Predicate Transformers and Linear Logic, yet another denotational model

Pierre Hyvernat

▶ **To cite this version:**

Pierre Hyvernat. Predicate Transformers and Linear Logic, yet another denotational model. Computer Science and Logic 2004, Sep 2004, Karpacz, Poland. pp.115–129. hal-00387490

HAL Id: hal-00387490
https://hal.science/hal-00387490

Submitted on 25 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Predicate transformers and Linear Logic

yet another Denotational Model

Pierre Hyvernat[1,2]

[1] Institut mathématique de Luminy, Marseille, France
[2] Chalmers Institute of Technology, Göteborg, Sweden
hyvernat@iml.univ-mrs.fr

Abstract. In the refinement calculus, monotonic predicate transformers are used to model specifications for (imperative) programs. Together with a natural notion of simulation, they form a category enjoying many algebraic properties.

We build on this structure to make predicate transformers into a denotational model of full linear logic: all the logical constructions have a natural interpretation in terms of predicate transformers (*i.e.* in terms of specifications). We then interpret proofs of a formula by a safety property for the corresponding specification.

Introduction

The first denotational model for linear logic was the category of *coherent spaces* ([1]). In this model, formulas are interpreted by graphs; and proofs by *cliques* (complete subgraphs). This forms a special case of domain *à la* Scott.

From a conceptual point of view, the construction of interfaces is a little different: first, the model looks a little more dynamic; then, *seeds* —the notion corresponding to cliques— are not closed under substructures; and finally, they are closed under arbitrary unions (usually, only directed unions are allowed).

What was a little unexpected is that the interpretation of linear proofs used in the relational model can be lifted directly to this structure to yield a denotational model of full linear logic in the spirit of _/hyper/multi-coherence or finiteness spaces.

A promising direction for further research is to explore the links between the model presented below and non-determinism as it appears both in the differential lambda-calculus ([2,3]) and different kind of process calculi. We expect such a link because of the following remarks: this model comes from the semantics of imperative languages; it can be extended to a model of the differential lambda calculus (which can be seen as a variant of "lambda calculus with resource") and there is a completely isomorphic category in which predicate transformers are replaced by (two-sided) transition systems. In particular, all of the logical operations presented below have natural interpretations in terms of processes...

1 Relations and Predicate Transformers

Definition 1. *A relation r between two sets is a subset of their cartesian product. We write $r^\sim$ for the* converse *relation: $r^\sim = \{(b,a) \mid (a,b) \in r\}$.*
The composition of two relations $r \subseteq A \times B$ and $r' \subseteq B \times C$ is defined by $r' \cdot r = \{(a,c) \mid (\exists b \in B)\ (a,b) \in r \wedge (b,c) \in r'\}$.
If X is a set, $\mathbf{Id}_X$ denotes the identity on X, i.e. $\mathbf{Id}_X = \{(a,a) \mid a \in X\}$.

There seems to be three main notions of morphisms between sets. These give rise to three important categories in computer science:

- **Set**, where morphisms are functions;
- **Rel**, where morphisms are (binary) relations;
- **Pow**, where morphisms are monotonic *predicate transformers.*

One can go from **Set** to **Rel** and from **Rel** to **Pow** using the same categorical construction ([4]) which cannot be applied further.

Definition 2. *A predicate transformer from A to B is a function from $\mathcal{P}(A)$ to $\mathcal{P}(B)$. A predicate transformer P is monotonic if $x \subseteq x'$ implies $P(x) \subseteq P(x')$.*

From now on, we will consider only monotonic predicate transformers. The adjective "monotonic" is thus implicit everywhere.

The term "predicate" might not be the most adequate but the terminology was introduced by E. Dijkstra some decades ago, and has been used extensively by computer scientists since then. Formally, a predicate on a set A can be identified with a subset of A by the separation axiom of ZF set theory; the confusion is thus harmless.

Definition 3. *If r is a relation between A and B, we write $\langle r \rangle : \mathcal{P}(A) \to \mathcal{P}(B)$ for the following predicate transformer: (called the direct image of r)*

$$\langle r \rangle(x) = \{b \in B \mid (\exists a \in A)\ (a,b) \in r \wedge a \in x\}\ .$$

Note that in the traditional version of the refinement calculus ([5]), our $\langle r \rangle$ is written $\{r^\sim\}$, but this notation clashes with set theoretic notation and would make our formulas very verbose with $\_^\sim$ everywhere.

2 Interfaces

Several denotational models of linear logic can be seen as "refinements" of the relational model. This very crude model interprets formulas by sets; and proofs by subsets. It is degenerate in the sense that any formula is identified with its linear negation! Coherent spaces ([1]), hypercoherent spaces ([6]), finiteness spaces ([7]) remove (part of) this degeneracy by adding structure on top of the relational model. We follow the same approach:

Definition 4. *An* interface *X is given by a set $|X|$ (called the* state space*) and a predicate transformer P_X on $|X|$ (called the* specification*).*

The term "specification" comes from computer science, where a specification usually takes the form:

> *if the program is started in a state satisfying ϕ, it will terminate; and the final state will satisfy ψ.*

Such a specification can be identified with the (monotonic) predicate transformer $\psi \mapsto$ "biggest such ϕ". This point of view is that of the **wp** calculus, introduced by Dijkstra ("**wp**" stands for "weakest precondition"). Note that the specification "goes backward in time": it associates to a set of final states (which we want to reach) a set of initial states (which guarantee that we will reach our goal).[3]

For a complete introduction to the field of predicate transformers in relation to specifications, we refer to [5].

In the coherence semantics, a "point" is a complete subgraph,[4] called a *clique.* Since the intuitions behind our objects are quite different, we change the terminology.

Definition 5. *Let X be an interface, a subset $x \subseteq |X|$ is called a* seed *of X if $x \subseteq P_X(x)$. We write $\mathcal{S}(X)$ for the collection of seeds of X.*

More traditional names for seeds are safety properties, or P-invariant properties: if some initial state is in x, no matter what, after each execution of a program satisfying specification P, the final state will still be in x. In other words, P maintains an invariant, namely "staying in x". In particular, there can be no program deadlock when starting from x.

The collection of cliques in the (hyper)coherent semantics forms a c.p.o.: the sup of any directed family exists. The collection of seeds in an interface satisfies the stronger property:

Lemma 1. *For any interface X, $\big(\mathcal{S}(X), \subseteq\big)$ is a complete sup-lattice.*

Proof. $\emptyset$ is trivially a seed; and by monotonicity of P, a union of seeds is a seed. □

The fact that seeds are closed under union may seem counter-intuitive at first; but one possible interpretation is that we allow for non-deterministic data. For example, all denotational models of linear logic have an object for the booleans: its state space is $\{t, f\}$, and the cliques are always $\emptyset$, $\{t\}$ and $\{f\}$. The union of $\{t\}$ and $\{f\}$ is usually not itself a clique because "one cannot get both true and false". However, if one interprets union as a non-deterministic sum, then $\{t, f\}$ is a perfectly sensible set of data.

However, nothing guarantees that a seed is the unions of all its finite subseeds; a given seed needs not even contain any finite seed!. (The canonical example being $P_X(x) = X$, with X infinite.)

[3] In a previous version, interfaces also had to enjoy the property $P(\emptyset) = \emptyset$ and $P(|X|) = |X|$. This condition doesn't interact well with second order interpretation and has thus been dropped.

[4] The intuition is that a set of data is coherent iff it is pairwise coherent.

3 Constructions on Interfaces

A denotational model interprets formulas as objects in a category (and proofs as morphisms). We thus need to define all the constructions of linear logic at the level of interfaces. The most interesting cases are the linear negation and the tensor product (and the exponentials, but they will be treated in section 6).

Note that there will always be an "ambient" set A for predicates. We write $\overline{x}$ for the A-complement of x.

Let $X = (|X|, P_X)$ and $Y = (|Y|, P_Y)$ be two interfaces;

Definition 6. *The* dual *of X is defined as $(|X|, P_X^\perp)$ where $P_X^\perp(x) = \overline{P_X(\overline{x})}$. We write it $X^\perp$. An* antiseed *of X is a seed in $X^\perp$.*

In terms of specifications, $a \in P^\perp(x)$ means *"if the program is started in a, and if execution terminates, the final state will be in x"*. If P is concerned with **wp** calculus, then $P^\perp$ is concerned with **wlp** calculus. (Weakest liberal precondition, also introduced by Dijkstra.)

This operation of "negation" is the reason we do not ask for any properties on the predicate transformer. It respects neither continuity nor commutation properties! In many respects, this operation is not very well-behaved.

Definition 7. *The* tensor *of X and Y is the interface $(|X| \times |Y|, P_X \otimes P_Y)$ where $P_X \otimes P_Y(r)$ is the predicate transformer*

$$r \mapsto \bigcup_{x \times y \subseteq r} P_X(x) \times P_Y(y) \ .$$

We write it $X \otimes Y$.

$P_X \otimes P_Y$ is the most natural transformer to construct on $|X| \times |Y|$. It was used in [8] to model parallel execution of independent pieces of programs. The intuition is the following: a program satisfies $P_X \otimes P_Y$ if, when you start it in the pair $(a_i, b_i) \in P_X \otimes P_Y(r)$ of initial states, the two final states will be related through r. In particular, this means that execution is *synchronous*: both executions need to terminate.

Definition 8. *The* with *of X and Y is the interface $(|X| + |Y|, P_X \,\&\, P_Y)$ where $P_X \,\&\, P_Y(x, y) = \big(P_X(x), P_Y(y)\big)$.*[5] *We write it $X \,\&\, Y$.*

This operation is not very interesting from the specification point of view: it is a kind of disjoint union.

Definition 9. *The other connectives are defined as usual:*

- $\mathbf{0} = (\emptyset, \mathbf{Id})$; $\top = \mathbf{0}^\perp$; $\mathbf{1} = (\{*\}, \mathbf{Id})$; $\bot = \mathbf{1}^\perp$;
- $X \oplus Y$ *(*plus*) is the interface* $\big(X^\perp \,\&\, Y^\perp\big)^\perp$;

[5] it uses implicitly the fact that $\mathcal{P}(|X| + |Y|) \simeq \mathcal{P}(|X|) \times \mathcal{P}(|Y|)$

– $X \parr Y$ *(*par*) is the interface* $\big(X^\perp \otimes Y^\perp\big)^\perp$*;*
– $X \multimap Y$ *is the interface* $X^\perp \parr Y$*.*

We have:

Lemma 2. $\bot = \mathbf{1}$*;* $\top = \mathbf{0}$ *and* $X \oplus Y = X \,\&\, Y$*.*

The proof is immediate. The first two equalities are satisfied in several of the denotational models of LL; the second one is a little less common. (For example, it is satisfied in finiteness spaces, but in no ...-coherence spaces.)

As an application of the definitions, let's massage the definition of $A \multimap B$ into something readable:
$(a,b) \in A \multimap B(r)$
$\quad\Leftrightarrow$ { definition }
$(a,b) \in \big(A^\perp \parr B\big)(r)$
$\quad\Leftrightarrow$ { definition, involutivity of $\_^\perp$ }
$(a,b) \in \big(A \otimes B^\perp\big)^\perp(r)$
$\quad\Leftrightarrow$ { definition of $\_^\perp$ }
$(a,b) \notin A \otimes B^\perp(\overline{r})$
$\quad\Leftrightarrow$ { definition of $\otimes$ }
$\neg\big((\exists x \times y \subseteq \overline{r})\ a \in A(x) \wedge b \in B^\perp(y)\big)$
$\quad\Leftrightarrow$ { logic }
$(\forall x \times y \subseteq \overline{r})\ a \notin A(x) \vee b \notin B^\perp(y)$
$\quad\Leftrightarrow$ { logic }
$(\forall x \times y \subseteq \overline{r})\ a \in A(x) \Rightarrow b \in B(\overline{y})$
$\quad\Leftrightarrow$ { lemma: $x \times y \subseteq \overline{r}$ iff $\langle r\rangle x \subseteq \overline{y}$ }
$(\forall \langle r\rangle x \subseteq \overline{y})\ a \in A(x) \Rightarrow b \in B(\overline{y})$
$\quad\Leftrightarrow$ { change of variable: $y \mapsto \overline{y}$ }
$(\forall \langle r\rangle x \subseteq y)\ a \in A(x) \Rightarrow b \in B(y)$.

From this, we derive:

Lemma 3. $(a,b) \in A \multimap B(r)$ *iff* $a \in A(x) \Rightarrow b \in B(\langle r\rangle x)$ *for all* $x \subseteq |X|$*.*
For any interface X*,* $\mathbf{Id}_{|X|} \in \mathcal{S}(X \multimap X)$*.*

The shapes of images along $X \parr Y$ are usually difficult to visualize, but we have the following on "rectangles":

Lemma 4. *Let* X *and* Y *be interfaces; then for all* $x \subseteq |X|$ *and* $y \subseteq |Y|$ *we have:* $P_X \otimes P_Y(x \times y) = P_X(x) \times P_Y(y) \subseteq P_X \parr P_Y(x \times y)$*.*

Proof. That $P_X \otimes P_Y(x \times y) = P_X(x) \times P_Y(y)$ is straightforward.
Suppose now $a \in P_X(x)$ and $b \in P_Y(y)$, let's show that $(a,b) \in P_X \parr P_Y(x \times y)$:
suppose $x' \times y' \subseteq \overline{x \times y}$
$\quad\Rightarrow$ { claim (see below) }
$x \subseteq \overline{x'} \vee y \subseteq \overline{y'}$
$\quad\Rightarrow$ { monotonicity }
$a \in P_X(\overline{x'}) \vee b \in P_Y(\overline{y'})$.

Claim: $x' \times y' \subseteq \overline{x \times y} \Rightarrow x \subseteq \overline{x'} \vee y \subseteq \overline{y'}$
Proof of claim: suppose $\neg(x \subseteq \overline{x'}) \wedge \neg(y \subseteq \overline{y'})$
$\Rightarrow x \cap x' \neq \emptyset \wedge y \cap y' \neq \emptyset$
$\Rightarrow x \times y \cap x' \times y' \neq \emptyset$
$\Rightarrow \neg(x' \times y' \subseteq \overline{x \times y})$. □

Furthermore, seeds in A and B are related to seeds in $A \otimes B$ and $A \parr B$ in the following way:

Lemma 5. *Let A and B be interfaces. We have:*

(i) if $x \in \mathcal{S}(A)$ and $y \in \mathcal{S}(B)$ then $x \times y \in \mathcal{S}(A \otimes B)$;
(ii) if $x \in \mathcal{S}(A)$ and $y \in \mathcal{S}(B)$ then $x \times y \in \mathcal{S}(A \parr B)$.

Proof. The first point is obvious; the second point is a direct consequence of Lemma 4. □

4 Linear Proofs and Seeds

The previous section gave a way to interpret any linear formula F by a interface F^*. (When no confusion arises, F^* is written F.) We now interpret linear proofs of F as subsets of the state space of F^*.[6] We refer to [1] or the abundant literature on the subject for the motivations governing those inference rules.

(1) If π is $\dfrac{}{\vdash \mathbf{1}}$ then $\pi^* = \{*\}$;

(2) if π is $\dfrac{}{\vdash \Gamma, \top}$ then $\pi^* = \emptyset$;

(3) if π is $\dfrac{\pi_1 \vdash \Gamma}{\vdash \Gamma, \bot}$ then $\pi^* = \big\{(\gamma, *) \mid \gamma \in \pi_1^*\big\}$;

(4) if π is $\dfrac{\pi_1 \vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}$ then $\pi^* = \big\{\big(\gamma, (a, b)\big) \mid (\gamma, a, b) \in \pi_1^*\big\}$;

(5) if π is $\dfrac{\pi_1 \vdash \Gamma, A \qquad \pi_2 \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}$
then $\pi^* = \pi_1^* \otimes \pi_2^* = \big\{\big(\gamma, \delta, (a, b)\big) \mid (\gamma, a) \in \pi_1^* \wedge (\delta, b) \in \pi_2^*\big\}$;

(6) if π is $\dfrac{\pi_1 \vdash \Gamma, A}{\vdash \Gamma, A \oplus B}$ then $\pi^* = \big\{\big(\gamma, (1, a)\big) \mid (\gamma, a) \in \pi_1^*\big\}$;

[6] recall that a sequent $A_1, \ldots A_n$ is interpreted by $A_1 \parr \ldots A_n$ and the notation $\pi \vdash \Gamma$ means "π is a proof of sequent Γ"

(7) if π is $\dfrac{\pi_1 \vdash \Gamma, B}{\vdash \Gamma, A \oplus B}$ then $\pi^* = \big\{\big(\gamma, (2,b)\big) \mid (\gamma, b) \in \pi_1^*\big\}$;

(8) if π is $\dfrac{\pi_1 \vdash \Gamma, A \qquad \pi_2 \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}$

then π^* is $\big\{\big(\gamma, (1,a)\big) | (\gamma, a) \in \pi_1^*\big\} \cup \big\{(\gamma, (2,b)) | \big(\gamma, b\big) \in \pi_2^*\big\}$;

(9) if π is $\dfrac{\pi_1 \vdash \Gamma, A \qquad \pi_2 \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta}$

then $\pi^* = \big\{(\gamma, \delta) \mid (\exists\, a)\ (\gamma, a) \in \pi_1^* \wedge (\delta, a) \in \pi_2^*\big\}$.

This interpretation is correct in the following sense:

Proposition 1. *If π a proof of F, then π^* is a seed in F^*.*

Proof. By induction on the structure of π: we will check that seeds propagate through the above constructions. It is mostly trivial computation, except for two interesting cases:

(5): suppose that π_1 is a seed in $\Gamma \parr A$ and that π_2 is a seed in $\Delta \parr B$. We need to show that $\pi_1 \otimes \pi_2 = \big\{\big(\gamma, \delta, (a,b)\big) \mid (\gamma, a) \in \pi_1 \wedge (\delta, b) \in \pi_2\big\}$ is a seed in the sequent $\Gamma \parr \Delta \parr (A \otimes B)$.
Let $\big(\gamma, \delta, (a,b)\big) \in \pi_1 \otimes \pi_2$
$\Leftrightarrow$
$(\gamma, a) \in \pi_1$ and $(\delta, b) \in \pi_2$
$\Rightarrow$ { π_1 and π_2 are seeds in Γ, A and Δ, B }
$(\gamma, a) \in \Gamma, A(\pi_1)$ and $(\delta, \pi_2) \in \Delta, B(\pi_2)$.
By contradiction, let $\big(\gamma, \delta, (a,b)\big) \notin \Gamma, \Delta, A \otimes B(\pi_1 \otimes \pi_2)$
$\Rightarrow$
$\big(\gamma, \delta, (a,b)\big) \in \Gamma^\perp \otimes \Delta^\perp \otimes (A \otimes B)^\perp(\overline{\pi_1 \otimes \pi_2})$
$\Rightarrow$ { for some $u \times v \times r \subseteq \overline{\pi_1 \otimes \pi_2}$: }
$\gamma \in \Gamma^\perp(u) \wedge \delta \in \Delta^\perp(v) \wedge \underbrace{(a,b) \in (A \otimes B)^\perp(r)}$
$\Rightarrow$
$\ldots \wedge \Big((\forall x \times y \subseteq \overline{r})\ a \in A^\perp(\overline{x}) \vee b \in B^\perp(\overline{y})\Big)$.

In particular, define $x = \langle \pi_1 \rangle u$ and $y = \langle \pi_2 \rangle v$; it is easy to show that $x \times y \subseteq \overline{r}$, so that we have $a \in A^\perp(\overline{x})$ or $b \in B^\perp(\overline{y})$.
Suppose $a \in A^\perp(\overline{x})$: we have $\gamma \in \Gamma^\perp(u)$ and $u \times \overline{x} \subseteq \overline{\pi_1}$ (easy lemma); so by definition, $(\gamma, a) \in \Gamma^\perp \otimes A^\perp(\overline{\pi_1})$, *i.e.* $(\gamma, a) \notin \Gamma, A(\pi_1)$! This is a contradiction.
Similarly, one can derive a contradiction from $b \in B^\perp(\overline{y})$.
This finishes the proof that $\pi_1 \otimes \pi_2$ is a seed of $\Gamma, \Delta, A \otimes B$.

(9): let π_1 be a seed in $\Gamma, A = \Gamma^\perp \multimap A$ and π_2 a seed in $\Delta, A^\perp$, *i.e.* $\pi_2^\sim$ is a seed in $A \multimap \Delta$. Let's show that $\pi = \big\{(\gamma, \delta) \mid (\exists a)\ (\gamma, a) \in \pi_1 \wedge (\delta, a) \in \pi_2\big\} = \pi_2^\sim \cdot \pi_1$ is a seed in Γ, Δ.
Suppose $(\gamma, \delta) \in \pi_2^\sim \cdot \pi_1$, *i.e.* that $(\gamma, a) \in \pi_1$ and $(a, \delta) \in \pi_2^\sim$ for some a. We will prove that (γ, δ) is in $\Gamma, \Delta(\pi) = \Gamma^\perp \multimap \Delta(\pi)$. According to Lemma 3, we need to show that if $\gamma \in \Gamma^\perp(u)$ then $\delta \in \Delta(\langle \pi \rangle u)$.

Let $\gamma \in \Gamma^{\perp}(u)$
$\Rightarrow$ $\{$ $(\gamma, a) \in \pi_1 \subseteq \Gamma^{\perp} \multimap A(\pi_1)$ $\}$
$a \in A\big(\langle \pi_1 \rangle u\big)$
$\Rightarrow$ $\{$ $(a, \delta) \in \pi_2^{\sim} \subseteq A \multimap \Delta(\pi_2^{\sim})$ $\}$
$\delta \in \Delta\big(\langle \pi_2^{\sim} \rangle \langle \pi_1 \rangle u\big)$
$\Leftrightarrow$
$\delta \in \Delta\big(\langle \pi \rangle u\big)$. □

5 Morphisms, Categorical Structure

To complete the formal definition of a category of interfaces, we need to define morphisms between interfaces. This is done in the usual way:

Definition 10. *A linear arrow from X to Y is a seed in $X \multimap Y$.*

Here is a nicer characterization of linear arrows from X to Y:

Lemma 6. *$r \in \mathcal{S}(X \multimap Y)$ iff $\langle r \rangle\big(P_X(x)\big) \subseteq P_Y\big(\langle r \rangle(x)\big)$ for all $x \subseteq |X|$.*

Proof. Suppose r is a seed in $X \multimap Y$, let $b \in \langle r \rangle P_X(x)$
$\Rightarrow$
there is some a s.t. $(a, b) \in r$ and $a \in P_X(x)$
$\Rightarrow$ $\{$ r is a seed in $X \multimap Y$ $\}$
$(a, b) \in P_X \multimap P_Y(r)$
$\Rightarrow$ $\{$ definition of $\multimap$ $\}$
$b \in P_Y(\langle r \rangle x)$.
Conversely, suppose $\langle r \rangle P_X(x) \subseteq P_Y \langle r \rangle(x)$; let $(a, b) \in r$, and $a \in P_X(x)$. We have $b \in \langle r \rangle P_X(x)$, and by hypothesis, $b \in P_Y(\langle r \rangle x)$. □

Lemma 7. *If $r \in \mathcal{S}(X \multimap Y)$ and $r' \in \mathcal{S}(Y \multimap Z)$ then $r' \cdot r \in \mathcal{S}(X \multimap Z)$.*

Proof. This is the essence of point *(9)* from Proposition 1; or a simple corollary to Lemma 6. □

Taken together with Lemma 3, this makes interfaces into a category:

Definition 11. *We write* **Int** *for the category with interfaces as objects and linear arrows as morphisms.*

This category is an enrichment of the usual category **Rel**. The construction can be summarized in the following way:

Lemma 8. Int *is obtained by lifting* **Rel** *through the following specification structure ([9]):*

- *if X is a set, $\mathrm{Pr}_X \equiv \mathcal{P}(X) \to \mathcal{P}(X)$;*
- *if $r \subseteq X \times Y$, $P \in \mathrm{Pr}_X$ and $Q \in \mathrm{Pr}_Y$, then $P\{r\}Q$ iff $\langle r \rangle \cdot P \subseteq Q \cdot \langle r \rangle$.*

Let's now turn our attention to the structure of this category:

Lemma 9. *In* **Int**, $\top$ *is terminal and* $\&$ *is the cartesian product.*

Proof. This is immediate. □

Lemma 10. $\_^\perp$ *is an involutive contravariant functor.*

Proof. Involutivity is trivial; contravariance is only slightly trickier:
r is a seed in $A \multimap B$
$\Leftrightarrow$ { Lemma 6 }
$\forall x\ \langle r\rangle A(x) \subseteq B\langle r\rangle x$
$\Leftrightarrow$
$\forall x\ \overline{B\langle r\rangle x} \subseteq \overline{\langle r\rangle A(x)}$
$\Leftrightarrow$ { lemma: $y \subseteq \overline{\langle r\rangle x}$ iff $\langle r^\sim\rangle y \subseteq \overline{x}$ }
$\forall x\ \langle r^\sim\rangle\overline{B\langle r\rangle x} \subseteq \overline{A(x)}$
$\Rightarrow$ { in particular, for x of the form $\overline{\langle r^\sim\rangle x}$; we have $\overline{x} \subseteq \langle r\rangle\overline{\langle r^\sim\rangle x}$ (lemma) }
$\forall x\ \langle r^\sim\rangle B^\perp(x) \subseteq A^\perp\big(\langle r^\sim\rangle x\big)$
i.e. $r^\sim$ is a seed in $B^\perp \multimap A^\perp$. The action of $\_^\perp$ on morphisms is just $\_^\sim$. □

Corollary 1. Int *is autodual through* $\_^\perp$*;* **0** *is initial; and* $\oplus$ *is the coproduct.*

It is now easy to see that linear arrows transform seeds into seeds, and, in the other direction, antiseeds into antiseeds:

Proposition 2. *Suppose* r *is a linear arrow from* X *to* Y*:*

(i) $\langle r\rangle$ *is a sup-lattice morphism from* $\mathcal{S}(X)$ *to* $\mathcal{S}(Y)$*;*
(ii) $\langle r^\sim\rangle$ *is a sup-lattice morphism from* $\mathcal{S}(Y^\perp)$ *to* $\mathcal{S}(X^\perp)$*.*

Proof. Let $r \in \mathcal{S}(X \multimap Y)$ and $x \subseteq X(x)$; we want to show that $\langle r\rangle x \subseteq Y(\langle r\rangle x)$.
Let $b \in \langle r\rangle x$
$\Leftrightarrow$
$(\exists a)\ (a,b) \in r \wedge a \in x$
$\Rightarrow$ { r is a seed in $X \multimap Y$ }
$(\exists a)\ (a,b) \in X \multimap Y(r) \wedge a \in x$
$\Rightarrow$ { definition of $X \multimap Y$ with the fact that $\langle r\rangle x \subseteq \langle r\rangle x$ }
$b \in Y(\langle r\rangle x)$.
Showing that $\langle r\rangle$ commutes with sups is immediate: it commutes with arbitrary unions, even when the argument is not a seed.
The second point follows because $r^\sim \in \mathcal{S}(Y^\perp \multimap X^\perp)$. □

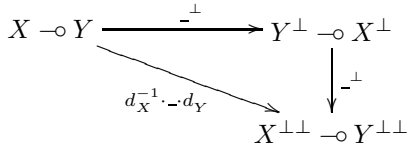
Lemma 11. $\otimes$ *[*$\parr$*] is a categorical tensor product with neutral element* **1** *[*$\perp$*].*

Proof. We need to show the bifunctoriality of $\otimes$. This was actually proved in the previous section (Proposition 1, point *(5)*). The bifunctoriality of $\parr$ follows by duality; and the rest is immediate. □

As a summary of this whole section, we have:

Proposition 3. Int *is a* $*$*-autonomous category. (In particular,* **Int** *is symmetric monoidal closed.)*

Proof. This amounts to checking trivial equalities, in particular, that the following diagram commutes: (where d is the natural isomorphism $X \simeq X^{\perp\perp}$)



It is immediate because $d = \mathbf{Id}$ and $\_^\perp = \_^\sim$. □

6 Exponentials

The category **Int** is thus a denotational model for multiplicative additive linear logic. Let's now add the exponentials $!X$ and $?X$.

Unsurprisingly, we will use finite multisets; here are the necessary definitions and notations:

Definition 12. *Let S be a set;*

- *if $(s_i)_{i\in I}$ and $(t_j)_{j\in J}$ are finite families on S, say $(s_i) \simeq (t_j)$ iff there is a bijection σ from I to J such that $s_i = t_{\sigma(i)}$ for all i in I.*
- *A finite multiset over S is an equivalence class of $\simeq$. We write $[s_i]$ for the equivalence class containing (s_i).*
- *$\mathcal{M}_f(S)$ is the collection of finite multisets over S.*
- *Concatenation of finite families[7] can be lifted to multisets; it is written $+$.*
- *If x and y are two subsets of S, we write $x*y$ for the set $\{[a,b] \mid a \in x \wedge b \in y\}$. Its indexed version is written $\prod_{i\in I} x_i$; it is a kind of commutative product.*
- *If U and V are two subsets of $\mathcal{M}_f(A)$, the set $\{u+v \mid u \in U \wedge v \in V\}$ is written $U * V$ (same symbol, but no confusion arises).*

Definition 13. *For $X = (|X|, P)$, define $!X = (\mathcal{M}_f(|X|), !P)$ where*

$$[a_1, \dots a_n] \in !P(U) \quad \Leftrightarrow \quad \big(\exists (x_i)_{1\le i\le n}\big) \prod_i x_i \subseteq U \wedge (\forall i = 1, \dots n)\, a_i \in P(x_i)$$

Let $?X = \big(!(X^\perp)\big)^\perp$.

Recall that a multiset $[a_i]$ is in $\prod x_i$ iff there is a bijection σ s.t. $\forall i, a_i \in x_{\sigma(i)}$.

A useful intuition is that $[a_1, \dots] \in !P(U)$ iff $[a_1, \dots]$ is in a "weak infinite tensor" $\bigoplus_n X^{\otimes n}(U)$. In terms of specifications and programs, it suggests multithreading: for an initial state $[a_1, \dots a_n]$, start n occurrences of the program in the states $a_1, \dots a_n$; the final state is nothing but the multiset of all the n final states.[8] The "weak" part means that we forget the link between a particular final state and a particular initial state.

Note that this is a "non-uniform" model in the sense that the web of $!X$ contains all finite multisets, not just those whose underlying set is a seed. It is thus closer to non-uniform (hyper)coherence semantics (see [10] or [11]) than to the traditional (hyper)coherence semantics.

[7] defined on the disjoint sum of the different index sets

[8] The interpretation of !, like that of $\otimes$ is a synchronous operation.

Let's prove a simple lemma about the exponentials:

Lemma 12. *Suppose $U \subseteq \mathcal{M}_f(|A|)$:*

(i) $[a] \in !A(U)$ *iff there is some* x *"included" in* U *(i.e.* $\forall a \in x\ [a] \in U$*) s.t.* $a \in A(x)$*;*
(ii) $l + l' \in !A(U)$ *iff there are* $V * V' \subseteq U$ *s.t.* $l \in !A(V)$ *and* $l' \in !A(V')$*;*
(iii) $[a] \in ?A(U)$ *iff for all* $\overline{x}$ *"included" in* $\overline{U}$*,* $a \in A(x)$*;*
(iv) $l + l' \in ?A(U)$ *iff for all* $\overline{V} * \overline{V'} \subseteq \overline{U}$*,* $l \in ?A(V)$ *or* $l' \in ?A(V')$*.*

Proof. The first point is immediate and the second is left as an exercise. The third and last point are consequences of the definition of ? in terms of !. □

Define now the interpretation of proofs with exponentials:

(10) if π is $\dfrac{\pi_1 \vdash \Gamma, A}{\vdash \Gamma, ?A}$ then $\pi^* = \big\{\big(\gamma, [a]\big) \mid (\gamma, a) \in \pi_1^*\big\}$;

(11) if π is $\dfrac{\pi_1 \vdash \Gamma}{\vdash \Gamma, ?A}$ then $\pi^* = \big\{\big(\gamma, [\,]\big) \mid \gamma \in \pi_1^*\big\}$;

(12) if π is $\dfrac{\pi_1 \vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}$ then $\pi^* = \big\{(\gamma, l + l') \mid (\gamma, l, l') \in \pi_1^*\big\}$;

(13) if π is $\dfrac{\pi_1 \vdash ?\Gamma, A}{\vdash ?\Gamma, !A}$

then we define $\big(\gamma_1, \dots \gamma_l, [a_1 \dots a_n]\big) \in \pi^*$ if for each $j = 1, \dots l$, there is a partition $\gamma_j = \sum_{1 \le i \le n} \gamma_j^i$ and the following holds: for each $i = 1, \dots n$, $(\gamma_1^i, \dots \gamma_l^i, a_i) \in \pi_1^*$.

Proposition 4. *If π a proof of $\vdash \Gamma$, then π^* is a seed of Γ.*

Proof. Points *(10)* and *(11)* are immediate.
(12): suppose π_1 is a seed $\Gamma, ?A, ?A$ and let $(\gamma, l + l')$ be an element of π.
By contradiction, suppose that $(\gamma, l + l') \notin \Gamma, ?A(\pi)$
$\Leftrightarrow$
$(\gamma, l + l') \in \Gamma^\perp \otimes !A^\perp(\overline{\pi})$
$\Leftrightarrow$ { for some $u \times U \subseteq \overline{\pi}$ }
$\gamma \in \Gamma^\perp(u) \wedge l + l' \in !A^\perp(U)$
$\Leftrightarrow$ { Lemma 12 }
$\gamma \in \Gamma^\perp(u) \wedge (\exists V * V' \subseteq U)\ l \in !A^\perp(V) \wedge l' \in !A^\perp(V')$
$\Rightarrow$ { lemma: $u \times V \times V' \subseteq \overline{\pi_1}$ }
$\gamma \in \Gamma^\perp(u) \wedge l \in !A^\perp(V) \wedge l' \in !A^\perp(V')$
$\Rightarrow$
$(\gamma, l, l') \in \Gamma^\perp \otimes !A^\perp \otimes !A^\perp(\overline{\pi_1})$
$\Leftrightarrow$
$(\gamma, l, l') \notin \Gamma, ?A, ?A(\pi_1)$, which contradicts the fact that π_1 is a seed in $\Gamma, ?A, ?A$.

(13): suppose that Γ contains only one formula B. The general case will follow from a lemma proved below (Lemma 13). Suppose that π_1 is a seed in $?B, A$; let $(l, [a_1, \dots a_n])$ be in π, *i.e.* $(l_i, a_i) \in \pi_1$ for $i = 1, \dots n$, for some partition $(l_1, \dots l_n)$ of l.
Suppose by contradiction that $(l, [a_1 \dots a_n]) \notin ?B, !A(\pi)$
$\Leftrightarrow$
$(l, [a_1, \dots a_n]) \in !B^\perp \otimes ?A^\perp(\overline{\pi})$
$\Leftrightarrow$ { for some $U \times V \subseteq \overline{\pi}$ }
$l \in !B^\perp(U) \wedge [a_1, \dots a_n] \in ?A^\perp(V)$
$\Rightarrow$ { definition of $?A$ }
$l \in !B^\perp(U) \wedge \Big(\big(\forall (x_i)\big) \ \prod x_i \subseteq \overline{V} \Rightarrow (\exists i)\ a_i \in A(\overline{x_i}) \Big)$
$\Leftrightarrow$ { Lemma 12 for l: for some (U_i) s.t. $\prod_i U_i \subseteq U$ }
$(\forall i)\ l_i \in !B^\perp(U_i) \wedge \Big(\big(\forall (x_i)_i\big) \ \prod_i x_i \subseteq \overline{V} \Rightarrow (\exists i)\ \dots$
$\Rightarrow$ { define $x_i = \langle \pi_1 \rangle U_i$; lemma: $\prod_i x_i \subseteq \overline{V}$ }
$\Big((\forall i)\ l_i \in !B^\perp(U_i) \Big) \wedge \Big((\exists i)\ a_i \in A(\overline{x_i}) \Big)$
$\Rightarrow$ { lemma: $U_i \times x_i \subseteq \overline{\pi_1}$ }
$(\exists i)\ (\exists U_i \times x_i \subseteq \overline{\pi_1})\ l_i \in !B^\perp(U_i) \wedge a_i \in A(\overline{x_i})$
$\Leftrightarrow$
$(l_i, a_i) \in !B^\perp \otimes A^\perp(\overline{\pi_1})$
$\Leftrightarrow$
$(l_i, a_i) \notin ?B, A(\pi_1)$, which contradicts the fact that π_1 is a seed in $?B, A$. □

Lemma 13. *For all interfaces X and Y, we have $!(X \,\&\, Y) = !X \otimes !Y$.*

Proof. The state spaces are isomorphic via $\mathcal{M}_f(|X|+|Y|) \simeq \mathcal{M}_f(|X|) \times \mathcal{M}_f(|Y|)$. We will use this transparently, for example $l_X + l_Y \in R$ iff $(l_X, l_Y) \in R$. This is possible because the sets are disjoint: we can always split a multiset in $x * y$ into two multisets in x and y. (In other words: if $x \cap y = \emptyset$ then $x * y \simeq x \times y$.)

Notice also that $(1, a) \in X \,\&\, Y(x, y) \Leftrightarrow a \in X(x)$ so that when considering a particular element of $X + Y(x, y)$, only one part of the argument (x, y) is really important; the other can be dropped (or replaced with $\emptyset$).

$\subseteq$: suppose $[a_1, \dots a_n] + [b_1, \dots b_m] \in !(X \,\&\, Y)(R)$
$\Leftrightarrow$ { for some $(x_i)_{i=1\dots n}$ and $(y_j)_{j=1\dots m}$ }
$\prod_i x_i * \prod_j y_j \subseteq R \wedge (\forall i)\ a_i \in X(x_i) \wedge (\forall j) b_j \in Y(y_j)$
$\Rightarrow$ { define $U' = \prod_i x_i$ and $V' = \prod_j y_j$ }
$(\exists U' \times V' \subseteq R)\ [a_i] \in !X(U') \wedge [b_j] \in !Y(V')$
$\Leftrightarrow$
$([a_1, \dots a_n], [b_1, \dots b_m]) \in !X \otimes !Y(R)$.

$\supseteq$: suppose $([a_1, \dots a_n], [b_1, \dots b_m]) \in !X \otimes !Y(R)$
$\Leftrightarrow$ { for some $U' \times V' \subseteq R$ }
$[a_1, \dots a_n] \in !X(U') \wedge [b_1, \dots b_m] \in !Y(V')$
$\Leftrightarrow$ { for some (x_i) s.t. $\prod x_i \subseteq U'$ and (y_j) s.t. $\prod y_j \subseteq V'$ }
$(\forall i)\ a_i \in X(x_i) \wedge (\forall j)\ b_j \in Y(y_j)$
$\Rightarrow$ { $\prod_i x_i \times \prod_j y_j \subseteq U \times V$ and thus $\prod_i x_i * \prod_j y_j \subseteq R$ }
$[a_1, \dots a_n] + [b_1, \dots b_m] \in !(X \,\&\, Y)(R)$. □

This allows us to transform any sequent $?\Gamma = ?B_1 \parr \ldots ?B_n$ into $?(B_1 \oplus \ldots B_n)$, and thus, formally ends the proof of Proposition 4 point *(13)*.

7 Linear Interfaces and Linear Seeds

What is the structure of those interfaces that come from a linear formula? The answer is unfortunately trivial:

Proposition 5. *If F is a linear formula, then $P_F = \mathbf{Id}_{\mathcal{P}|F|}$.*

Proof. Immediate induction. Let's treat the case of the exponentials:
suppose $F(x) = x$; suppose moreover that $[a_1, \ldots a_n] \in U$
$\Rightarrow$
$a_i \in F(\{a_i\})$ for all i and $\bigcap\{a_i\} = \{[a_1, \ldots a_n]\} \subseteq U$
$\Rightarrow$
$[a_1, \ldots a_n] \in !F(U)$
Similarly, suppose $[a_1, \ldots a_n] \in !F(U)$
$\Rightarrow$
each $a_i \in F(x_i) = x_i$ for some (x_i) s.t. $\bigcap x_i \subseteq U$
$\Rightarrow$ $\{\ [a_1, \ldots a_n] \in \bigcap x_i\ \}$
$[a_1, \ldots a_n] \in U$. □

In particular, every subset of $|F|$ is a clique and an anticlique: the situation is thus quite similar to the purely relational model. In the presence of atoms however, interfaces become much more interesting.

Adding atoms is sound because the proof of Proposition 1 doesn't rely on the particular properties of interfaces. Note that we need to introduce a general axiom rule and its interpretation:

(14) if π is $\dfrac{}{\vdash X, X^\perp}$ then $\pi^* = \mathbf{Id}_{|X|} = \{(a, a) \mid a \in |X|\}$.

This is correct in the sense that π^* is always a clique in $X \parr X^\perp$.

With such atoms, the structure of linear interfaces gets non trivial.[9] For example, let's consider the following atom $X = \big(\{-, +\}, P\big)$ defined by:

- $P(\emptyset) = \emptyset$ and $P(|X|) = |X|$;
- $P(\{+\}) = \{-\}$ and $P(\{-\}) = \{+\}$.

This is the simplest example of an interesting interface, and corresponds to a "switch" specification. (Interpret $-$ as "off" and $+$ as "on".)

Lemma 14. *if P is the above specification:*

(i) $P^\perp = P$;
(ii) $P \cdot P = \mathbf{Id}$;

[9] We can extend this to a model for Π^1 logic, and even to full second order, see [12].

(iii) $\mathcal{S}(X) = \{\emptyset, \{+,-\}\}$;
(iv) $\{(+,-),(-,+)\} \in \mathcal{S}(X \otimes X)$.

Proof. This is just trivial computation... □

Point *(iv)* shows in particular that a seed in $X \otimes Y$ needs not contain a product of seeds in X and Y. (Compare with Lemma 5.)

The hierarchy generated from this single interface is however still relatively simple: call a specification *deterministic* if it commutes with non-empty unions and intersections.

Lemma 15. *Let F be any specification constructed from the above P and the linear connectives. Then F is deterministic. Moreover, F is of the form $\langle f \rangle$ where f is an obvious bijection on the state space of F.*[10]

A less trivial (in the sense that it is not deterministic) specification is the following: if X is a set, $\mathsf{magic}_X(x) = X$. In terms of programming, the use of the magic command allows to reach any predicate, even the empty one!

Lemma 16. $\mathbf{Id}_{|X|} \subsetneq \mathsf{magic}_X \multimap \mathsf{magic}_X(\mathbf{Id}_{|X|})$ *if* $X \neq \emptyset$.

Thus we cannot strengthen the definition of seeds to read "$x = P(x)$" without imposing further constraints on our specifications. It is still an open question to find a nice class of predicate transformers for which it would be possible. (However, considerations about second order seem to indicate that strengthening the definition of seeds in such a way is not a good idea.)

In the case with atoms, because the structure of seeds (sup-lattice) is quite different from the structure of cliques in the ...-coherent model (domain), it is difficult to relate seeds and cliques. In particular, a seed needs not be a clique (since the union of arbitrary cliques is not necessarily a clique); and a clique needs not be a seed (since a subset of a seed is not necessarily a seed).

Conclusion

One aspect which was not really mentioned here is the fact that linear arrows from A to B are equivalent to the notion of *forward data refinement* (Lemma 6) from the refinement calculus. In particular, a linear proof of $A \multimap B$ is a proof that specification B *implements* specification A. It would interesting to see if any application to the refinement calculus could be derived from this work. In the same direction, trying to make sense of the notions of *backward data refinement*, or of *general data refinement* in terms of linear logic could prove interesting.[11]

[10] where $\langle f \rangle(x) = \{f(a) \mid a \in x\}$

[11] A data refinement from specification F to specification G is a predicate transformer P s.t. $P \cdot F \subseteq G \cdot P$; a forward [resp. backward] data refinement is a data refinement which commutes with arbitrary unions [resp. arbitrary intersections].

The fact that this model is degenerate in the propositional case is disappointing, but degeneracy disappear when we consider Π^1 logic, and *a fortiori* when we consider full second-order (see [12]). The point of extending this propositional model to Π^1 is to remove the dependency on specific valuations for the atoms present in a formula.

One the interesting consequences of this work is that a a proof of a formula F gives a guarantee that the system specified by the formula F can avoid deadlocks seems to point toward other fields like process calculi and similar models for "real" computations. This direction is currently being pursued together with the following link with the differential lambda-calculus ([2]): one property of this model which doesn't reflect any logical property is the following; we have a natural transformation $A \multimap !A$ called *co-dereliction*, which has a natural interpretation in terms of differential operators on formulas (see [3]). Note that such a natural transformation forbids any kind of completeness theorem, at least as far as "pure" linear logic is concerned.

References

1. Girard, J.Y.: Linear logic. Theoretical Computer Science **50** (1987)
2. Ehrhard, T., Regnier, L.: The differential lambda calculus. Theoretical Computer Science **309** (2003) 1–41
3. Ehrhard, T., Regnier, L.: Differential interaction nets. unpublished note (2004)
4. Gardiner, P.H.B., Martin, C.E., de Moor, O.: An algebraic construction of predicate transformers. Science of Computer Programming **22** (1994) 21–44
5. Back, R.J., von Wright, J.: Refinement Calculus: a systematic introduction. Graduate texts in computer science. Springer-Verlag, New York (1998)
6. Ehrhard, T.: Hypercoherences: a strongly stable model of linear logic. Mathematical Structures in Computer Science **3** (1993) 365–385
7. Ehrhard, T.: Finiteness spaces. to appear in Mathematical Structures in Computer Science (2004)
8. Back, R.J., von Wright, J.: Product in the refinement calculus. Technical Report 235, Turku Center for Computer Science (1999)
9. Abramsky, S., Gay, S.J., Nagarajan, R.: A specification structure for deadlock-freedom of synchronous processes. Theoretical Computer Science **222** (1999) 1–53
10. Bucciarelli, A., Ehrhard, T.: On phase semantics and denotational semantics: the exponentials. Annals of Pure and Applied Logic **109** (2001) 205–241
11. Boudes, P.: Non-uniform hypercoherences. In Blute, R., Selinger, P., eds.: Electronic Notes in Theoretical Computer Science. Volume 69., Elsevier (2003)
12. Hyvernat, P.: Predicate transformers and linear logic: second order. unpublished note (2004)