldots,\vec{u}_k, r(u_1,\ldots,u_k)$,

$$p_{\vec{t}_1,\ldots,\vec{t}_k,\mathbf{r}(t_1,\ldots,t_k)}(\vec{u}_1,\ldots,\vec{u}_k,r(u_1,\ldots,u_k)) = \prod_{i=1}^{k}(1\times r)\circ p_{\vec{t}_i,t_i}\circ(1\times r^\star)$$

where $p_{\vec{t}_i,t_i}(\vec{u}_i,u_i) = (\vec{u}'_i,u'_i)$. Combining these observations:

$$\begin{aligned}
\sum\{p_{\vec{t}} \mid \vec{t}\in\mathcal{S}[\![\omega(P_1,\ldots,P_k)]\!]\} &= \sum\{(1\times r)\circ\prod_{i=1}^{k}p_{\vec{t}_i,t_i}\circ(1\times r^\star) \mid p_{\vec{t}_i,t_i}\in\mathcal{S}[\![P_i]\!]\}\} \\
&= (1\times r)\circ\prod_{i=1}^{k}\sum\{p_{\vec{t}_i,t_i} \mid \vec{t}_i,t_i\in\mathcal{S}[\![P_i]\!]\}\circ(1\times r^\star) \\
&= (1\times r)\circ\mathcal{E}(g_1)\times\cdots\times\mathcal{E}(g_k)\circ(1\times r^\star) \\
&= \mathcal{E}(\omega(g_1,\ldots,g_k))
\end{aligned}$$

We consider & next. Let $g = \&_{\gamma}^{\alpha,\beta}(g_1,g_2)$ and $d = [\vec{x},\gamma\mapsto x]$. Then,

$$g^2d = d \Rightarrow \begin{cases} x = \perp, \vec{x} = \vec{\perp} \\ \vee(x = l(y)\wedge g_1^2([\vec{x},\alpha\mapsto y]) = [\vec{x},\alpha\mapsto y]) \\ \vee(x = r(y)\wedge g_2^2([\vec{x},\alpha\mapsto y]) = [\vec{x},\beta\mapsto y]) \end{cases}$$

Recall that $\mathtt{dom}(g)$ is defined as

$$\mathtt{dom}(g) = \mathtt{Int}\{d\in D \mid g^2d = d\}$$

Thus, $\mathtt{dom}(g) = \{[\vec{x},\gamma\mapsto l(y)] \mid [\vec{x},\alpha\mapsto y]\in\mathtt{dom}(g_1)\}\cup\{[\vec{x},\gamma\mapsto l(y)] \mid [\vec{x},\beta\mapsto y]\in\mathtt{dom}(g_2)\}$ and

$$\mathcal{E}(g)[\vec{x},\gamma\mapsto x] = \begin{cases} [\vec{x}',\gamma\mapsto l(y')], & x = l(y)\wedge\ \mathcal{E}(g_1)[\vec{x},\alpha\mapsto y] = [\vec{x}',\alpha\mapsto y'] \\ [\vec{x}',\gamma\mapsto r(y')], & x = r(y)\wedge\ \mathcal{E}(g_2)[\vec{x},\beta\mapsto y] = [\vec{x}',\beta\mapsto y'] \end{cases}$$

Result now follows from induction hypothesis on g_1, g_2 and definition of $\mathcal{S}[\![\&_{\gamma}^{\alpha,\beta}(P_1,P_2)]\!]$.

The proof for ! is similar and is omitted. ∎

Let $[P]_\xi$ denote the equivalence class of P under α-conversion. We can define a LRA A_l as follows:

- The interpretation of an LRA term is its ξ-equivalence class $[P]_\xi$.
- $[P]_\xi \approx_l [Q]_\xi \stackrel{\text{def}}{\Longleftrightarrow} FB(f,\sigma_f) = FB(g,\sigma_g)$, where f, g are the $\mathcal{GI}$ interpretations of P, Q respectively.
- $[P]_\xi{\Downarrow_l} \Longleftrightarrow l\ P$

Proposition 10 *A_l is an LRA.*

Proof: **P1** is built into the definition; **P0** is an immediate corollary of the contents of Section 6. **P3** follows from Lemma 13. Next, we prove **P2**. Suppose $P \rightarrow Q$ is an instance of any of **R1**–**R9**, and that $Q\Downarrow_l$. Then, $FB(f, \sigma_f) = FB(g, \sigma_g)$. So

$$\begin{array}{rcll} \mathcal{E}(FB(f,\sigma_f)) & = & \mathcal{E}(FB(g,\sigma_g)) & \\ & = & \sum\{p_{\vec{t}} \mid \vec{t} \in \mathcal{S}[\![Q]\!]\} & \text{By Assumption} \\ & = & \sum\{p_{\vec{t}} \mid \vec{t} \in \mathcal{S}[\![P]\!]\} & \text{By Lemma 12} \end{array}$$

P4 is proved similarly. ∎

Proof of Theorem 4: Immediate from Proposition 10 and Theorem 2. This proof shows the power and versatility of Theorem 2. The authors do not know any other direct argument. ∎

The remainder of this section is devoted to drawing out some important consequences of Theorem 4.

7.2 Correspondence with Cut-elimination

As we saw in Section 6, the $\mathcal{GI}$ interpretation fell short of full correspondence with cut-elimination in that the commutative conversions **R10** and **R11** for the additives are not valid with respect to the congruence $\approx$. (Recall that $f \approx g \Longleftrightarrow FB(f, \sigma_f) = FB(g, \sigma_g)$.) We will now use Theorem 4 to show that we can define a new LRA with the coarser congruence:

$$f \approx_{\mathcal{E}} g \stackrel{\text{def}}{\Longleftrightarrow} \mathcal{E}(FB(f,\sigma_f)) = \mathcal{E}(FB(g,\sigma_g))$$

which does validate **R10** and **R11**.

Firstly, a lemma.

Lemma 14 *If* $\sum_{i \in I} p_{\vec{t}_i,\sigma_i} = \sum_{j \in J} p_{\vec{t}_j,\sigma_j}$, $\{(\vec{t}_i, \sigma_i)\}_{i \in I} = \{(\vec{t}_j, \sigma_j)\}_{j \in J}$.

Proof: We write $\vec{u}_k$ for the element obtained by replacing all variables in $\vec{t}_k$ by $\perp$. Let

$$\begin{array}{rcl} S_1 & = & \{\vec{t}_i \mid p_{\vec{t}_i,\sigma_i} \text{ is a summand in } \sum_{i \in I} p_{\vec{t}_i,\sigma_i}\} \\ S_2 & = & \{\vec{t}_j \mid p_{\vec{t}_j,\sigma_i} \text{ is a summand in } \sum_{j \in J} p_{\vec{t}_j,\sigma_j}\} \end{array}$$

Let $\vec{t}_i \in S_1$. From hypothesis of lemma, there is a $j \in J$ such that $\vec{u}_j \sqsubseteq \vec{u}_i$. Furthermore, there is a $i' \in I$ such that $\vec{u}_{i'} \sqsubseteq \vec{u}_j$. Thus, $\vec{u}_{i'} \sqsubseteq \vec{u}_j \sqsubseteq \vec{u}_i$. But $\vec{u}_{i'} \sqsubseteq \vec{u}_i$ implies $i = i'$. Hence $\vec{u}_j = \vec{u}_i$. This implies that $(\vec{t}_i, \sigma_i) = (\vec{t}_j, \sigma_j)$ ∎

As an immediate corollary, we have:

Proposition 11 *If P, Q are LRA terms typable in* $\mathtt{CLL}_2$, *with $\mathcal{GI}$ interpretations f, g respectively, then:*

$$f \approx_{\mathcal{E}} g \Longleftrightarrow \mathcal{S}[\![P]\!] = \mathcal{S}[\![Q]\!]$$

Proof: Result got by combining Theorem 4 and Lemma 14. ∎

Now, we define an LRA $A_{\mathcal{E}}$ as follows:

- The interpretation of an LRA term is its ξ-equivalence class $[P]_\xi$.
- $[P]_\xi \approx_{\mathcal{E}} [Q]_\xi \stackrel{\text{def}}{\iff} f \approx_{\mathcal{E}} g$, where f, g are the $\mathcal{GI}$interpretations of P, Q respectively.
- $[P]_\xi \Downarrow_l \iff f\Downarrow$

Proposition 12 *$A_{\mathcal{E}}$ is an LRA; if $P \rightarrow Q$ is an instance of* **R10** *and* **R11**, *then $A_{\mathcal{E}}[\![P]\!] \approx_{\mathcal{E}} A_{\mathcal{E}}[\![Q]\!]$.*

Proof: **P1** is built into definitions. **P3** follows from the results of Section 6. **P0**, **P2** and **P4** follow from Lemma 12 and Proposition 11. ∎

We can now state the main result of this section, which follows immediately from Proposition 12.

Theorem 5 *Let Π be a proof of a sequent $\vdash [\Delta], \Gamma$ in $\mathtt{CLL}_2$, with $[\![\Pi]\!] = f$. Then, if Π reduces to Π' by any sequence of contractions, with $[\![\Pi']\!] = g$, then*

$$\mathcal{E}(FB(f, \sigma_f)) \approx_{\mathcal{E}} \mathcal{E}(FB(g, \sigma_g))$$

In particular, if Π' is any cut-free proof obtained from Π by cut-elimination, then $\mathcal{E}(FB(f, \sigma_f)) = \mathcal{E}(g)$.

7.3 Connection with Coherence spaces

Finally, we sketch how Theorem 4 can be used to connect the Geometry of Interaction with a denotational semantics based on coherence spaces [Gir87].

To proceed, we need some additional notions. Let Σ^+, Σ^- be isomorphic copies of the signature Σ, so that for each f in Σ we have f^+ in Σ^+ and f^- in Σ^-. We define $(f^+)^\star = f^-$ and $(f^-)^\star = f^+$. We can extend this in an obvious way to an involution $(\cdot)^\star$ on $W(\Sigma^+ \cup \Sigma^-, X^+ \cup X^-)$. Defining $\perp^\star = \perp$, this extends to a continuous involution on $W_\perp^\infty(\Sigma^+ \cup \Sigma^-, X^+ \cup X^-)$. We can reformulate all our previous work in terms of this refined algebra. The idea is that each operation now carries a "polarity" indicating which side of the duality it originates from. So, for example, we use $\mathtt{m}^+$ for Tensor and $\mathtt{m}^-$ for Par. This extra information is mere decoration as far as the Geometry of Interaction is concerned, but carrying it around will help us to make a precise correspondence with the denotational semantics. For example, the message exchange function $\sigma_f : D^{2m} \rightarrow D^{2m}$ is now defined by:

$$\sigma_f(t_1, u_1, \ldots, t_m, u_m) = (u_1^\star, t_1^\star, \ldots, u_m^\star, t_m^\star)$$

We shall consider a semantics based on a "universal domain" X defined by the domain equation

$$\begin{aligned} X \quad = \quad & (X \otimes X) \,\&\, (X \wp X) \,\&\, I \,\&\, \perp \,\&\, (X \& X) \\ & \&\, (X \oplus X) \,\&\, !X \,\&\, ?X \end{aligned} \tag{1}$$

Analyzing it in the spirit of the first author's "Domain theory in logical form" [Abr91a], we find that the tokens of the web of X and the coherence relation are defined inductively as follows: The constructors on tokens are elements of $\{u^+, u^-, m^+, m^-, l^+, r^+, l^-, r^-, \{^+\ldots\}^+, \{^-\ldots\}^-\}$. Also, given constructors ω_1, ω_2, we have:

$$\omega_1 \neq \omega_2, \{\omega_1, \omega_2\} \neq \{l^+, r^+\} \Rightarrow \omega_1(t_1, \ldots, t_n) \frown \omega_2(u_1, \ldots, u_m)$$

The detailed construction is presented below.

Tokens	*Coherence*
u^+, u^-	
$m^+(t_1, t_2), m^-(t_1, t_2)$	$t_1 \stackrel{\frown}{\smile} u_1, t_2 \stackrel{\frown}{\smile} u_2 \Rightarrow m^+(t_1, t_2) \stackrel{\frown}{\smile} m^+(u_1, u_2)$ $t_1 \frown u_1 \vee\ t_2 \stackrel{\frown}{\smile} u_2 \Rightarrow m^-(t_1, t_2) \frown m^+(u_1, u_2)$
$l^+(t), r^+(t), l^-(t), r^-(t)$	$t \stackrel{\frown}{\smile} u \Rightarrow \omega(t) \stackrel{\frown}{\smile} \omega(u),\ \omega \in \{l^+, r^+, l^-, r^-\}$
$t_i \frown t_j, 1 \leq i < j \leq n \Rightarrow \{^+t_1, \ldots, t_n\}^+$ $t_i \smile t_j, 1 \leq i < j \leq n \Rightarrow \{^-t_1, \ldots, t_n\}^-$	$(\forall i, j.\ t_i \stackrel{\frown}{\smile} u_j) \Rightarrow \{^+t_1, \ldots, t_n\}^+ \stackrel{\frown}{\smile} \{^+u_1, \ldots, u_m\}^+$ $(\exists i, j.\ t_i \smile u_j) \Rightarrow \{^-t_1, \ldots, t_n\}^- \frown \{^-u_1, \ldots, u_m\}^-$

A semantic function $\mathcal{D}$ mapping $\mathtt{CLL}_2$ proofs of sequents $\vdash A_1, \ldots, A_k$ into points of $\parr_{i=1}^k X$ can be given along the much the same lines as that in [Gir87]. The main difference is that axioms are interpreted by the identity relation on $|X|$, while the quantifiers are interpreted trivially. This semantics uses the definitions of the exponentials given in [Gir87], based on sets rather than trees. Consider the functors

$$\begin{aligned} F(A) &= A \,\&\, I \,\&\, (F(A) \otimes F(A)) \\ G(A) &= A \oplus \bot \oplus (G(A) \parr G(A)) \end{aligned}$$

and the domain Y defined by

$$\begin{aligned} Y = {} & (Y \otimes Y) \,\&\, (Y \parr Y) \,\&\, I \,\&\, \bot \,\&\, (Y \& Y) \\ & \&\, (Y \oplus Y) \,\&\, F(Y) \,\&\, G(Y) \end{aligned} \tag{2}$$

Again, analyzing Y the spirit of the first author's "Domain theory in logical form" [Abr91a], we find that the tokens of the web of Y and the coherence relation are defined inductively as follows: The constructors on tokens are elements of $\Sigma^+ \cup \Sigma-$. Also, given constructors ω_1, ω_2, we have:

$$\omega_1 \neq \omega_2, \{\omega_1, \omega_2\} \notin \{\{l^+, r^+\}, \{d^+, c^+\}, \{d^-, c^-\}, \{w^-, c^-\}\} \Rightarrow \omega_1(t_1, \ldots, t_n) \frown \omega_2(u_1, \ldots, u_m)$$

The detailed construction is presented below.

Tokens	*Coherence*
u^+, u^-	
$m^+(t_1, t_2), m^-(t_1, t_2)$	$t_1 \overset{\frown}{\smile} u_1, t_2 \overset{\frown}{\smile} u_2 \Rightarrow m^+(t_1, t_2) \overset{\frown}{\smile} m^+(u_1, u_2)$ $t_1 \frown u_1 \vee\ t_2 \overset{\frown}{\smile} u_2 \Rightarrow m^-(t_1, t_2) \frown m^+(u_1, u_2)$
$l^+(t), r^+(t), l^-(t), r^-(t)$	$t \overset{\frown}{\smile} u \Rightarrow \omega^+(t) \overset{\frown}{\smile} \omega^+(u),\ \omega \in \{l^+, r^+, l^-, r^-\}$
$w^+, w^-, d^+(t), d^-(t)$ $t \overset{\frown}{\smile} u \Rightarrow c^+(t, u)$ $t \smile u \Rightarrow c^-(t, u)$	$t \overset{\frown}{\smile} u \Rightarrow \omega(t) \overset{\frown}{\smile} \omega(u),\ \omega \in \{d^+, d^-,\}$ $(\forall i = 1, 2.\, t_i \overset{\frown}{\smile} u_i) \Rightarrow c^+(t_1, u_1) \overset{\frown}{\smile} c^+(t_2, u_2)$ $(\exists i, j \in \{1, 2\}).\, t_i \frown u_j) \Rightarrow c^-(t_1, u_1) \overset{\frown}{\smile} c^-(t_2, u_2)$ $t \overset{\frown}{\smile} u_1, t \overset{\frown}{\smile} u_2 \Rightarrow d^+(t) \overset{\frown}{\smile} c^+(u_1, u_2)$ $t \smile u_1 \vee\ t \smile u_2 \Rightarrow d^-(t) \frown c^-(u_1, u_2)$

A semantics $\mathcal{D}_0$ based on Y rather than X can be given, by modifying $\mathcal{D}$ to work on trees rather than sets for the exponentials. Moreover, for any A, there are clearly canonical maps

$$F(A) \to !A \qquad G(A) \to ?A$$

Since the functors used in the domain equations (1), (2) are covariant, these maps can be lifted to a canonical map $h : Y \to X$. More precisely, h is the unique function satisfying:

$$h(\omega(\vec{x})) \quad = \quad \omega(\vec{h}(\vec{x})) \quad \text{if } \omega \in \{u^+, u^-, l^+, l^-, r^+, r^{-1}, m^+, m^-\}$$

$$
\begin{array}{rcll}
h(\omega(\vec{x})) & = & \{^+\vec{h}(\vec{x})\}^+, & \text{if } \omega \in \{d^+, w^+\} \\
h(\omega(\vec{x})) & = & \{^-\vec{h}(\vec{x})\}^-, & \text{if } \omega \in \{d^-, w^-\} \\
h(c^+(x_1, x_2)) & = & \{^+t_1, \ldots, t_n, u_1, \ldots, u_m\}^+, & \text{if } h(x_1) = \{^+t_1, \ldots, t_n\}^+, h(x_2) = \{^+u_1, \ldots, u_m\}^+ \\
h(c^-(x_1, x_2)) & = & \{^-t_1, \ldots, t_n, u_1, \ldots, u_m\}^-, & h(x_1) = \{^-t_1, \ldots, t_n\}^-, h(x_2) = \{^-u_1, \ldots, u_m\}^-
\end{array}
$$

Now, a simple structural induction on P yields:

Proposition 13 *The canonical map h is a homomorphism of the semantics: for any* $\mathtt{CLL}_2$ *proof P,*

$$h \circ \mathcal{D}_0[\![P]\!] = \mathcal{D}[\![P]\!]$$

Now we consider the refined version $\mathcal{S}^\star$ of the shape semantics based on $W(\Sigma^+ \cup \Sigma^-, X^+ \cup X^-)$. For example, the clause for Par becomes

$$\mathcal{S}^\star[\![\parr(P)]\!] = \{\vec{t}, \mathtt{m}^-(t, u) \mid \vec{t}, t, u \in \mathcal{S}^\star[\![P]\!]\}$$

Note that $\mathcal{S}^\star$ works with linear term tuples, where x occurs once as x^+, once as x^-; substitutions must respect polarities, *i.e.* if we substitute t for x^+, we must substitute $t^\star$ for x^-.

Proposition 14 *For every proof P in* $\mathtt{CLL}_2$*, $\mathcal{D}_0[\![P]\!]$ is the set of ground instances of $\mathcal{S}^\star[\![P]\!]$.*

Proof: Proof proceeds by structural induction. We prove the inductive case for $\parr$. The other inductive cases are similar and are omitted

$$\mathcal{S}^\star[\![\parr(P)]\!] = \{\vec{t}, \mathtt{m}^-(t, u) \mid \vec{t}, t, u \in \mathcal{S}^\star[\![P]\!]\}$$

Let S be the set of ground instances. Then,

$$
\begin{array}{rcl}
S & = & \{\vec{u}, \mathtt{m}^-(u_1, u_2) \mid \vec{u}, \mathtt{m}^-(u_1, u_2) \text{ is a ground instance of } \vec{t}, \mathtt{m}^-(t_1, t_2) \in \mathcal{S}^\star[\![\parr(P)]\!]\} \\
& = & \{\vec{u}, \mathtt{m}^-(u_1, u_2) \mid \vec{u}, u_1, u_2 \text{ is a ground instance of } \vec{t}, t_1, t_2 \in \mathcal{S}^\star[\![P]\!]\} \\
& = & \{\vec{u}, \mathtt{m}^-(u_1, u_2) \mid \vec{u}, u_1, u_2 \in \mathcal{D}_0[\![P]\!]\} \\
& = & [\![\parr P]\!]
\end{array}
$$

∎

Proposition 15 *Let f be the interpretation of a* $\mathtt{CLL}_2$ *proof P. The set of all ground instances of $\mathcal{S}^\star[\![P]\!]$ is exactly the set of $\vec{t} \in W(\Sigma^+ \cup \Sigma^-, \varnothing)^n$ such that $FB(f, \sigma_f)(\vec{t}) = \vec{t}$.*

Proof: Using Theorem 4 it suffices to show that the set of all ground instances of $\mathcal{S}^\star[\![P]\!]$ is exactly the set of $\vec{t} \in W(\Sigma^+ \cup \Sigma^-, \varnothing)^n$ such that $\vec{t}$ is in the domain of definition of $\sum\{p_{\vec{t}} \mid \vec{t} \in \mathcal{S}[\![P]\!]\}$. This is immediate from the definitions. ∎

Theorem 6 *For any* $\mathtt{CLL}_2$ *proof* P*, with* $\mathcal{GI}$ *interpretation* f*,*

$$\mathcal{D}[\![P]\!] = h(\{\vec{t} \in W(\Sigma^+ \cup \Sigma^-, \emptyset)^n \mid FB(f, \sigma_f)(\vec{t}) = \vec{t}\})$$

Proof: Combining Propositions 13, 14 and 15. ∎

It should be emphasised that the denotational semantics of P has been recovered from the *function* f, without any reference to P.

References

[Abr90] S. Abramsky. The lazy λ-calculus. In D. Turner, editor, *Research Topics in Functional Programming*, pages 65–117. Addison Wesley, 1990.

[Abr91a] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

[Abr91b] S. Abramsky. Proofs as processes. Unpublished Lecture, 1991.

[Abr93] S. Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 1993. To appear. Revised version of Imperial College Technical Report DoC 90/20, October 1990.

[AKP80] Arvind, V. Kathail, and K. Pingali. A dataflow architecture with tagged tokens. In *Proceedings of International Conference on Circuits and Computers*, 1980.

[Bar84] H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* North-Holland, revised edition, 1984.

[BH90] H. Barendregt and K. Hemerik. Types in lambda calculi and programming languages. In *Proceedings of ESOP 90*, 1990.

[Coh81] P. M. Cohn. *Universal Algebra.* D. Reidel, 1981.

[Den74] J. B Dennis. *First version of a data flow procedural language*, volume 19 of *Lecture Notes in Computer Science.* Springer-Verlag, 1974.

[FS91] P. J. Freyd and A. Scedrov. *Categories, Allegories*, volume 39 of *North-Holland Mathematical Library.* Elsevier Science Publishers, 1991.

[Gir87] J.-Y. Girard. Linear Logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[Gir88a] J.-Y. Girard. Geometry of interaction 2: Deadlock-free algorithms. In P. Martin-Löf and G. Mints, editors, *International Conference on Computer Logic, COLOG 88*, pages 76–93. Springer-Verlag, 1988. Lecture Notes in Computer Science 417.

[Gir88b] J.-Y. Girard. Mulitplicatives. *Rendiconti del seminario matematico dell' università e politecnico di Torino*, 1988. Special issue on Logic and Computer Science.

[Gir89a] J.-Y. Girard. Geometry of interaction 1: Interpretation of System F. In R. Ferro et al., editor, *Logic Colloquium 88*. North Holland, 1989.

[Gir89b] J.-Y. Girard. Towards a geometry of interaction. In J. W. Gray and A. Scedrov, editors, *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 69–108. American Mathematical Society, 1989.

[GTWW77] J. A. Goguen, J. W. Thatcher, E. G. Wagner, and J. B. Wright. Initial algebra semantics and continuous algebras. *Journal of the ACM*, 24:68–95, 1977.

[Gue81] I. Guessarian. *Algebraic Semantics*, volume 99 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1981.

[Kah77] G. Kahn. The semantics of a simple language for parallel programming. In *Information Processing 74*, pages 993–998. North-Holland, 1977.

[KM77] G. Kahn and D. MacQueen. Coroutines and networks of parallel proceses. In *Proceedings of IFIP Congress*. North-Holland, 1977.

[KP78] G. Kahn and G. Plotkin. Domaines concrets. Technical Report 336, IRIA-Laboria, 1978.

[Mil90] R. Milner. Functions as processes. In *Proceedings of ICALP 90*, volume 443 of *Lecture Notes in Computer Science*, pages 167–180. Springer-Verlag, 1990.

[Mil91] R. Milner. The polyadic π-calculus: A tutorial. Technical report, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, 1991.

[Mog86] E. Moggi. Categories of partial morphisms and the λ_p-calculus. In D. Pitt, S. Abramsky, A. Poigné, and D. Rydeheard, editors, *Category Theory and Computer Programming*, pages 242–251. Springer-Verlag, 1986. *LNCS* Vol. 240.

[Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–93, 1991.

[MPW89] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes. Technical report, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, 1989.

[Sco76] D. S. Scott. Data types as lattices. *SIAM J. Computing*, 5:522–587, 1976.

[Sco80] D. S. Scott. Relating theories of lambda calculus. In J. R. Hindley and J. P. Seldin, editors, *To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus and Formalism*, pages 403–450. Academic Press, 1980.

[See89] R. Seely. Linear logic, $\star$-autonomous categories and cofree coalgebras. In J. W. Gray and A. Scedrov, editors, *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 371–382. American Mathematical Society, 1989.

[Wal89] R. F. C. Walters. Datatypes in distributive categories. *Bulletin of the Australian Mathematical Society*, 40:79–82, 1989.