A Relational Account of Call-by-Value Sequentiality

Jon G. Riecke
Anders B. Sandholm

BRICS Report Series RS-99-10

ISSN 0909-0878 March 1999

A Relational Account of Call-by-Value Sequentiality[*]

Jon G. Riecke
Bell Laboratories
Lucent Technologies
700 Mountain Avenue
Murray Hill, NJ 07974, USA

Anders Sandholm
BRICS[†] Department of Computer Science
University of Aarhus
Ny Munkegade, Bldg. 540
DK-8000 Aarhus C, Denmark

March, 1999

Abstract

We construct a model for FPC, a purely functional, sequential, call-by-value language. The model is built from partial continuous functions, in the style of Plotkin, further constrained to be uniform with respect to a class of logical relations. We prove that the model is fully abstract.

1 Introduction

The problem of finding an abstract description of sequential functional computation has been one of the most enduring problems of semantics. The problem dates from a seminal paper of Plotkin (1977), who pointed out that certain elements in Scott models are not definable. In Plotkin's example, the function

$$por(x, y) = \begin{cases} \mathit{true} & \text{if } x \text{ or } y = \mathit{true} \\ \mathit{false} & \text{if } x \text{ and } y = \mathit{false} \\ \bot & \text{otherwise} \end{cases}$$

where $\bot$ denotes divergence, cannot be programmed in the language PCF, a purely functional, sequential, call-by-name language with booleans and numbers as base types. The problem is called the "sequentiality problem" because,

[*]A preliminary version of this paper appeared in *Proceedings, Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 258–267. IEEE Computer Society Press, 1997.

[†]Basic Research in Computer Science, Centre of the Danish National Research Foundation.

intuitively, the only way to program *por* involves evaluating boolean expressions in parallel. Of course, there are other elements in the Scott model of PCF that cannot be programmed: the domains have uncountably many elements. Nevertheless, the *por* function causes problems for reasoning about programs: it causes two terms in the language PCF to be distinct denotationally even though the terms cannot be distinguished by any program.

The problem of modelling sequentiality is enduring because it is robust. For instance, changing the reduction strategy of PCF from call-by-name to call-by-value makes no difference: versions of the *por* function reappear in the standard Scott model (albeit at higher type). Even for languages that lack an explicit base type, e.g., the polymorphic λ-calculus with recursion, the known Scott models contain parallel elements that cause a failure of full abstraction.

When examples like *por* do not exist, the denotational model is said to be **fully abstract**. More precisely, full abstraction requires an operational definition of equivalence (interchangeability in all programs) to match operational equivalence. Shortly after Plotkin's paper, Milner proved that there was exactly *one* fully abstract model of PCF meeting certain conditions (Milner, 1977). Until recently, all descriptions of this fully abstract model have used operational semantics (see, for instance, (Mulmuley, 1987; Stoughton, 1988)). New constructions using logical relations (O'Hearn and Riecke, 1995; Sieber, 1992) have yielded a more abstract understanding of Milner's model. Game semantics (Abramsky et al., 1994; Hyland and Ong, 1995; Nickau, 1994) has also been used to give other semantic constructions of fully abstract models of PCF, even though it is still open whether these models are isomorphic to Milner's model.

This paper adapts and extends the logical-relations model for PCF to a call-by-value setting. More precisely, it constructs a model for a call-by-value, purely functional, sequential language called FPC. FPC includes a base type with one convergent value, strict products, strong (categorical) sums, functions, and recursive types. Full abstraction for FPC is interesting for at least two reasons. First, FPC can be regarded as the purely functional, non-polymorphic sublanguage of Standard ML (Milner et al., 1997): recursive types and sums are the basis of `datatype` declarations, and both Standard ML and FPC are call-by-value. By studying FPC, we learn more about programming languages like Standard ML. Second, FPC can serve as an expressive metalanguage for denotational semantics (Gunter, 1992; Plotkin, 1985). FPC, for instance, has enough expressive power to encode a call-by-value version of PCF (the base type of numbers can be encoded via a recursive type). Given a fully abstract translation (Riecke, 1993) from a language into FPC, the model of FPC yields a fully abstract model of the language.

There is another, purely technical reason to be interested in FPC: since it contains only one trivial base type, we can learn more about the structure of Sieber's logical relations by studying FPC. Sieber's model of PCF, and the fully abstract model of PCF using Kripke relations (O'Hearn and Riecke, 1995), begin from a set of n-ary relations (for all n) at a **flat** base type—i.e., where the convergent elements of the domain are unordered. PCF thus builds in a limited form of sum type, available only at the base type. When we decompose that base type into a sum of the primitive base type, we may learn more about the nature of sequential computation.

As a preview to seeing how our relations decompose Sieber's, it is helpful to recall Sieber's definition. Suppose $A \subseteq B \subseteq \{1, \ldots, n\}$, and let $S^n_{A,B}$ be

$$\{(d_1, \ldots, d_n) \mid (\forall i \in A.\, d_i \neq \bot) \implies (\forall i, j \in B.\, d_i = d_j)\}.$$

Then R is an n-ary **sequentiality relation** if R is the intersection of relations of the form $S^n_{A,B}$. Sieber's sequentiality relations have an elegant semantic definition: nothing in the definition refers to the terms or operations of PCF. The relations can be used in simple proofs of the non-definability of elements, since all PCF terms preserve the sequentiality relations. For instance, it is easy to show that *por* does not preserve the sequentiality relation $S^3_{\{1,2\},\{1,2,3\}}$, and hence it must not be definable. The relations also seem to say something about sequential computation: if certain elements of a tuple must converge, then other elements must converge.

On a technical level, Sieber's definition of "sequentiality relation" seems limited to flat base types. It is difficult to see how, for instance, to extend the definition to complex sums such as $\mathbf{int} \oplus (\mathbf{int} \Rightarrow \mathbf{int})$, since it does not make sense to check for equality at functional type. Instead of directly extending Sieber's relations, our relations break down sequentiality relations into two components. The first captures the sequential behavior of *termination*: if certain elements in a tuple in the relation terminate, then certain other elements in the tuple must terminate. The second captures the behavior of *sums*: if certain elements in a tuple lie in one side of a sum type, other components must lie in the same side of the sum type. This is much like Sieber's definition in asking for *equality* of all B-indexed elements of a tuple in $S^n_{A,B}$. The interesting case comes when this second component of relations is lifted to types other than sums: when, for example, the tuple is a tuple of elements in a function type. In essence, the second component of relations encodes a form of "computation tree," stating which subtuples of a tuple form consistent traces of the computation so far.

We begin by introducing the syntax and operational semantics of FPC. We then describe the form of the relations, showing the decomposition into the two portions described above. We follow the O'Hearn-Riecke construction for PCF (O'Hearn and Riecke, 1995) and lift the relations to *Kripke relations of varying arity* (Jung and Tiuryn, 1993). We then define a category, in which the objects have partial-order as well as relational structure, and in which the morphisms preserve the partial-order and relational structure of objects. A model of FPC lives inside this category. The Kripke relations are then used to establish the full abstraction of the model.

2 The Language FPC

FPC is a call-by-value, purely functional language with single base type `unit`, sums, products, functions, and recursive types (Gunter, 1992; Plotkin, 1985).

Our version of FPC has types given by the grammar

$$s, t ::= \texttt{void} \mid \texttt{unit} \mid (s \oplus t) \mid (s \otimes t) \mid (s \Rightarrow t) \mid \alpha \mid (\texttt{rec}\ \alpha.\ t)$$

where α ranges over a collection of **type variables**. This version of FPC has one more base type, namely `void`, than the standard version of FPC. The type `void` stands for the type with no convergent elements, and can be represented by the type $(\texttt{rec}\ \alpha.\ \alpha)$, but having an explicit name makes some of the notation simpler. Types are identified up to renaming of type variables bound by `rec`. We assume that all types appearing in terms and the typing rules are closed unless otherwise noted.

The raw terms of FPC are given by the grammar

$$\begin{array}{rcl} M, N, P & ::= & \Omega \mid x \mid (\lambda x : t.\ M) \mid (M\ N) \mid \\ & & \langle\rangle \mid \langle M, N\rangle \mid (\texttt{proj}_i\ M) \mid (\texttt{inj}_i\ M) \mid \\ & & (\texttt{case}\ M\ \texttt{of}\ \texttt{inj}_1(x).N\ \texttt{or}\ \texttt{inj}_2(x).P) \mid \\ & & (\texttt{intro}_{\texttt{rec}\ \alpha.\ s}\ M) \mid (\texttt{elim}_{\texttt{rec}\ \alpha.\ s}\ M) \end{array}$$

Ω denotes a divergent term, and only appears in our version of FPC in order to make the proofs simpler. A typing judgement is a formula of the form $\Gamma \vdash M : t$ where M is a term, t a type, and Γ is a **typing context**, i.e., a finite function from variables to types. Rules for deriving typing judgements appear in Table 1.

Evaluation rules, written in natural style, for FPC appear in Table 2. In the rules, we use the notation $M[N/x]$ to denote capture-free substitution of N for x in M. Notice that function application in FPC is call-by-value: arguments to functions must be values before they are substituted into bodies of functions. We write $M \Downarrow$ if there is a term V such that $M \Downarrow V$, and $M \Uparrow$ if there is no such V. The operational approximation relation can then be defined as follows:

Definition 2.1 $M \sqsubseteq_{FPC} N$ if for any context $C[\cdot]$ such that $C[M]$ and $C[N]$ are closed, well-typed terms, $C[M] \Downarrow$ implies $C[N] \Downarrow$.

FPC is a sparse language, but it still has enough computing power for many applications. For instance, Plotkin (1985) and Gunter (1992) show how to build recursion operators using recursive types. One can encode a sequencing operation in FPC: $(M; N)$ stands for the term $((\lambda x : s.\ N)\ M)$, where x does not occur free in N. Indeed, the semantics of many programming languages—including non-functional languages—can be given by translation to FPC. FPC's main deficiency as a metalanguage for denotational semantics is a lack of parametric polymorphism (as in the Girard-Reynolds calculus (Girard, 1971; Reynolds, 1974)), which precludes a good representation of abstract data types.

3 Category of Meanings

This section defines a category suitable for interpreting FPC, and gives various constructions for defining the meaning of types and terms. Objects in the

Table 1: Type Rules for FPC.

$$\Gamma, x : t \vdash x : t$$

$$\Gamma \vdash \Omega : s$$

$$\Gamma \vdash \langle\rangle : \texttt{unit}$$

$$\frac{\Gamma, x : s \vdash M : t}{\Gamma \vdash (\lambda x : s.\, M) : (s \Rightarrow t)}$$

$$\frac{\Gamma \vdash M : (s \Rightarrow t) \qquad \Gamma \vdash N : s}{\Gamma \vdash (M\ N) : t}$$

$$\frac{\Gamma \vdash M : s \qquad \Gamma \vdash N : t}{\Gamma \vdash \langle M, N\rangle : (s \otimes t)}$$

$$\frac{\Gamma \vdash M : (s_1 \otimes s_2)}{\Gamma \vdash (\texttt{proj}_i\ M) : s_i}$$

$$\frac{\Gamma \vdash M : s_i}{\Gamma \vdash (\texttt{inj}_i\ M) : (s_1 \oplus s_2)}$$

$$\frac{\Gamma \vdash M : (s_1 \oplus s_2) \qquad \Gamma, x : s_i \vdash N_i : t}{\Gamma \vdash (\texttt{case}\ M\ \texttt{of}\ \texttt{inj}_1(x).N_1\ \texttt{or}\ \texttt{inj}_2(x).N_2) : t}$$

$$\frac{\Gamma \vdash M : s[\texttt{rec}\ \alpha.\ s/\alpha]}{\Gamma \vdash (\texttt{intro}_{\texttt{rec}\ \alpha.\ s}\ M) : (\texttt{rec}\ \alpha.\ s)}$$

$$\frac{\Gamma \vdash M : (\texttt{rec}\ \alpha.\ s)}{\Gamma \vdash (\texttt{elim}_{\texttt{rec}\ \alpha.\ s}\ M) : s[\texttt{rec}\ \alpha.\ s/\alpha]}$$

Table 2: Evaluation Rules for FPC.

$$\langle\rangle \Downarrow \langle\rangle$$

$$(\lambda x : s.\, M) \Downarrow (\lambda x : s.\, M)$$

$$\frac{M \Downarrow (\lambda x : s.\, M') \qquad N \Downarrow V' \qquad M'[V'/x] \Downarrow V}{(M\ N) \Downarrow V}$$

$$\frac{M_1 \Downarrow V_1 \qquad M_2 \Downarrow V_2}{\langle M_1, M_2\rangle \Downarrow \langle V_1, V_2\rangle}$$

$$\frac{M \Downarrow \langle V_1, V_2\rangle}{(\texttt{proj}_i\ M) \Downarrow V_i}$$

$$\frac{M \Downarrow V}{(\texttt{inj}_i\ M) \Downarrow (\texttt{inj}_i\ V)}$$

$$\frac{M \Downarrow (\texttt{inj}_i\ V) \qquad N_i[V/x] \Downarrow R}{(\texttt{case}\ M\ \texttt{of}\ \texttt{inj}_1(x).N_1\ \texttt{or}\ \texttt{inj}_2(x).N_2) \Downarrow R}$$

$$\frac{M \Downarrow V}{(\texttt{intro}_{\texttt{rec}\ \alpha.\, s}\ M) \Downarrow (\texttt{intro}_{\texttt{rec}\ \alpha.\, s}\ V)}$$

$$\frac{M \Downarrow (\texttt{intro}_{\texttt{rec}\ \alpha.\, s}\ V)}{(\texttt{elim}_{\texttt{rec}\ \alpha.\, s}\ M) \Downarrow V}$$

category will have both partial-order structure and relational structure. More precisely, the objects are composed of dcpo's, i.e., **directed-complete posets** not necessarily possessing a least element, and relations on those dcpo's. The morphisms of the category preserve the dcpo structure (i.e., are partial, continuous functions), and preserve the relational structure (a property we call **uniformity**). The formal definitions of dcpo and continuity may be found elsewhere (Gunter and Scott, 1990; Plotkin, 1985).

The construction consists of five main parts:

1. Definition of the relational structures (Sections 3.2-3.3).
2. Definition of the category, and the verification that it is a dcpo-enriched category (Section 3.4).
3. Definitions of meanings for `void`, `unit` as objects in the category, and of tensor product $\otimes$, coproduct $\oplus$, and function space $\Rightarrow$ as functors, and verification that they satisfy certain conditions (Sections 3.5-3.6).
4. Verification that the category is "partial cartesian closed" (defined below), and hence a model of FPC without recursive types (Section 3.7).
5. Construction of colimits, and verification that the colimits produce objects in the category (Section 3.8).

Most of the proofs are straightforward and follow well-established patterns from category theory.

3.1 Preliminaries

Given two sets X, Y, the set $(X \oplus Y)$ is the disjoint union, $(X \times Y)$ is the cartesian product, $[X \to_t Y]$ is the set of total functions from X to Y, and $[X \to_p Y]$ is the set of partial functions from X to Y. The identity function on a set X is denoted id_X, and $(h; f)$ denotes the diagrammatic composition of two functions h and f. The functions $inj_i : X_i \to (X_1 \oplus X_2)$ are the injection functions into the disjoint sum, and $proj_i : (X_1 \times X_2) \to X_i$ are the projection functions from the cartesian product. For products, we abuse notation and use $\langle x, y\rangle$ for elements of $X \times Y$ and $\langle f, g\rangle$ for functions $W \to (X \times Y)$ when $f : W \to X$ and $g : W \to Y$.

Partial functions require some special notation. We write $f(d)\downarrow$ when a partial function f is defined on argument d, and $f(d)\uparrow$ when $f(d)$ is undefined. Kleene equality takes definedness between mathematical expressions into account: we write $exp_1 \simeq exp_2$ if, whenever one side is defined, both sides are defined and equal. Similarly, we extend an ordering relation $\sqsubseteq$ to $\precsim$ as follows: $exp_1 \precsim exp_2$ if, when exp_1 is defined and equal to e_1, then exp_2 is defined and equal to some e_2 and $e_1 \sqsubseteq e_2$.

Following Jung and Tiuryn, we often represent the elements of relations as *finite, total functions* from indices to values instead of tuples of values; this simplifies some of the definitions and proofs (Jung and Tiuryn, 1993; O'Hearn and Riecke, 1995).

3.2 Termination and path theories

There are two kinds of relations in the model, **termination relations** and **computational relations**.

The first kind of relation uses subsets of indices generated from simple implicational theories. If w is a finite set of indices, then a **w-termination theory** is a set of implications of the form $(w' \vdash d)$ where $d \in w$ and w' is a subset of w. Intuitively, each implication states a property similar to the Sieber's sequentiality relations: if a function halts on the indices in the premise, it must halt on the index in the conclusion. Indeed, the termination part of Sieber's relations $S^n_{A,B}$ can be encoded using implications:

$$\text{if } A \subseteq B \subseteq w = \{1, \ldots , n\} \text{ and } B \setminus A = \{d_1, \ldots , d_k\},$$
$$\text{then } S^w_{A,B} \text{ corresponds to the implications } (A \vdash d_1), \ldots , (A \vdash d_k).$$

The subsets of w that validate the implications in the theory are the building blocks of termination relations. Suppose $w' \subseteq w$; then we say that w' **validates** $(w'' \vdash d)$, written $w' \models (w'' \vdash d)$, if $w'' \subseteq w'$ implies $d \in w'$. If T is a w-termination theory, we say that $w' \subseteq w$ is a **T-model** if $w' \models \psi$ for all $\psi \in T$.

There is an alternative characterization of T-models that can be helpful in proving facts about termination relations. Suppose w is a finite set and $X \subseteq \mathcal{P}(w)$. Then X is a **closure system** if $w \in X$ and X is closed under intersection. A closure system determines a w-termination theory:

Proposition 3.1 *Suppose w is a finite set. Then $X \subseteq \mathcal{P}(w)$ is a closure system iff there is a w-termination theory T such that X is the set of T-models.*

Proof: $(\Leftarrow)$ Suppose T is a w-theory. Then obviously w is a T-model. To see closure under intersection, suppose w_1, w_2 are T-models. Suppose $(w' \vdash d) \in T$, and $w' \subseteq w_1 \cap w_2$. Then $d \in w_1$ and $d \in w_2$, hence $d \in w_1 \cap w_2$. Thus, $w_1 \cap w_2$ is also a T-model.

$(\Rightarrow)$ Suppose X is a closure system. Define the theory T by

$$(w'' \vdash d) \in T \text{ iff for all } w' \in X,\ w' \models (w'' \vdash d).$$

Let $Y = \{w' \mid w' \text{ is a } T\text{-model}\}$; we want to show that $X = Y$. It is obvious that $X \subseteq Y$: if $w' \in X$, then by construction it validates all the formulas in T and hence is a T-model. Conversely, suppose $w' \notin X$. Let

$$w_0 = \bigcap \{w_1 \in X \mid w' \subseteq w_1\}.$$

Since X is closed under finite intersections, and the set above is finite because w is, w_0 must be in X. But since $w' \notin X$ and $w' \subseteq w_0$, it must be the case that there is a $d \in w_0$ such that $d \notin w'$. Now consider the formula $(w' \vdash d)$. Note that $(w' \vdash d) \in T$, and $w' \not\models (w' \vdash d)$. Thus, $w' \notin Y$, completing the proof. ■

(See (Wechler, 1992), page 22 for the same proof.) The proof is similar to a part of Sieber's proof that the sequentiality relations are precisely those that are closed under the operations of PCF (Sieber, 1992).

The second kind of relation is built from sets of sets of T-models. If T is a w-termination theory, define a **path set** to be a set $\{w_1, \ldots, w_n\}$ of nonempty, disjoint T-models. (The reason for the name "path set" will become clear shortly.) By convention, when we write a path set we assume there are no duplicates. For example, when we write the path set $\{w_1, \ldots, w_n\}$, each w_i differs from the others. A **T-path theory** S is a set of path sets that satisfies the following conditions:

- $\{\} \in S$ (using the alternative notation for the empty set);
- If w' is a nonempty T-model, then $\{w'\} \in S$;
- If $X = \{w_1, \ldots, w_k\}$ and $X' = \{w'_1, \ldots, w'_l\}$ are in S, and $w_{i,j} = (w_i \cap w'_j)$, then $X \sqcap X' = \{w_{i,j} \mid w_{i,j} \text{ is nonempty}\} \in S$; and
- If $\{w_1, \ldots, w_k\}$ and $\{w'_1, \ldots, w'_l\}$ are in S, and for some j and all i, $w'_i \subseteq w_j$, then

$$\{w_1, \ldots, w_{j-1}, w'_1, \ldots, w'_l, w_{j+1}, \ldots, w_k\} \in S.$$

This definition appears to be related to the notion of a Grothendieck topology (Fiore and Simpson, 1999).

3.3 Relations

Termination theories are the building blocks of the first kind of relations, the termination relations. Let T be a w-termination theory, and D be a dcpo. A **T-termination relation on D** is a set of the form

$$R \subseteq \bigcup_{w' \text{ is a } T\text{-model}} [w' \rightarrow_t D],$$

such that the following properties hold:

1. **Directed completeness:** R is directed complete, where $f \sqsubseteq g$ iff f and g have the same domain w', and for all $d \in w'$, $f(d) \sqsubseteq g(d)$ in D.
2. **Downward closure:** For any $f \in R$ with domain w' and T-model $w'' \subseteq w'$, $(\underline{\lambda} d \in w''.\, f(d)) \in R$.

Here, $(\underline{\lambda} d \in w''.\, f(d))$ stands for the function with domain w'', whose return value is $f(d)$; thus, an element of a termination relation R is a function from a subset of indices to elements of the dcpo. To get some intuition, it is again helpful to think of indices as possible arguments to a function, and the elements as the return values of the function. An element of R then represents a "related" set of values returned by a function.

There is a subtle point in the definition of the $\sqsubseteq$ relation: only elements of R that have the *same domain* can be related. One might imagine a different definition, with "tuples" in the relations being partial functions whose domains are T-models. Under this alternative definition, two tuples might be related by $\sqsubseteq$ even if they had different domains. The definition of $\otimes$ on relations does not produce a directed complete relation, however, using this alternative definition.

The second form of relations, computational relations, is built from path theories. These relations lift a termination relation to a partial function on indices. Suppose S is a T-path theory and R is a T-termination relation on D. The **computational relation** R_S is defined by

$$\begin{aligned} R_S = \{\, f \in [w \to_p D] \mid {} & \text{there exists } \{w_1, \ldots, w_k\} \in S \text{ such that } f(d)\!\downarrow \text{ iff} \\ & d \in w_i \text{ for some } i, \text{ and for all } i,\ (\underline{\lambda} d \in w_i.\, f(d)) \in R \ \} \end{aligned}$$

In this case, the order relation on elements of R_S is defined in the usual way:

$$f \sqsubseteq g \text{ if whenever } f(d)\!\downarrow, \text{ then } g(d)\!\downarrow \text{ and } f(d) \sqsubseteq g(d).$$

The computational relations are reminiscent of Moggi's analysis of call-by-value via monads (Moggi, 1991), and they will play a similar role in the semantics below.

These definitions have a common intuition of "tests" that one can apply to a function, with the tests designed so that only sequential functions will pass all tests. In this case, a test consists of a number of simultaneous applications of the function to arguments, where the arguments must also pass the test. Termination theories are used to test the termination behavior of functions; path theories are used to test the consistency of the branching structure of functions.

For example, let `bool` denote the FPC type (`unit` $\oplus$ `unit`) and `true`,`false` denote the terms ($\mathtt{inj}_1\langle\rangle$), ($\mathtt{inj}_2\langle\rangle$), and consider the term

$$f = (\lambda x : \mathtt{bool}.\ \mathtt{if}\ x\ \mathtt{then}\ \mathtt{false}\ \mathtt{else}\ \mathtt{true}).$$

Suppose we pick the termination theory $\{1, 2 \vdash 3\}$. For the function f to pass a test determined by this termination theory, we must apply it to three arguments themselves satisfy the test. Such a test of three arguments might be, for instance, the tuple `true`, `true`, `true`. When the function is applied to three arguments, the result is `false`, `false`, `false`. All three answers terminate, so this test of the function succeeds. The test `true`, `true`, Ω does not yield a successful test: the function returns `false`, `false`, and diverges respectively. Fortunately, however, the arguments themselves do not pass the test themselves, so the function need not satisfy this test.

Tests of the branching behavior of functions using the path theories are similar. A computation branches based on its inputs and returns some final results at the end. Each set in $\{w_1, \ldots, w_n\}$ represents a path in that computational tree; the answers returned at the end of a path must be consistent, hence the restriction of the function to $(\underline{\lambda} d \in w_i.\, f(d))$ must be in R. For example, suppose the termination theory is as above, and that the path theory contains the set $\{\{1\}, \{2\}, \{3\}\}$. Then the function need not return consistent results given three arguments, but the function must halt on all three arguments.

We can now see from where the four conditions on path theories come. The first condition says that the empty set is a valid test; the second says that a potential path set that does no branching is a valid path set; the third says path sets can be combined; the fourth says that an element of a path set may be replaced by finer-grain path set, which amounts to adding a set of branches to a non-branching part of the computation.

3.4 Definition of the category

One could build a category from termination and path theories, but the result would probably not contain a fully abstract model of FPC (see the discussion in Section 7). Instead, we extend the relations to Kripke relations of varying arity (Jung and Tiuryn, 1993). Kripke relations of this kind begin from an **index category** whose objects are finite sets and whose morphisms are total functions (not necessarily all of them). Suppose $\mathcal{C}$ is an index category. A **$\mathcal{C}$-termination theory** T is an $\mathrm{Ob}(\mathcal{C})$-indexed family of w-termination theories T^w such that, for any $\varphi : v \xrightarrow{\mathcal{C}} w$, if w' is a T^w-model, then $\varphi^{-1}(w')$ is a T^v-model, where $\varphi^{-1}(w') = \{d \in v \mid \varphi(d) \in w'\}$. A Kripke relation is a set of termination relations that must fit together. Let $\mathcal{C}$ be an index category, T be a $\mathcal{C}$-termination theory, and D be a dcpo. A family R is a **$\mathcal{C}, T$-termination relation on** D if R is an $\mathrm{Ob}(\mathcal{C})$-indexed family of T^w-termination relations R^w on D satisfying the

> **Kripke monotonicity condition:** For any $f \in R^w$ with domain w' and $\varphi : v \xrightarrow{\mathcal{C}} w$, then $(\underline{\lambda} d \in v'.\ f(\varphi(d))) \in R^v$, where $v' = \varphi^{-1}(w')$.

Path theories also extend straightforwardly to the Kripke case. Let $\mathcal{C}$ be an index category and T be a $\mathcal{C}$-termination theory. Then S is a **$\mathcal{C}, T$-path theory** if S is a family, indexed by objects w of $\mathcal{C}$, of T^w-path theories. If S is a $\mathcal{C}, T$-path theory and R is a $\mathcal{C}, T$-termination relation on D, we let R^w_S denote the computational relation built from R^w and S^w.

We now have enough machinery to build the category $\mathcal{RCPO}$ (for dcpo's with relational structure).

- Objects. An object A consists of a dcpo $|A|$ and a $\mathcal{C}, T$-termination relation $A(T, S)$ on $|A|$ for each index category $\mathcal{C}$, $\mathcal{C}$-termination theory T, and $\mathcal{C}, T$-path theory S. Objects must also satisfy the

  > **Concreteness Condition**: For any $a \in |A|$, $(\underline{\lambda} d \in w.\ a) \in A(T, S)^w$.

- Morphisms. A morphism $f : A \to B$ is a partial continuous function $f : |A| \to_p |B|$ satisfying the

  > **Uniformity Condition**: For all $\mathcal{C}$, $\mathcal{C}$-termination theories T, $\mathcal{C}, T$-path theories S, and $h \in A(T, S)^w$, $(h; f) \in B(T, S)^w_S$.

The definitions of composition and identities are the same as on partial continuous functions. It is straightforward to check that $\mathcal{RCPO}$ is indeed a category; the only non-obvious step is checking that composition produces uniform functions, which follows from

Lemma 3.2 *If $f \in A(T,S)^w_S$ and $g : A \to B$, then $(f;g) \in B(T,S)^w_S$.*

Proof: By definition, there exists a path set $\{w_1, \dots, w_n\}$ such that $f(d)\downarrow$ iff $d \in w_i$ for some i and $(\underline{\lambda} d \in w_i.\, f(d)) \in A(T,S)^w$ for all i. By the uniformity of g, for all i, $h_i = ((\underline{\lambda} d \in w_i.\, f(d)); g) \in B(T,S)^w_S$. Thus, for all i, there exists a path set $\{w_{i,1}, \dots, w_{i,k_i}\}$ such that

- $h_i(d)\downarrow$ iff $d \in w_{i,j}$ for some j; and
- $(\underline{\lambda} d \in w_{i,j}.\, h_i(d)) \in B(T,S)^w$ for all j.

Note that each $w_{i,j} \subseteq w_i$, since the domain of h_i is a subset of w_i. Thus, by the properties of path sets, $\{w_{1,1}, \dots, w_{n,k_n}\}$ is a path set. Note also that $g(f(d))\downarrow$ iff $d \in w_{i,j}$ for some i, j, and $(\underline{\lambda} d \in w_{i,j}.\, g(f(d))) \in B(T,S)^w$ for all i, j. Thus, $(f;g) \in B(T,S)^w_S$. ■

Moreover, the category is dcpo-enriched, i.e., under the usual pointwise ordering of morphisms with $f \sqsubseteq g$ iff for any $a \in |A|$, $f(a) \precsim g(a)$, the set of morphisms from A to B $Hom_{\mathcal{RCPO}}(A,B)$ is a dcpo. The proof of this fact, stated precisely as Proposition 3.5 below, requires two lemmas:

Lemma 3.3 *Suppose $f \in A(T,S)^w_S$ with path set $W = \{w_1, \dots, w_k\}$. Suppse $W' = \{w'_1, \dots, w'_l\} \in S$, and $\bigcup W = \bigcup W'$. Then $W \sqcap W' = W'' = \{w''_1, \dots, w''_m\}$ is also a path set for f. That is, $d \in w''_i$ iff $f(d)\downarrow$ and for all i, $(\underline{\lambda} d \in w''_i.\, f(d)) \in A(T,S)^w$.*

Proof: It is easy to see that $\bigcup W'' = \bigcup W$, and hence it follows that $d \in w''_i \in W''$ iff $f(d)\downarrow$. ■

Lemma 3.4 *If $A(T,S)$ is a $\mathcal{C}, T$-relation on A, then $A(T,S)^w_S$ is directed complete.*

Proof: Suppose $X = \{f_i \mid i \in I\}$ is directed in $A(T,S)^w_S$, and $f = (\bigsqcup X)$. Then for all i, there exists a path set $\sigma_i = \{w_{i,1}, \dots, w_{i,k_i}\}$ such that $f_i(d)\downarrow$ iff $d \in w_{i,j}$ for some j, and $(\underline{\lambda} d \in w_{i,j}.\, f_i(d)) \in A(T,S)^w$ for all j. Note that if $f_i, f_j \in X$, then there is an f_l such that $f_i, f_j \sqsubseteq f_l$ (by directedness), and hence $\bigcup \sigma_i, \bigcup \sigma_j \subseteq \bigcup \sigma_l$. Since each $\bigcup \sigma_i$ is a subset of the finite set w, there is a k such that for all $f_i \sqsupseteq f_k$, $\bigcup \sigma_i = \bigcup \sigma_k$.

Let $w' = \bigcup \sigma_k$. It is easy to see that $d \in w'$ iff $f(d)\downarrow$. What we need to show is that w' can be subdivided into a set of disjoint sets that forms a path set, and that this path set "witnesses" the membership of f in $A(T,S)^w_S$. There are only finitely many distinct path sets $W_1, \dots, W_k$ for the f_i's above f_k such that for all i, $\bigcup W_i = w'$. Define

$$W = W_1 \sqcap W_2 \sqcap \ldots \sqcap W_k.$$

By Lemma 3.3, W is a path set for f. Now, note that

$$f = \bigcup \{ \bigsqcup \{ (\underline{\lambda} d \in w_j .\, f_i(d)) \mid f_i \sqsupseteq f_k \} \mid w_j \in W \}$$

By the directed completeness of $A(T,S)^w$, for any $w_j \in W$,

$$\bigsqcup \{ (\underline{\lambda} d \in w_j .\, f_i(d)) \mid f_i \sqsupseteq f_k \} \in A(T,S)^w .$$

Thus, $f \in A(T,S)^w_S$. ■

Proposition 3.5 *$Hom_{\mathcal{RCPO}}(A,B)$ is a dcpo.*

Proof: Suppose $X = \{f_i \mid i \in I\} \subseteq Hom_{\mathcal{RCPO}}(A,B)$ is directed. We need to show that $(\bigsqcup X)$ is uniform; it is routine to show that it is continuous. Suppose $h \in A(T,S)^w$. By the uniformity of each f_i, $(h; f_i) \in B(T,S)^w_S$. It is easy to see that $\{\, (h; f_i) \mid i \in I \,\}$ is directed. Thus, by Lemma 3.4, $(h; (\bigsqcup X)) \in B(T,S)^w_S$. ■

3.5 Tensor products and coproducts

The category admits certain standard constructions. For instance, it is easy to show that *void* and *unit*, defined by

$$\begin{array}{rclrcl} |void| & = & \emptyset & |unit| & = & \{\top\} \\ void(T,S)^w & = & \emptyset & unit(T,S)^w & = & \{\, (\underline{\lambda} d \in w' .\, \top) \mid w' \text{ is a } T^w\text{-model} \,\} \end{array}$$

are objects in the category. The relational component of *unit* captures much of the intuition embedded in termination relations: elements in the termination relation are functions which terminate precisely on elements in a T^w-model.

The category has a notion of tensor product. Suppose A, B are objects, and $f : A \to B$ and $g : C \to D$ are morphisms. Define

$$\begin{array}{rcl} |A \otimes B| & = & |A| \times |B| \quad \text{(cartesian product)} \\ (A \otimes B)(T,S)^w & = & \{\langle g,h \rangle \mid g \in A(T,S)^w,\ h \in B(T,S)^w, \\ & & \qquad \text{and } g,h \text{ have the same domain}\} \\ (f \otimes g)\langle x,y \rangle & = & \langle f(x), g(y) \rangle \end{array}$$

Proposition 3.6 $(- \otimes -)$ *is a bifunctor on $\mathcal{RCPO}$, covariant in both arguments.*

Proof: The only non-obvious part is checking that the relations generated by $\otimes$ is directed complete. Suppose $X = \{f_i \mid i \in I\} \subseteq (A \otimes B)(T,S)^w$ is directed. Then all of the elements of X have the same domain, say w' (which is a T^w-model). Note that there exist $g_i \in A(T,S)^w$ and $h_i \in B(T,S)^w$ such that $f_i = \langle g_i, h_i \rangle$. Therefore, it must be the case that the sets

$$X_1 = \{g_i \mid i \in I\} \qquad\qquad X_2 = \{h_i \mid i \in I\}$$

are directed. By hypothesis, $\bigsqcup X_1 \in A(T,S)^w$ and $\bigsqcup X_2 \in B(T,S)^w$. Since $(\bigsqcup X) = \langle \bigsqcup X_1, \bigsqcup X_2 \rangle$, $(\bigsqcup X) \in (A \otimes B)(T,S)^w$. ■

The proof of Proposition 3.6 requires that directed sets in $(A \otimes B)(T,S)^w$ have the same domain, which explains the definition earlier of the order on elements of termination relations.

The category also has coproducts in the usual sense. Suppose A, B are objects, and $f : A \to B$ and $g : C \to D$ are morphisms. Define

$$\begin{array}{rcl} |A \oplus B| & = & |A| \oplus |B| \quad \text{(disjoint union)} \\ (A \oplus B)(T,S)^w & = & \{\, (g; \mathit{inj}_1) \mid g \in A(T,S)^w \,\} \cup \{\, (g; \mathit{inj}_2) \mid g \in B(T,S)^w \,\} \\ (f \oplus g)(x : A \oplus C) & = & \begin{cases} f(y) & \text{if } x = \mathit{inj}_1(y) \\ g(y) & \text{if } x = \mathit{inj}_2(y) \end{cases} \end{array}$$

This definition yields the coproduct in the category, as the following proposition shows.

Proposition 3.7 *$\mathcal{RCPO}$ has coproducts.*

Proof: It is not hard to prove that $\oplus$ is a bifunctor on $\mathcal{RCPO}$. To finish the proof that $\oplus$ yields the coproduct in the category, suppose $f : A \to C$ and $g : B \to C$ are morphisms. Define

$$[f,g](d) = \begin{cases} f(a) & \text{if } d = \mathit{inj}_1(a) \text{ and } f(a)\downarrow \\ g(b) & \text{if } d = \mathit{inj}_2(b) \text{ and } g(b)\downarrow \\ \text{undefined} & \text{otherwise} \end{cases}$$

We need to show that $\mathit{inj}_1 : A \to (A \oplus B)$, $\mathit{inj}_2 : B \to (A \oplus B)$, and $[f,g] : (A \oplus B) \to C$ are morphisms in the category, and that $[f,g]$ is the unique morphism making the diagram

$$\begin{tikzcd} A \arrow[r, "\mathit{inj}_1"] \arrow[dr, "f"'] & (A \oplus B) \arrow[d, "{[f,g]}"] & B \arrow[l, "\mathit{inj}_2"'] \arrow[dl, "g"] \\ & C & \end{tikzcd}$$

commute.

To see that the injections are morphisms, consider the case of inj_1. It is obvious that inj_1 is a partial continuous function, so we only need to verify that inj_1 satisfies the uniformity condition. Suppose $h \in A(T,S)^w$ with domain $w' \subseteq w$. We need to show $(h; \mathit{inj}_1) \in (A \oplus B)(T,S)^w_S$. Since w' is a T^w-model, $\{w'\}$ is itself a path set. It is then easy to see that $(h; \mathit{inj}_1)(d) \downarrow$ iff $d \in w'$, and $(\underline{\lambda} d \in w'.\, (h; \mathit{inj}_1)(d)) = (h; \mathit{inj}_1) \in (A \oplus B)(T,S)^w$ (by definition of $(A \oplus B)(T,S)^w$). Thus $(h; \mathit{inj}_1) \in (A \oplus B)(T,S)^w_S$.

To see that $[f,g]$ is a morphism, it is easy to check that $[f,g]$ is a partial continuous function. We need only verify the uniformity condition. Suppose $h \in (A \oplus B)(T,S)^w$; we want to show that $(h; [f,g]) \in C(T,S)^w_S$. By definition, either $h = (h_1; \mathit{inj}_1)$ for $h_1 \in A(T,S)^w$ or $h = (h_2; \mathit{inj}_2)$ for $h_2 \in B(T,S)^w$. Consider the first case (the second case is analogous). Then $(h; [f,g]) = (h_1; f)$. By uniformity, $(h_1; f) \in C(T,S)^w_S$. Thus, $(h; [f,g]) \in C(T,S)^w_S$.

What remains to be shown is that $[f, g]$ makes the diagram commute, and that $[f, g]$ is the unique such morphism. We consider the left triangle (the right triangle is analogous). So suppose $a \in |A|$. Observe that $f(a)\downarrow$ iff $(inj_1; [f, g])(a)\downarrow$, so if $f(a)\downarrow$, then

$$(inj_1; [f, g])(a) \quad = \quad f(a)$$

and hence the triangle commutes. To see the uniqueness of $[f, g]$, suppose that $h : (A \oplus B) \to C$ is such that the diagram commutes, i.e., $f = (inj_1; h)$ and $g = (inj_2; h)$. Since the injection functions are total functions, the domain of h must be

$$\{d \in |A \oplus B| \mid (d = inj_1(a) \text{ and } f(a)\downarrow) \text{ or } (d = inj_2(b) \text{ and } f(b)\downarrow)\}.$$

Thus $[f, g]$ and h have the same domain.

Now suppose $h(d)\downarrow$. Suppose $d = inj_1(a)$ (the other case is analogous). Then

$$h(d) \quad = \quad (inj_1; h)(a) = f(a) = [f, g](d).$$

That is, $h = [f, g]$. ■

3.6 Function space

The category also has an operation associated with a space of functions. Suppose A, B are objects, and $f : A \to B$ and $g : C \to D$ are morphisms. Define

$$\begin{array}{rcl}
|A \Rightarrow B| & = & Hom_{\mathcal{RCPO}}(A, B) \\
(A \Rightarrow B)(T, S)^w & = & \{ f \mid \forall \varphi : v \xrightarrow{\mathcal{C}} w, g \in A(T, S)^v. \\
& & \quad (\underline{\lambda} d \in v.\ (f(\varphi(d))\ (g(d)))) \in B(T, S)^v_S \} \\
(f \Rightarrow g)(h : B \to C) & = & (f; h; g) : A \to D
\end{array}$$

The definition of function space on relations is the same as the one for Kripke relations (Jung and Tiuryn, 1993; O'Hearn and Riecke, 1995), except that the result of applying related arguments to related results must be in the computational relation, not the termination relation. Again, this is reminiscent of the monad style of semantics (Moggi, 1991).

We must verify that $\Rightarrow$ is a bifunctor, contravariant in the first argument and covariant in the second. The proof is divided into two parts:

Proposition 3.8 *$(A \Rightarrow B)$ is an object.*

Proof: By Proposition 3.5, we know that $|A \Rightarrow B|$ is directed complete. Suppose $\mathcal{C}$ is an index category, T is a $\mathcal{C}$-termination theory, and S is a $\mathcal{C}, T$-path theory. We need to show that $(A \Rightarrow B)(T, S)$ is a $\mathcal{C}, T$-termination relation and the concreteness condition holds.

First, we must show that $(A \Rightarrow B)(T, S)$ is a $\mathcal{C}, T$-termination relation. To see directed completeness, suppose $X = \{f_i \mid i \in I\} \subseteq (A \Rightarrow B)(T, S)^w$ is directed. Then all of the elements of X have the same domain, say w' (which is a T^w-model). To see $(\bigsqcup X) \in (A \Rightarrow B)(T,S)^w$, suppose $\varphi : v \xrightarrow{\mathrm{e}} w$ and $h \in A(T,S)^v$. Then for all $i \in I$, $g_i = (\underline{\lambda} j \in v.\, f_i\ (\varphi(j))\ (h(j))) \in B(T,S)^v_S$. Since $\{g_i \mid i \in I\}$ is directed, by Lemma 3.4 $\bigsqcup g_i \in B(T,S)^v_S$. Thus,

$$(\underline{\lambda} j \in v.\, (\bigsqcup X)\ (\varphi(j))\ (h(j))) \in B(T,S)^v_S$$

as needed.

The difficulty lies in showing downward closure and Kripke monotonicity. Suppose $f \in (A \Rightarrow B)(T,S)^w$ with domain w', and suppose w'' is a T^w-model with $w'' \subseteq w'$. Let $f' = (\underline{\lambda} d \in w''.\, f(d))$. To see that $f' \in (A \Rightarrow B)(T,S)^w$, suppose $\varphi : v \xrightarrow{\mathrm{e}} w$ and $g \in A(T,S)^v$. Then we know that

$$h = (\underline{\lambda} d \in v.\, f\ (\varphi(d))\ (g(d))) \in B(T,S)^v_S.$$

Thus, there is some path set $\{v_1, \ldots, v_n\}$ such that $h(d)\!\downarrow$ iff $d \in v_i$ for some i, and for all i,

$$(\underline{\lambda} d \in v_i.\, f\ (\varphi(d))\ (g(d))) \in B(T,S)^v.$$

Let $h' = (\underline{\lambda} d \in v.\, f'\ (\varphi(d))\ (g(d)))$. We need to show that $h' \in B(T,S)^v_S$. First, we need to build up the right path set. Notice that $v'' = \varphi^{-1}(w'')$ is a T^v-model by the definition of $\mathcal{C}$-termination theories. Since $X = \{v_1, \ldots, v_n\}$ and $X'' = \{v''\}$ are path sets, $X \sqcap X'' = \{v'_1, \ldots, v'_k\}$ is a path set (recall that $X \sqcap X''$ is the set of all $(v \cap v'')$, with $v \in X$ and $v'' \in X''$, that are non-empty). It is clear that $h'(d)\!\downarrow$ iff $d \in v'_i$ for some i. Also, since each $v'_i \subseteq v_j$ for some j, it follows by downward closure that $(\underline{\lambda} d \in v'_i.\, f\ (\varphi(d))\ (g(d))) \in B(T,S)^v$ for any i. Because $(\underline{\lambda} d \in v'_i.\, f\ (\varphi(d))\ (g(d))) = (\underline{\lambda} d \in v'_i.\, f'\ (\varphi(d))\ (g(d)))$, it follows that

$$(\underline{\lambda} d \in v'_i.\, f'\ (\varphi(d))\ (g(d))) \in B(T,S)^v.$$

Thus, $h' \in B(T,S)^v_S$ as desired.

To see Kripke monotonicity, suppose $\varphi : v \xrightarrow{\mathrm{e}} w$ and $f \in (A \Rightarrow B)(T,S)^w$ with domain w'. Let $v' = \varphi^{-1}(w')$; we know that v' is a T^v-model since T is a $\mathcal{C}$-termination theory. We want

$$(\underline{\lambda} d \in v'.\, f(\varphi(d))) \in (A \Rightarrow B)(T,S)^v.$$

So suppose $\psi : u \xrightarrow{\mathrm{e}} v$ and $g \in A(T,S)^u$. Then

$$\begin{array}{rcl} (\underline{\lambda} d \in u.\, (\underline{\lambda} e \in v'.\, f(\varphi(e)))\ (\psi\ d)\ (g(d))) & = & (\underline{\lambda} d \in u.\, f(\varphi(\psi(d)))\ (g(d))) \\ & \in & B(T,S)^u_S \end{array}$$

by the fact that $(\psi;\varphi) : u \xrightarrow{\mathrm{e}} w$. Thus, $(\underline{\lambda} d \in v'.\, f(\varphi(d))) \in (A \Rightarrow B)(T,S)^v$.

Finally, to show concreteness, suppose $f \in |A \Rightarrow B|$; we need to show that $(\underline{\lambda} i \in w.\, f) \in (A \Rightarrow B)(T,S)^w$. From the definition, we must show that, for $\varphi : v \xrightarrow{\mathrm{e}} w$ and $h \in A(T,S)^v$, $(\underline{\lambda} j \in v.\, ((\underline{\lambda} i \in w.\, f)\ (\varphi(j)))\ (h(j))) \in B(T,S)^v_S$. But this reduces to $(\underline{\lambda} j \in v.\, f\ (h(j))) \in B(T,S)^v_S$, which is the uniformity condition on $\mathcal{RCPO}$-morphisms. ■

Proposition 3.9 $(- \Rightarrow -)$ *is a bifunctor on* $\mathcal{RCPO}$ *that is contravariant in the first argument and covariant in the second.*

Proof: By Proposition 3.8, if A, B are objects, then $|A \Rightarrow B|$ is too. To check the functor part, suppose $f : A' \to A$ and $g : B \to B'$ in $\mathcal{RCPO}$. The uniformity condition is preserved by composition (see Lemma 3.2), as is relevant domain-theoretic structure, so we may conclude that for any h, $(f; h; g) \in |A' \Rightarrow B'|$. To see that $(f \Rightarrow g)$ *as a function* satisfies the uniformity condition, consider $m \in (A \Rightarrow B)(T, S)^w$: we need to show that

$$(\underline{\lambda} d \in w.\, f; m(d); g) \in (A' \Rightarrow B')(T, S)^w_S.$$

If we can show that $h' = (\underline{\lambda} d \in w'.\, \underline{\lambda} a.\, g(m\ d\ f(a))) \in (A' \Rightarrow B')(T, S)^w$, where w' is the domain of m, then we will be done—the required path set will be $\{w'\}$. So suppose $\varphi : v \xrightarrow{e} w$ and $h \in A'(T, S)^v$ and let

$$\begin{aligned} h'' &= (\underline{\lambda} d \in v.\, h'(\varphi(d))(h(d))) \\ &= (\underline{\lambda} d \in v.\, g(m(\varphi(d))(f(h(d))))). \end{aligned}$$

We then need to show that $h'' \in B'(T, S)^v_S$. By the uniformity of f, we get $(h; f) \in A(T, S)^v_S$. Thus, there exists a path set $\{v_1, \ldots, v_n\}$ such that $f(h(d))\downarrow$ iff $d \in v_i$ for some i, and $f_i = (\underline{\lambda} d \in v_i.\, f(h(d))) \in A(T, S)^v$ for all i. Thus, for each f_i,

$$h_i = (\underline{\lambda} d \in v_i.\, m\ (\varphi(d))\ (f_i(d))) \in B(T, S)^v_S$$

Therefore, for each i, there exists a path set $\{v_{i,1}, \ldots, v_{i,k_i}\}$ such that

- $h_i(d)\downarrow$ iff $d \in v_{i,j}$ for some j; and
- $h_{i,j} = (\underline{\lambda} d \in v_{i,j}.\, m\ (\varphi(d))\ (f_i(d))) \in B(T, S)^v$ for all j.

Note as well that each $v_{i,j} \subseteq v_i$. By the uniformity of g, $(h_{i,j}; g) \in B'(T, S)^v_S$. Thus, for each i, j, there exists a path set $\{v_{i,j,1}, \ldots, v_{i,j,l_{i,j}}\}$ such that

- $h_{i,j}(d)\downarrow$ iff $d \in v_{i,j,k}$ for some k; and
- $h_{i,j,k} = (\underline{\lambda} d \in v_{i,j,k}.\, g(m\ (\varphi(d))\ (f_i(d)))) \in B'(T, S)^v$ for all k.

Again, all of the $v_{i,j,k} \subseteq v_{i,j}$. Thus $\{v_{1,1,1}, \ldots, v_{n,k_n,l_{n,k_n}}\}$ is a path set and

- $h''(d)\downarrow$ iff $d \in v_{i,j,k}$ for some i, j and k; and
- $(\underline{\lambda} d \in v_{i,j,k}.\, h''(d)) = h_{i,j,k} \in B'(T, S)^v$ for all i, j and k.

It follows that $h'' \in B'(T, S)^v_S$ and thus $h' \in (A' \Rightarrow B')(T, S)^w$ as desired. ■

3.7 Partial CCC

$\mathcal{RCPO}$ is a **partial cartesian closed category** (Curien and Obtułowicz, 1989). Since there are a number of conditions to check, we break the proof up into three parts: $\mathcal{RCPO}$ is a category of partial morphisms, it is partial cartesian, and it is partial cartesian closed.

A **category of partial morphisms** is a category with a notion of "total" morphisms and a "restriction relation" $\leqslant$ on morphisms. Moreover, the identities must be total, and composition must preserve totality and be monotonic with respect to $\leqslant$. In $\mathcal{RCPO}$, the total morphisms are simply the total functions. If f and f' are morphisms from A to B, then $f \leqslant f'$ iff

$$f(a){\downarrow} \text{ implies } f(a) = f'(a), \text{ for all } a \in |A|.$$

It is thus easy to see that

Proposition 3.10 *$\mathcal{RCPO}$ is a category of partial morphisms.*

A category of partial morphisms is **cartesian** if it has an object U—called a **domain classifier**—with a map $\bigcirc_A : A \to U$ for any object A. There must also be a morphism $f \cap g : A \to U$ for every $f, g : A \to U$, and the following properties must hold:

- If $f : A \to U$, then $f \leqslant \bigcirc_A$.
- The total arrows are exactly those arrows $f : A \to B$ such that $f; \bigcirc_B = \bigcirc_A$.
- If $f, f', g : A \to B$ are such that $f, f' \leqslant g$ and $f; \bigcirc_B \leqslant f'; \bigcirc_B$, then $f \leqslant f'$.
- If $g : B \to U$ and $h, h' : A \to B$ are such that $h \leqslant h'$, then $(h; g) = (h'; g) \cap (h; \bigcirc_B)$.

Furthermore, for a category of partial morphisms to be cartesian it must have, for any pair of objects A, B, a partial product given by an object $A \otimes B$, projection arrows $\mathit{proj}_1 : A \otimes B \to A$ and $\mathit{proj}_2 : A \otimes B \to B$, and a pairing operation, associating $\langle f, g\rangle : A \to B \otimes C$ with $f : A \to B, g : A \to C$ such that the following properties hold:

- proj_1 and proj_2 are total.
- Pairing is monotonic with respect to $\leqslant$ in both arguments.
- For any $h : A \to (B \otimes C)$, $\langle h; \mathit{proj}_1, h; \mathit{proj}_2\rangle = h$
- If $f : A \to B$ and $g : A \to C$, then $(\langle f, g\rangle; \mathit{proj}_1) \leqslant f$ and $(\langle f, g\rangle; \mathit{proj}_2) \leqslant g$.
- If $f : A \to B$ and $g : A \to C$, then for all $h : D \to A$ we have $(h; \langle f, g\rangle) = \langle (h; f), (h; g)\rangle$.

- If $f : A \to B$ and $g : A \to C$, then $(\langle f, g\rangle; \bigcirc_{B\otimes C}) = (f; \bigcirc_B) \cap (g; \bigcirc_C)$.

Proposition 3.11 $\mathcal{RCPO}$ *is a partial cartesian category.*

Proof: The domain classifier is the object *unit*; for every object A, we pick the morphism $\bigcirc_A : A \to \mathit{unit}$ to be the unique morphism mapping all elements of A to $\top$. Also, define

$$(f \cap f')(a) = \begin{cases} \top & \text{if } f(a)\downarrow \text{ and } f'(a)\downarrow \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The partial product is the tensor product defined previously, and the projection arrows and pairing operation the obvious ones. It is easy to see that these maps exist, and that the above properties hold. ■

Let $(A \to_T B)$ denote the set of total morphisms between objects A, B in $\mathcal{RCPO}$. A partial cartesian category is a **partial cartesian closed category** if there exists a binary operation $\Rightarrow$ of partial exponentiation on objects such that for any objects A, B, C there exist natural transformations *curry*$(-)$ from $(A \otimes B) \to C$ onto $A \to_T (B \Rightarrow C)$, and *uncurry*$(-)$ from $A \to_T (B \Rightarrow C)$ onto $(A \otimes B) \to C$, $\mathit{uncurry}(\mathit{curry}(f)) = f$ and $\mathit{curry}(\mathit{uncurry}(g)) = g$, and for any $f' : D \to A$,

$$(f'; \mathit{curry}(f)) \leqslant \mathit{curry}(\langle \mathit{proj}_1; f', \mathit{proj}_2\rangle; f).$$

Proposition 3.12 $\mathcal{RCPO}$ *is a partial cartesian closed category.*

Proof: The partial exponentiation is the function space $\Rightarrow$ defined previously. For $f : (A \otimes B) \to C$ and $g : A \to_T (B \Rightarrow C)$, define

$$\begin{aligned} \mathit{curry}(f) &= (\underline{\lambda} a \in |A|.\, \underline{\lambda} b \in |B|.\, f\ \langle a, b\rangle) \\ \mathit{uncurry}(g) &= (\underline{\lambda}\langle a, b\rangle \in |A \otimes B|.\, g(a)(b)). \end{aligned}$$

We first need to show that the maps are well-defined, and that they satisfy the uniformity conditions. To see that $\mathit{curry}(f)$ is a well-defined total function from $|A|$ to $|B \Rightarrow C|$, suppose $a \in |A|$ and $h \in B(T, S)^w$; we need to show $(h; \mathit{curry}(f)(a)) \in C(T, S)^w_S$.

By the concreteness condition $h' = (\underline{\lambda} d \in w'.\, a) \in A(T, S)^w$, where w' is the domain of h. Then $\langle h', h\rangle \in (A \otimes B)(T, S)^w$ and by uniformity of f we get $(\langle h', h\rangle; f) \in C(T, S)^w_S$. Since $(\langle h', h\rangle; f) = (h; \mathit{curry}(f)(a))$, we are done.

To see the uniformity of $\mathit{curry}(f)$, suppose $h_A \in A(T, S)^w$; we need to show $(h_A; \mathit{curry}(f)) \in (B \Rightarrow C)(T, S)^w_S$. Let w' be the domain of h_A. Since $(\underline{\lambda} d \in w'.\, \mathit{curry}(f)(h_A(d))) = (h_A; \mathit{curry}(f))$ and $\{w'\}$ is a path set, it suffices to prove $(h_A; \mathit{curry}(f)) \in (B \Rightarrow C)(T, S)^w$. Now, given $\varphi : v \to w$ and
$h_B \in B(T, S)^v$ we must prove $(\underline{\lambda} d \in v.\, (h_A; \mathit{curry}(f))(\varphi(d))(h_B(d))) \in C(T, S)^v_S$. By Kripke Monotonicity of $A(T, S)^w$, $h'_A = (\varphi; h_A) \in A(T, S)^v$. Let v_1 be the domain of h'_A and v_2 be the domain of h_B, and $v' = (v_1 \cap v_2)$. By downward closure,

$$h''_A = (\underline{\lambda} d \in v'.\, h'_A(d)) \in A(T,S)^v$$
$$h''_B = (\underline{\lambda} d \in v'.\, h_B(d)) \in B(T,S)^v$$

Therefore $\langle h''_A, h''_B \rangle \in (A \otimes B)(T,S)^v$. By the uniformity of f,

$$\begin{array}{rcl}(\underline{\lambda} d \in v.\, (h_A; \mathit{curry}(f))(\varphi(d))(h_B(d))) & = & (\underline{\lambda} d \in v.\, \mathit{curry}(f)(h''_A(d))(h''_B(d))) \\ & = & (\underline{\lambda} d \in v.\, (\langle h''_A, h''_B \rangle; f)(d)) \\ & \in & C(T,S)^v_S\end{array}$$

as desired.

Similarly, we need to show that $\mathit{uncurry}(g)$ is a well-defined partial function from $|A \otimes B|$ to $|C|$ and that it satisfies the uniformity condition. Well-definedness follows immediately from definition. Now for uniformity, suppose $h \in (A \otimes B)(T,S)^w$; we need to show $(h; \mathit{uncurry}(g)) \in C(T,S)^w_S$. Notice that h has the form $h = \langle h_A, h_B \rangle$, where $h_A \in A(T,S)^w$ and $h_B \in B(T,S)^w$. By uniformity of g, $(h_A; g) \in (B \Rightarrow C)(T,S)^w_S$. This means that there exists a path set $\{w_1, \ldots, w_n\}$ such that

- $(h_A; g)(d)\!\downarrow$ iff $d \in w_i$ for some i; and
- $(\underline{\lambda} d \in w_i.\, (h_A; g)(d)) \in (B \Rightarrow C)(T,S)^w$ for all i.

Now, since $h_B \in B(T,S)^w$ and id_w (the identity) is a morphism in $\mathcal{C}$, we have

$$g_i = (\underline{\lambda} d \in w.\, (\underline{\lambda} e \in w_i.\, (h_A; g)(e))(d)(h_B(d))) \in C(T,S)^w_S.$$

Thus there exist path sets $\{w_{i,1}, \ldots, w_{i,k_i}\}$ such that

- $g_i(d)\!\downarrow$ iff $d \in w_{i,j}$ for some i, j; and
- $(\underline{\lambda} d \in w_{i,j}.\, g_i(d)) \in C(T,S)^w$ for all i, j.

¿From the observation that $w_i \supseteq w_{i,j}$, it follows that $\{w_{1,1}, \ldots, w_{n,k_n}\}$ is a path set. Furthermore, for

$$g' = (\underline{\lambda} d \in w.\, g(h_A(d))(h_B(d))),$$

we have that

- $g'(d)\!\downarrow$ iff $d \in w_{i,j}$ for some i, j; and
- $(\underline{\lambda} d \in w_{i,j}.\, g'(d)) \in C(T,S)^w$ for all i, j.

Since $g' = (h; \mathit{uncurry}(g))$, we are done.

It is not hard to prove that $\mathit{curry}(-)$ and $\mathit{uncurry}(-)$ are natural transformations, and that they form an isomorphism pair. Thus, suppose $f' : D \to A$; we need to show that

$$(f'; \mathit{curry}(f)) \leqslant \mathit{curry}(\langle \mathit{proj}_1; f', \mathit{proj}_2 \rangle; f).$$

Given $d \in |D|$, if $f'(d)\uparrow$ then $(f'; \mathit{curry}(f))(d)$ is not defined either and there is nothing to show. So suppose $f'(d)\downarrow$. Then it suffices to show

$$(f'; \mathit{curry}(f))(d) = \mathit{curry}(\langle \mathit{proj}_1; f', \mathit{proj}_2\rangle; f)(d).$$

Now, if $f\langle f'(d), b\rangle\uparrow$, then both the left and right hand side functions above are undefined at b as well. If on the other hand $f\langle f'(d), b\rangle\downarrow$ then both of the above functions will equal that value on b. ■

3.8 Colimits

Coproducts and the partial cartesian closed structure give most of the structure necessary to interpret FPC types. The only part left is the interpretation of recursive types. To this end, we rework the standard colimit construction from domain theory (Abramsky and Jung, 1994; Gunter, 1992; Plotkin, 1985).

The basis of the colimit construction is embedding-projection pairs. Given objects A, B, then f is an **embedding-projection pair** (ep-pair for short) if $f = \langle f^e : A \to B, f^p : B \to A\rangle$, $(f^e; f^p) = id_A$, and $(f^p; f^e) \sqsubseteq id_B$. We abuse notation and write $f : A \to B$ when f is an ep-pair from A to B, and $id : A \to A$ for the ep-pair $\langle id, id\rangle$. The composition of ep-pairs is pointwise: if $f : A \to B$ and $g : B \to C$, then $(f; g) = \langle f^e; g^e, g^p; f^p\rangle : A \to C$.

Colimits are formed from chains of objects connected by ep-pairs. Formally, an **expanding sequence** is a tuple

$$(\{D_n \mid n \geq 0\}, \{f_n \mid f_n : D_n \to D_{n+1} \text{ is an ep-pair}\}).$$

When $n \leq m$, we write $f_{mn} : D_n \to D_m$ for the ep-pair defined by induction as follows:

$$\begin{array}{rcl} f_{nn} & = & id \\ f_{(m+1)n} & = & f_{mn}; f_m. \end{array}$$

Given an expanding sequence, define D by

$$\begin{array}{rcl} |D| & = & \{g \in [\mathbf{N} \to_p \bigcup D_i] \mid g(i)\downarrow \text{ implies } g(i) \in D_i, \text{ and} \\ & & \qquad\qquad \text{for all } n \leq m,\, g(n) \simeq f^p_{mn}(g(m))\} \\ D(T,S)^w & = & \{h \in [w' \to_t D] \mid w' \text{ is a } T^w\text{-model, and} \\ & & \qquad\qquad \text{for all } \varphi : v \xrightarrow{\mathrm{e}} w \text{ and } n \in \mathbf{N}, \\ & & \qquad\qquad (\underline{\lambda} d \in v.\, h\,(\varphi(d))\,(n)) \in D_n(T,S)^v_S\} \end{array}$$

where ordering on $|D|$ is pointwise ordering on the functions. The ep-pairs $\langle \mu^e_m : D_m \to D, \mu^p_m : D \to D_m\rangle$, defined by

$$\begin{array}{rcl} \mu^p_m(x) & \simeq & x(m) \\ \mu^e_m(x)(l) & \simeq & \displaystyle\bigsqcup_{k \geq m,l} (f^p_{kl}(f^e_{km}(x))) \end{array}$$

make D into a colimit of the expanding sequence, as the next two lemmas show.

Lemma 3.13 *D is an object in $\mathcal{RCPO}$.*

Proof: It is easy to show that $|D|$ is directed complete. To see that $D(T,S)^w$ is a w-termination relation, we check directed completeness and leave the straightforward checks of downward closure, Kripke monotonicity, and concreteness to the reader. Suppose that $\{h_i \mid i \in I\} \subseteq D(T,S)^w$ is directed, where each h_i has domain w'. Let h be the least upper bound of the h_i's; we need to show that $h \in D(T,S)^w$. To show this, consider any $\varphi : v \to w$ and $n \in \mathbf{N}$. Then

$$(\underline{\lambda} d \in v.\, h\,(\varphi(d))\,(n)) = \bigsqcup_{i \in I} (\underline{\lambda} d \in v.\, h_i\,(\varphi(d))\,(n)) \in D_n(T,S)^w_S$$

by the directed completeness of the $D_n(T,S)^w_S$'s. Thus,

$$(\underline{\lambda} d \in v.\, h\,(\varphi(d))\,(n)) \in D_n(T,S)^w_S$$

as required. ■

Lemma 3.14 *For all m, $\mu^e_m : D_m \to D$ and $\mu^p_m : D \to D_m$ are morphisms. Moreover, $(\mu^p_m; \mu^e_m) \sqsubseteq id_D$, $(\mu^e_m; \mu^p_m) = id_{D_m}$, and*

$$\bigsqcup_{m \geq 0} \mu^p_m; \mu^e_m \quad = \quad id_D$$

Proof: The only difficult part of the proof lies in showing that μ^e_m and μ^p_m satisfy the uniformity property. To see that μ^p_m is uniform, suppose $h \in D(T,S)^w$. Then $(h; \mu^p_m) = (\underline{\lambda} d \in w.\, h\,(d)\,(m)) \in D_m(T,S)^w_S$, which is what we needed to show.

The proof of the uniformity of μ^e_m is more involved. Suppose $h \in D_m(T,S)^w$, and consider $g = (h; \mu^e_m)$. We show that $g \in D(T,S)^w$, which will show that $g \in D(T,S)^w_S$. Note that g has the same domain as h, so consider any given $\varphi : v \xrightarrow{e} w$ and $n \in \mathbf{N}$. Note that

$$(\underline{\lambda} d \in v.\, g\,(\varphi(d))\,(n)) = (\underline{\lambda} d \in v.\, \mu^e_m\,\,(h\,\,(\varphi(d)))\,\,n) = \bigsqcup_{k \geq m,n} (\varphi; h; f^e_{km}; f^p_{kn})$$

By the uniformity of f^p_{kn} and f^e_{km}, we know that $(\varphi; h; f^e_{kn}; f^p_{km}) \in D_n(T,S)^v_S$. Thus, by the directed completeness of $D_n(T,S)^w_S$,

$$(\underline{\lambda} d \in v.\, g\,(\varphi(d))\,(n)) \in D_n(T,S)^v_S$$

as required. ■

4 Interpretation of FPC

The constructions in the category $\mathcal{RCPO}$ in Section 3 can now be used to build a model of FPC (Gunter, 1992; Riecke and Subrahmanyam, 1997). A model has both a meaning for types and for terms. Types denote objects in the category, and terms denote morphisms.

The meaning of a closed type is clear except for the meanings of recursive types, which necessitates describing the meaning of an open type expression. For this reason, the meaning of a type is a *functor* from its free type variables to the category $\mathcal{RCPO}$. In making this precise, most of the difficulty lies in finding the right category of free type variables. To this end, define a **pre-ep-pair** to be a pair $f = \langle f^e, f^p \rangle$ of morphisms $f^e : D \to E$ and $f^p : E \to D$ in $\mathcal{RCPO}$; we write $f : D \hookrightarrow E$ as shorthand for saying that f is such a pre-ep-pair. Note that a pre-ep-pair is just like an ep-pair, but without the requirements that $(f^e; f^p) = id$ and $(f^p; f^e) \sqsubseteq id$. The use of pre-ep-pairs instead of ep-pairs makes certain theorems (particularly those in Appendix B) easier to prove.

A **type environment** η is a function from type variables to objects of $\mathcal{RCPO}$, and a **type environment map** $\pi : \eta \to \eta'$ is a function from type variables to pre-ep-pairs such that $\pi(\alpha) : \eta(\alpha) \hookrightarrow \eta'(\alpha)$ for all α. The category $\mathcal{E}$ has type environments as objects, and type environment maps as morphisms. It is simple to check that $\mathcal{E}$ is a category.

The meaning of a type s, then, is a functor $[\![s]\!] : \mathcal{E} \to \mathcal{RCPO}$. The definition on objects of $\mathcal{E}$—except in the case of recursive types—is

$$\begin{array}{rcl}
[\![\alpha]\!]\eta &=& \eta(\alpha) \\
[\![\mathtt{unit}]\!]\eta &=& unit \\
[\![s \oplus t]\!]\eta &=& ([\![s]\!]\eta \oplus [\![t]\!]\eta) \\
[\![s \otimes t]\!]\eta &=& ([\![s]\!]\eta \otimes [\![t]\!]\eta) \\
[\![s \Rightarrow t]\!]\eta &=& ([\![s]\!]\eta \Rightarrow [\![t]\!]\eta)
\end{array}$$

The operation on morphisms of $[\![s]\!]\pi$—except for recursive types—is

$$\begin{array}{rcl}
[\![\alpha]\!]\pi &=& \pi(\alpha) \\
[\![\mathtt{unit}]\!]\pi &=& \langle id_{unit}, id_{unit} \rangle \\
[\![s \oplus t]\!]\pi &=& ([\![s]\!]\pi \oplus [\![t]\!]\pi) \\
[\![s \otimes t]\!]\pi &=& ([\![s]\!]\pi \otimes [\![t]\!]\pi) \\
[\![s \Rightarrow t]\!]\pi &=& ([\![s]\!]\pi \Rightarrow [\![t]\!]\pi)
\end{array}$$

where, abusing notation, $\oplus$, $\otimes$, and $\Rightarrow$ work on pre-ep-pairs as follows:

$$\begin{array}{rcl}
([\![s]\!]\pi \oplus [\![t]\!]\pi) &=& \langle ([\![s]\!]\pi)^e \oplus ([\![t]\!]\pi)^e, ([\![s]\!]\pi)^p \oplus ([\![t]\!]\pi)^p \rangle \\
([\![s]\!]\pi \otimes [\![t]\!]\pi) &=& \langle ([\![s]\!]\pi)^e \otimes ([\![t]\!]\pi)^e, ([\![s]\!]\pi)^p \otimes ([\![t]\!]\pi)^p \rangle \\
([\![s]\!]\pi \Rightarrow [\![t]\!]\pi) &=& \langle ([\![s]\!]\pi)^p \Rightarrow ([\![t]\!]\pi)^e, ([\![s]\!]\pi)^e \Rightarrow ([\![t]\!]\pi)^p \rangle
\end{array}$$

(Recall the actions of $\oplus$, $\otimes$, and $\Rightarrow$ on morphisms from Section 3.)

The case of recursive types requires more work. Suppose $s = (\mathtt{rec}\ \alpha.\ t)$. Let $\left\langle T^i_{s,\eta}, f^i_{s,\eta} \mid i \geq 0 \right\rangle$ be the expanding sequence given by

$$\begin{array}{rclrcl}
T^0_{s,\eta} &=& void & f^0_{s,\eta} &=& !_{T^1_{s,\eta}} \\
T^{n+1}_{s,\eta} &=& [\![t]\!](\eta[\alpha \mapsto T^n_{s,\eta}]) & f^{n+1}_{s,\eta} &=& [\![t]\!](id[\alpha \mapsto f^n_{s,\eta}])
\end{array}$$

$$\begin{tikzcd}
T^0_{s,\eta_1} \arrow[r, "f^0_{s,\eta_1}"] \arrow[d, "p^0_{s,\pi}"'] & T^1_{s,\eta_1} \arrow[r, "f^1_{s,\eta_1}"] \arrow[d, "p^1_{s,\pi}"'] & T^2_{s,\eta_1} \arrow[r, "f^2_{s,\eta_1}"] \arrow[d, "p^2_{s,\pi}"'] & \cdots & \llbracket s\rrbracket\eta_1 \arrow[d, "\llbracket s\rrbracket\pi"] \\
T^0_{s,\eta_2} \arrow[r, "f^0_{s,\eta_2}"] & T^1_{s,\eta_2} \arrow[r, "f^1_{s,\eta_2}"] & T^2_{s,\eta_2} \arrow[r, "f^2_{s,\eta_2}"] & \cdots & \llbracket s\rrbracket\eta_2
\end{tikzcd}$$

Figure 1: Colimits and definition of $\llbracket s\rrbracket(\pi)$, where $s = (\texttt{rec}\ \alpha.\ t)$.

where $!_D$ is the unique ep-pair (not just pre-ep-pair) from *void* to D. This is an expanding sequence because the maps $f^n_{s,\eta}$'s are ep-pairs (by a simple induction on the definition). If D is the colimit of the expanding sequence, then define $\llbracket s\rrbracket\eta = D$. Moreover, define the morphisms $\mathit{intro}_{s,\eta} : \llbracket t[s/\alpha]\rrbracket\eta \to \llbracket s\rrbracket\eta$ and $\mathit{elim}_{s,\eta} : \llbracket t[s/\alpha]\rrbracket\eta \to \llbracket s\rrbracket\eta$ and

$$\begin{aligned}
\mathit{intro}_{s,\eta} &= \bigsqcup_{n\geq 0} \llbracket t\rrbracket \mathit{id}[\alpha \mapsto \mu^n_{s,\eta}]^p ; \mu^{n+1,e}_{s,\eta} \\
\mathit{elim}_{s,\eta} &= \bigsqcup_{n\geq 0} \mu^{n+1,p}_{s,\eta} ; \llbracket t\rrbracket \mathit{id}[\alpha \mapsto \mu^n_{s,\eta}]^e
\end{aligned}$$

These maps will be used to give meaning to `intro` and `elim`.

For the morphism part of the functor in the case of recursive types, consider the diagram in Figure 1. Given $\pi : \eta_1 \to \eta_2$, consider the expanding sequences $\langle T^i_{s,\eta_1}, f^i_{s,\eta_1} \mid i \geq 0\rangle$, and $\langle T^i_{s,\eta_2}, f^i_{s,\eta_2} \mid i \geq 0\rangle$, with colimiting cocones $\langle \llbracket s\rrbracket\eta_1, \mu^i_{s,\eta_1} \mid i \geq 0\rangle$ and $\langle \llbracket s\rrbracket\eta_2, \mu^i_{s,\eta_2} \mid i \geq 0\rangle$, respectively. Define the family of pre-ep-pairs $p^n_{s,\pi}$, where $p^n_{s,\pi}$ is a pre-ep-pair from T^n_{s,η_1} to T^n_{s,η_2} by

$$\begin{aligned}
p^0_{s,\pi} &= !_{\mathit{void}} \\
p^{k+1}_{s,\pi} &= \llbracket t\rrbracket(\pi[\alpha \mapsto p^k_{s,\pi}])
\end{aligned}$$

A simple induction argument shows that the diagram in Figure 1 commutes (assuming that $\llbracket\cdot\rrbracket$ is a functor, which holds by construction). From this fact, it is clear that

$$\langle \llbracket s\rrbracket\eta_2, \mu^n_{s,\eta_2} \circ p^n_{s,\pi} \mid n \geq 0\rangle$$

is a cocone for the expanding sequence $\langle T^i_{s,\eta_1}, f^i_{s,\eta_1} \mid i \geq 0\rangle$. Define $\llbracket s\rrbracket\pi$ to be the unique mediating map from $\llbracket s\rrbracket\eta_1$ to $\llbracket s\rrbracket\eta_2$, which must exist by the colimiting properties.

The meaning of terms can now be given using the combinators of the category. If $\Gamma = x_1 : t_1, ..., x_n : t_n$, define $\llbracket\Gamma\rrbracket = \llbracket t_1\rrbracket \otimes \ldots \otimes \llbracket t_n\rrbracket$. (If Γ is empty, $\llbracket\Gamma\rrbracket$ is the object *unit*). Elements of $|\llbracket\Gamma\rrbracket|$ are called **environments**. For an environment $\rho \in |\llbracket\Gamma\rrbracket|$, we write $\rho(x)$ for projection to the component corresponding to variable x. The meaning of a judgement $\Gamma \vdash M : t$ is an $\mathcal{RCPO}$-morphism $[\![\Gamma \vdash M : t]\!] : [\![\Gamma]\!] \to [\![t]\!]$. The definition is by induction on the structure of terms:

$$[\![\Gamma, x : t \vdash x : t]\!]\rho = \rho(x) \qquad\qquad [\![\Gamma \vdash \langle\rangle : \texttt{unit}]\!]\rho = \top$$

$$[\![\Gamma \vdash (\lambda x : s.\, M) : (s \Rightarrow t)]\!]\rho = f, \quad \text{where } f(d) \simeq [\![\Gamma, x : s \vdash M : t]\!]\rho[x \mapsto d]$$

$$[\![\Gamma \vdash (M\ N) : t]\!]\rho \simeq ([\![\Gamma \vdash M : (s \Rightarrow t)]\!]\rho)\ ([\![\Gamma \vdash N : s]\!]\rho)$$

$$[\![\Gamma \vdash (\texttt{intro}_{\texttt{rec}\ \alpha.\ s}\ M) : (\texttt{rec}\ \alpha.\ s)]\!]\rho \simeq \mathit{intro}([\![\Gamma \vdash M : s[\texttt{rec}\ \alpha.\ s/\alpha]]\!]\rho)$$

$$[\![\Gamma \vdash (\texttt{elim}_{\texttt{rec}\ \alpha.\ s}\ M) : s[\texttt{rec}\ \alpha.\ s/\alpha]]\!]\rho \simeq \mathit{elim}([\![\Gamma \vdash M : (\texttt{rec}\ \alpha.\ s)]\!]\rho)$$

$$[\![\Gamma \vdash \langle M, N\rangle : (t_1 \otimes t_2)]\!]\rho \simeq \langle [\![\Gamma \vdash M : t_1]\!]\rho, [\![\Gamma \vdash N : t_2]\!]\rho\rangle$$

$$[\![\Gamma \vdash (\texttt{proj}_i\ M) : t_i]\!]\rho \simeq \mathit{proj}_i([\![\Gamma \vdash M : (t_1 \otimes t_2)]\!]\rho)$$

$$[\![\Gamma \vdash (\texttt{inj}_i\ M) : (t_1 \oplus t_2)]\!]\rho \simeq \mathit{inj}_i([\![\Gamma \vdash M : t_i]\!]\rho)$$

$$[\![\Gamma \vdash (\texttt{case}\ M\ \texttt{of}\ \texttt{inj}_1(x).N_1\ \texttt{or}\ \texttt{inj}_2(x).N_2) : t]\!]\rho \simeq \\ \begin{cases} [\![\Gamma, x : s_i \vdash N_i : t]\!]\rho[x \mapsto d] & \text{if } [\![\Gamma \vdash M : (s_1 \oplus s_2)]\!]\rho = \mathit{inj}_i(d) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Here, the notation $\rho[x \mapsto d]$ denotes the environment in which the x component is extended (or overwritten) to d. For notational convenience, if $\emptyset \vdash M : s$, we write $[\![M]\!]$ for the corresponding element $[\![\emptyset \vdash M : s]\!]\top \in [\![s]\!]$.

The model is adequate:

Theorem 4.1 (Adequacy) *Suppose M is a closed FPC term of type s. Then $M \Downarrow V$ iff $[\![M]\!]\downarrow$.*

The proof appears in the Appendix.

Corollary 4.2 *For closed terms M and N of type s,*

$$\textit{if } [\![M]\!] \lesssim [\![N]\!], \textit{ then } M \sqsubseteq_{FPC} N.$$

Proof: Suppose $[\![M]\!] \lesssim [\![N]\!]$. To show $M \sqsubseteq_{FPC} N$, suppose $C[\cdot]$ is a context such that $C[M] \Downarrow$. By the Adequacy Theorem 4.1, $[\![C[M]]\!]\downarrow$. By the hypothesis, it follows that $[\![C[N]]\!]\downarrow$ and thus $C[N] \Downarrow$. ■

5 Examples

Even though the semantic category $\mathcal{RCPO}$ is arguably complicated, it does support simple reasoning about definability of FPC terms. This section gives two examples of non-definability proofs, with non-sequential functions drawn from the literature.

5.1 Parallel convergence testing is not definable

Consider the partial continuous function

$$f : (\mathit{unit} \Rightarrow \mathit{unit}) \otimes (\mathit{unit} \Rightarrow \mathit{unit}) \to \mathit{unit}$$

defined

$$f\langle g, h\rangle = \begin{cases} \top & \text{if } g(\top)\!\downarrow \text{ or } h(\top)\!\downarrow \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The function f appears to need to do its calculation "in parallel." We would like to prove that f is *not* a morphism in the category, which will immediately imply that f is not definable.

The argument follows the proof of Sieber that *por* is not definable (also due to Plotkin, see (Astesiano and Costa, 1980)). In this case, we need to exhibit an index category $\mathcal{C}$, a choice of $\mathcal{C}$-termination theory T, and a $\mathcal{C}, T$-path theory S. Pick $\mathcal{C}$ to be the category with just one index set $w = \{1, 2, 3\}$ and the identity map, T^w to be the theory with just one implication $(1, 2 \vdash 3)$, and S^w to be the set $\{\, \{\, \}, \{w'\} \mid w' \text{ is a } T^w\text{-model}\,\}$. Let $h : \mathit{unit} \to \mathit{unit}$ be the identity function, and $h' : \mathit{unit} \to \mathit{unit}$ be the empty partial function. Also, let

$$g_1 = \langle h, h'\rangle, \qquad g_2 = \langle h', h\rangle, \qquad g_3 = \langle h', h'\rangle.$$

Returning to standard tuple notation for relations, we claim

$$(g_1, g_2, g_3) \in ((\mathit{unit} \Rightarrow \mathit{unit}) \otimes (\mathit{unit} \Rightarrow \mathit{unit}))(T, S)^w.$$

To prove the claim, we must show that $h_1 = (h, h', h')$ and $h_2 = (h', h, h')$ are both in $(\mathit{unit} \Rightarrow \mathit{unit})(T, S)^w$; this will show that $(g_1, g_2, g_3) = \langle h_1, h_2\rangle$ is in $((\mathit{unit} \Rightarrow \mathit{unit}) \otimes (\mathit{unit} \Rightarrow \mathit{unit}))(T, S)^w$.

We show the claim for the case of h_1 and leave the analogous case of h_2 to the reader. Pick any $u \in \mathit{unit}(T, S)^w$, and let $u' = (\underline{\lambda} d \in w.\, h_1(d)(u(d)))$. We must show $u' \in \mathit{unit}(T, S)^w_S$. By choice of T^w, the domain of u is either $\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}, \{2, 3\}$, or $\{1, 2, 3\}$. We verify the fact for the cases $\{\}$ and $\{1, 2, 3\}$. The remaining cases are very similar and omitted here. First, consider the case of $\emptyset$. By definition, the domain of u' is $\emptyset$ as well and the existence of the path set $\{\}$ justifies $u' \in \mathit{unit}(T, S)^w_S$. For the case of $\mathit{dom}(u) = \{1, 2, 3\}$ we get $\mathit{dom}(u') = \{1\}$ since $h_1 = (h, h', h')$. Thus, via the path set $\{\{1\}\}$, we conclude that $u' \in \mathit{unit}(T, S)^w_S$.

However, since $f(g_1) = \top$, $f(g_2) = \top$, and $f(g_3)$ is undefined,

$$(f(g_1), f(g_2), f(g_3)) \notin \mathit{unit}(T, S)^w_S,$$

so f is not uniform. Thus, f cannot be a morphism, and hence it cannot be definable.

5.2 Sieber's example is not definable

The second example is also due to Sieber (1992), a modified example due to Curien (1986). Let *bool* be the object $(\mathit{unit} \oplus \mathit{unit})$, and

$$A = (\mathit{unit} \Rightarrow \mathit{bool}) \otimes (\mathit{unit} \Rightarrow \mathit{bool}).$$

Let *true* denote $\mathit{inj}_1(\top) \in \mathit{bool}$ and *false* denote $\mathit{inj}_2(\top) \in \mathit{bool}$. Consider the morphisms $g_1, g_2, g_3, g_4 : A \to \mathit{unit}$ defined by

$$\begin{array}{rcl}
g_1\langle h_1, h_2\rangle & \simeq & \begin{cases} \top & \text{if } h_2(\top) = \mathit{true} \\ \top & \text{if } h_1(\top) = \mathit{true} \text{ and} \\ & \quad h_2(\top) = \mathit{false} \\ \text{undefined} & \text{otherwise} \end{cases} \\
g_2\langle h_1, h_2\rangle & \simeq & \begin{cases} \top & \text{if } h_1(\top) = \mathit{false} \\ \text{undefined} & \text{otherwise} \end{cases} \\
g_3\langle h_1, h_2\rangle & \simeq & \begin{cases} \top & \text{if } h_2(\top) = \mathit{false} \\ \text{undefined} & \text{otherwise} \end{cases} \\
g_4\langle h_1, h_2\rangle & \simeq & \text{undefined}
\end{array}$$

We claim that any $f : (A \Rightarrow \mathit{unit}) \to \mathit{unit}$ satisfying

$$f(g_1) = \top, \quad f(g_2) = \top, \quad f(g_3) = \top, \quad f(g_4)\uparrow$$

is not definable in FPC.

The proof of nondefinability requires a nontrivial use of path sets. Pick $\mathcal{C}$ to be the trivial index category $\mathcal{C}$ with object $w = \{1, 2, 3, 4\}$. Let T^w be the termination theory T^w with just one implication $(1, 2, 3 \vdash 4)$. Let S be the set of path sets $\{w_1, \ldots, w_n\}$ such that

- If $1 \in w_i$ and $2 \in w_j$, then $i = j$; and
- If $1 \in w_i$ and $3 \in w_j$, then $i = j$.

It is not hard to prove that this is a $\mathcal{C}, T$-path theory, and that

$$(g_1, g_2, g_3, g_4) \in (A \Rightarrow \mathit{unit})(T, S)^w.$$

However, it is evident that

$$(f(g_1), f(g_2), f(g_3), f(g_4)) \notin \mathit{unit}(T, S)^w_S$$

because of the conditions on the path sets. Thus, there is no such definable f.

6 Full Abstraction

We prove full abstraction for FPC in two steps. Let Finite FPC be the sublanguage of FPC with no recursive types. First, we reduce the full abstraction problem to the problem of showing that all elements in Finite FPC types are definable. The reduction uses techniques from (Riecke and Subrahmanyam, 1997), and we give the basic outline of the proof. Second, we prove that all elements in Finite FPC types are definable by terms. The proof follows the structure of the proof of full abstraction for PCF (O'Hearn and Riecke, 1995).

6.1 Reduction to definability for Finite FPC

Define the unwinding of all type recursions to depth n by

$$\begin{array}{rcl@{\qquad\qquad}rcl} \alpha^n & = & \alpha & (s \oplus t)^n & = & (s^n \oplus t^n) \\ \mathtt{void}^n & = & \mathtt{void} & (s \otimes t)^n & = & (s^n \otimes t^n) \\ \mathtt{unit}^n & = & \mathtt{unit} & (s \to t)^n & = & (s^n \to t^n) \end{array}$$

$$(\mathtt{rec}\,\alpha.\,t)^n = \underbrace{t^n[t^n[\ldots[t^n}_{n}[\mathtt{void}/\alpha]/\alpha]\ldots]/\alpha]/\alpha$$

In Appendix B, we construct terms $\mathtt{clm}^p_{s,n} : s \to s^n$ and $\mathtt{clm}^e_{s,n} : s^n \to s$ that, intuitively, denote a sequence of "colimiting" maps. The terms satisfy the following property:

Lemma 6.1 $\bigsqcup_{n\geq 0}([\![\mathtt{clm}^p_{s,n}]\!];[\![\mathtt{clm}^e_{s,n}]\!]) = id_{[\![s]\!]}$.

The construction uses the techniques in (Riecke and Subrahmanyam, 1997), although we repeat the salient details in the Appendix for completeness. These terms allow us to reduce the problem to definability for Finite FPC.

Theorem 6.2 *Suppose for all Finite FPC types s, all elements of $[\![s]\!]$ are definable by closed terms. Let M, N be closed FPC terms of type s. If $[\![M]\!] \not\sqsubseteq [\![N]\!]$, then $M \not\sqsubseteq_{FPC} N$.*

Proof: We claim that it is enough to prove the theorem for all Finite FPC types s. To see this, suppose s is a general FPC type. We know by Lemma 6.1 that $[\![\mathtt{clm}^p_{s,n}\ M]\!] \not\sqsubseteq [\![\mathtt{clm}^p_{s,n}\ N]\!]$ for some n. Both of the terms $M' = (\mathtt{clm}^p_{s,n}\ M)$ and $N' = (\mathtt{clm}^p_{s,n}\ N)$ have a Finite FPC type s. Thus, if we can find a context $C[\cdot]$ distinguishing M' from N', then the context $C[\mathtt{clm}^p_{s,n}\ [\cdot]]$ distinguishes M and N.

We now prove the theorem for all Finite FPC types s by induction on s, giving two of the five cases and leaving the others to the reader. In the first case, suppose $s = (t \oplus t')$. There are three subcases:

- $[\![M]\!]\!\downarrow$ and $[\![N]\!]\!\uparrow$. Then, by Adequacy, the context $[\cdot]$ distinguishes M and N.
- $[\![M]\!] = inj_i(d)$ and $[\![N]\!] = inj_j(e)$, where $i \neq j$. By Adequacy, we get that $M \Downarrow (\mathtt{inj}_i\ V)$ and $N \Downarrow (\mathtt{inj}_j\ V')$. Suppose $i = 1$ (the other case is analogous); then the context $(\mathtt{case}\ [\cdot]\ \mathtt{of}\ \mathtt{inj}_1(x).x\ \mathtt{or}\ \mathtt{inj}_2(x).\Omega)$ distinguishes M and N.

- $[\![M]\!] = inj_i(d)$ and $[\![N]\!] = inj_i(e)$. By Adequacy, $M \Downarrow (\texttt{inj}_i\ V)$ and $N \Downarrow (\texttt{inj}_i\ V')$. Note that $V \not\sqsubseteq_{FPC} V'$. Suppose $i = 1$ (the other case is analogous), and the context distinguishing V, V' is $C[\cdot]$. Then the context $(\texttt{case}\ [\cdot]\ \texttt{of}\ \texttt{inj}_1(x).C[x]\ \texttt{or}\ \texttt{inj}_2(x).\Omega)$ distinguishes M and N.

In the second case, suppose $s = (t \to t')$. If $[\![M]\!]\!\downarrow$ and $[\![N]\!]\!\uparrow$, then the context $[\cdot]$ distinguishes M and N. If both are defined, on the other hand, by Adequacy, $M \Downarrow (\lambda x : t.\, M')$ and $N \Downarrow (\lambda x : t.\, N')$. Since $[\![M]\!] \not\sqsubseteq [\![N]\!]$, there must be an argument $d \in [\![t]\!]$ such that $[\![M]\!](d) \not\sqsubseteq [\![N]\!](d)$. By hypothesis, there is a closed Finite FPC term P such that $d = [\![P]\!]$. Thus, $[\![M\ P]\!] \not\sqsubseteq [\![N\ P]\!]$. By induction, there is a context $C[\cdot]$ distinguishing $(M\ P)$ and $(N\ P)$. Thus, the context $C[[\cdot]\ P]$ distinguishes M and N. ■

6.2 Definability of elements of Finite FPC types

To prove that all elements of Finite FPC type are representable by terms, we will consider a particular index category $\mathcal{C}$, particular $\mathcal{C}$-termination theory T, and particular $\mathcal{C}, T$-path theory S. Define the index category to be the set of sets of the form

$$[s_1, \ldots, s_n] = |[\![s_1]\!] \otimes \ldots \otimes [\![s_n]\!]|.$$

where $[\,] = \mathit{unit}$. Morphisms are the projections $[s_1, \ldots, s_{n+k}] \to [s_1, \ldots, s_n]$. For any object $w = [s_1, \ldots, s_n]$ of $\mathcal{C}$, let $s = (s_1 \otimes \ldots \otimes s_n)$ and

$$X^w = \{\, w' \subseteq w \mid \text{there is a closed } M : (s \Rightarrow \texttt{unit})\ .\ [\![M]\!](d)\!\downarrow \text{ iff } d \in w' \,\}.$$

The path sets in S^w are defined as follows. $\{w_1, \ldots, w_k\}$ is a path set if:

- $w_1, \ldots, w_k \in X^w$; and
- there is an $M : (s \Rightarrow \bar{k})$ such that $[\![M]\!](d) = \mathbf{i}$ iff $d \in w_i$, where

  $$\bar{n} = \underbrace{(\texttt{unit} \oplus \ldots \oplus \texttt{unit})}_{n}$$

  and $\mathbf{1} = inj_1(\top)$, $\mathbf{2} = inj_2(inj_1(\top))$, and so on.

In the following, we use if-then-else and other notation in terms when working with the types $\bar{n}$; these are just shorthand for Finite FPC terms.

Using the alternative characterization of T-models given by Proposition 3.1, it is not hard to establish the following

Lemma 6.3 *The set $X = \{\, X^w \mid w \in Obj(\mathcal{C}) \,\}$ defines a $\mathcal{C}$-termination theory T by taking X^w to be a set of models.*

Proof: Let $\varphi : v \xrightarrow{\mathcal{C}} w$ and

$$\begin{array}{rclcrcl} w & = & [s_1, \ldots, s_n] & \quad & s & = & (s_1 \otimes \ldots \otimes s_n) \\ v & = & [s_1, \ldots, s_n, \ldots, s_{n+k}] & \quad & s' & = & (s_1 \otimes \ldots \otimes s_n \otimes \ldots \otimes s_{n+k}) \end{array}$$

First, we establish that each $X^w \in X$ is a closure system; by Proposition 3.1, it will then be clear that each T^w is a w-theory. Note that $w \in X^w$ (the requisite term is $(\lambda x : s.\, \langle\rangle)$ of type $(s \Rightarrow \mathtt{unit})$). So suppose $w_1, w_2 \in X^w$. Then there exist M_1, M_2 with $[\![M_i]\!](d)\!\downarrow$ iff $d \in w_i$. Consider the term

$$M = (\lambda x : s.\, (M_1\ x); (M_2\ x)).$$

Then it is clear that $[\![M]\!](d)\!\downarrow$ iff $d \in w_1 \cap w_2$. Thus, $(w_1 \cap w_2) \in X^w$.

Second, suppose w' is a T^w-model. Then by definition, there is a term $N : (s \Rightarrow \mathtt{unit})$ such that $[\![N]\!](d)\!\downarrow$ iff $d \in w'$. Let $v' = \varphi^{-1}(w')$, and

$$M = (\lambda x : s'.\, N\ \langle x_1, \ldots, x_n\rangle)$$

where x_i is shorthand for the i^{th} projection of the term x. Then $[\![M]\!](d)\downarrow$ iff $d \in v'$, so v' is a T^v-model as required. We conclude that T is indeed a $\mathcal{C}$-termination theory. ■

Lemma 6.4 *S is a $\mathcal{C}, T$-path theory.*

Proof: We need to show S^w satisfies four properties. Suppose w' is a T^w-model and $X = \{w_1, \ldots, w_k\}$ and $X' = \{w'_1, \ldots, w'_l\}$ are path sets in S^w. Then there exists a term M defining w', and terms N and P defining these path sets.

- The term $(\lambda x : s.\, \Omega)$ defines the path set $\{\}$.
- The term $(\lambda x : s.\, (M\ x); \mathbf{1})$ defines the path set $\{w'\}$.
- If $X \sqcap X' = \{w_{i_1} \cap w_{j_1}, \ldots, w_{i_n} \cap w_{j_n}\}$, then the term

  $$\begin{array}{l}(\lambda x : s.\, \mathtt{if}\ ((N\ x) = i_1) \wedge ((P\ x) = j_1)\ \mathtt{then}\ \mathbf{1}\ \mathtt{else}\ \ldots \\ \qquad\qquad \mathtt{if}\ ((N\ x) = i_n) \wedge ((P\ x) = j_n)\ \mathtt{then}\ (i_n * j_n)\ \mathtt{else}\ \Omega)\end{array}$$

  defines the path set $X \sqcap X'$.
- Suppose there is a j such that for all i, $w'_i \subseteq w_j$. Then the term

  $$\begin{array}{l}(\lambda x : s.\, \mathtt{if}\ (N\ x) < j\ \mathtt{then}\ (N\ x)\ \mathtt{else} \\ \qquad\qquad \mathtt{if}\ (N\ x) > j\ \mathtt{then}(N\ x) + l - 1\ \mathtt{else}\ (P\ x) + j - 1)\end{array}$$

  defines the path set $\{w_1, \ldots, w_{j-1}, w'_1, \ldots, w'_l, w_{j+1}, \ldots, w_k\}$.

This completes the proof. ■

The next lemma is the main one needed for full abstraction. It shows that every element of the computational relations is represented by a term in Finite FPC.

Lemma 6.5 *Suppose $w = [s_1, \ldots, s_n]$ and $s = (s_1 \otimes \ldots \otimes s_n)$. Then $g \in [\![t]\!](T, S)^w_S$ iff there exists a closed, Finite FPC term $M : (s \Rightarrow t)$ such that $g = [\![M]\!]$.*

Proof: By induction on t.

- $t = \texttt{void}$. If $g \in [\![t]\!](T,S)^w_S$, then the term $M = (\lambda x : s.\, \Omega)$ defines g. The converse is also trivial.

- $t = \texttt{unit}$. For ($\Leftarrow$), suppose $M : (s \Rightarrow \texttt{unit})$. By definition, M defines a T^w-model w' via

$$[\![M]\!](d)\!\downarrow \text{ iff } d \in w'.$$

  Thus, $[\![M]\!] \in [\![\texttt{unit}]\!](T,S)^w_S$. For ($\Rightarrow$), suppose $g \in [\![\texttt{unit}]\!](T,S)^w_S$. Thus, there exists a path set $\{w_1, \ldots, w_k\}$ such that

  - $g(d)\!\downarrow$ iff $d \in w_i$ for some i; and
  - $(\underline{\lambda} d \in w_i.\, g(d)) \in [\![\texttt{unit}]\!](T,S)^w$.

  (Of course, in this particular case of `unit`, the second clause is redundant.) By the definition of path sets, there is a $P : (s \Rightarrow \bar{k})$ such that $[\![P]\!](d) = \mathbf{i}$ iff $d \in w_i$, and undefined otherwise. Let

$$M = (\lambda x : s.\, (P\ x); \langle\rangle).$$

  It is not hard to see that $[\![M]\!] = g$ as needed.

- $t = (t_1 \oplus t_2)$. For ($\Leftarrow$), suppose $M : (s \Rightarrow (t_1 \oplus t_2))$. Consider the following two terms:

$$\begin{array}{rcl} N_1 & = & (\lambda x : s.\, \texttt{case}\ (M\ x)\ \texttt{of}\ \texttt{inj}_1(y).y\ \texttt{or}\ \texttt{inj}_2(y).\Omega) : (s \Rightarrow t_1) \\ N_2 & = & (\lambda x : s.\, \texttt{case}\ (M\ x)\ \texttt{of}\ \texttt{inj}_1(y).\Omega\ \texttt{or}\ \texttt{inj}_2(y).y) : (s \Rightarrow t_2) \end{array}$$

  By induction ($\Leftarrow$), $[\![N_i]\!] \in [\![t_i]\!](T,S)^w_S$. Thus, we have for some path sets $X = \{w_1, \ldots, w_k\}$ and $X' = \{w'_1, \ldots, w'_l\}$,

  - $[\![N_1]\!](d)\!\downarrow$ iff $d \in w_i$ for some i;
  - $(\underline{\lambda} d \in w_i.\, [\![N_1]\!](d)) \in [\![t_1]\!](T,S)^w$ for all i;
  - $[\![N_2]\!](d)\!\downarrow$ iff $d \in w'_i$ for some i; and
  - $(\underline{\lambda} d \in w'_i.\, [\![N_2]\!](d)) \in [\![t_2]\!](T,S)^w$ for all i.

  Note that for any i, j, w_i and w'_j are disjoint: if $d \in w_i$, then $[\![N_1]\!](d)\!\downarrow$, so $[\![M]\!](d) = \mathit{inj}_1(a)$, so $[\![N_2]\!](d)\!\uparrow$, so $d \notin w'_j$. Note also that the set $\{w_1, \ldots, w_k, w'_1, \ldots, w'_l\}$ is a legal path set via the term

$$(\lambda x : s.\, \texttt{case}\ (M\ x)\ \texttt{of}\ \texttt{inj}_1(y).(P_1\ x)\ \texttt{or}\ \texttt{inj}_2(y).(P_2\ x) + k)$$

  if P_1 is the term for X and P_2 is the term for X'. Finally, note that

  - $[\![M]\!](d)\!\downarrow$ iff $d \in w_i$ or w'_i for some i; and

– For the first set of T^w-models,

$$(\underline{\lambda} d \in w_i.\, [\![M]\!](d)) = ((\underline{\lambda} d \in w_i.\, [\![N_1]\!](d)); \mathit{inj}_1) \in [\![t_1 \oplus t_2]\!](T,S)^w$$

and for the second set,

$$(\underline{\lambda} d \in w'_i.\, [\![M]\!](d)) = ((\underline{\lambda} d \in w'_i.\, [\![N_2]\!](d)); \mathit{inj}_2) \in [\![t_1 \oplus t_2]\!](T,S)^w.$$

Therefore, $[\![M]\!] \in [\![t]\!](T,S)^w_S$.

For ($\Rightarrow$), suppose $g \in [\![t_1 \oplus t_2]\!](T,S)^w_S$. By definition, there exists a path set $\{w_1, \dots, w_k\}$ such that

– $g(d)\downarrow$ iff $d \in w_i$ for some i; and
– $g_i = (\underline{\lambda} d \in w_i.\, g(d)) \in [\![t_1 \oplus t_2]\!](T,S)^w$ for all i.

Then by definition, $g_i = (h_i; \mathit{inj}_{j_i})$ for some $h_i \in [\![t_{j_i}]\!](T,S)^w$. Since $\{w_i\}$ itself is a path set we get that $h_i \in [\![t_{j_i}]\!](T,S)^w_S$, and thus by induction ($\Rightarrow$), there exists an $N_i : (s \Rightarrow t_{j_i})$ such that

$$d \in w_i \text{ iff } [\![N_i]\!](d)\downarrow, \text{ and } h_i = (\underline{\lambda} d \in w_i.\, [\![N_i]\!](d)).$$

Since $\{w_1, \dots, w_k\}$ is a path set, there is a term $P : s \Rightarrow \bar{k}$ defining that path set. Let $M_i = (\lambda x : s.\, \texttt{inj}_{j_i}\ (N_i\ x))$ and

$$M = (\lambda x : s.\, \texttt{if}\ (P\ x) = 1\ \texttt{then}\ (M_1\ x)\ \texttt{else}\ \dots$$
$$\texttt{if}\ (P\ x) = k\ \texttt{then}\ (M_k\ x)\ \texttt{else}\ \Omega).$$

Then $[\![M]\!] = g$.

- $t = (t_1 \otimes t_2)$. For ($\Leftarrow$), suppose $M : (s \Rightarrow (t_1 \otimes t_2))$. Consider the following two terms:

$$N_i \quad = \quad (\lambda x : s.\, \texttt{proj}_i\ (M\ x)) : (s \Rightarrow t_i).$$

By induction ($\Leftarrow$), $[\![N_i]\!] \in [\![t_i]\!](T,S)^w_S$. That is, there exists some path sets $X = \{w_1, \dots, w_k\}$ and $X' = \{w'_1, \dots, w'_l\}$, such that

– $[\![N_1]\!](d)\downarrow$ iff $d \in w_i$ for some i;
– $(\underline{\lambda} d \in w_i.\, [\![N_1]\!](d)) \in [\![t_1]\!](T,S)^w$ for all i;
– $[\![N_2]\!](d)\downarrow$ iff $d \in w'_i$ for some i; and
– $(\underline{\lambda} d \in w'_i.\, [\![N_2]\!](d)) \in [\![t_2]\!](T,S)^w$ for all i.

Note that $X \sqcap X' = \{w''_1, \dots, w''_m\}$ is a path set, and note that

$$[\![M]\!](d)\downarrow \text{ iff } d \in w''_i \text{ for some } i; \text{ and}$$

$$(\underline{\lambda} d \in w''_i.\, [\![M]\!](d)) = \langle (\underline{\lambda} d \in w''_i.\, [\![N_1]\!](d)), (\underline{\lambda} d \in w''_i.\, [\![N_2]\!](d)) \rangle$$
$$\in [\![t_1 \otimes t_2]\!](T,S)^w.$$

Therefore, $[\![M]\!] \in [\![t]\!](T,S)^w_S$.

For the converse, suppose $g \in [\![t_1 \otimes t_2]\!](T,S)^w_S$. Then there exists a path set $\{w_1, \dots, w_k\}$ such that

  - $g(d)\downarrow$ iff $d \in w_i$ for some i; and
  - $g_i = (\underline{\lambda} d \in w_i.\, g(d)) \in [\![t_1 \otimes t_2]\!](T,S)^w$.

By definition, $g_i = \langle h_{i,1}, h_{i,2}\rangle$ for some $h_{i,j} \in [\![t_j]\!](T,S)^w$. By induction ($\Rightarrow$), there exist $N_{i,j} : (s \Rightarrow t_j)$ such that

$$d \in w_i \text{ iff } [\![N_{i,j}]\!](d)\downarrow, \text{ and } h_{i,j} = (\underline{\lambda} d \in w_i.\, [\![N_{i,j}]\!](d)).$$

Since we have a path set, there is a term $P : s \Rightarrow \bar{k}$ defining that path set. Let $M_i = (\lambda x : s.\, \langle N_{i,1}\ x, N_{i,2}\ x\rangle)$ and

$$M = (\lambda x : s.\, \texttt{if } (P\ x) = 1 \texttt{ then } (M_1\ x) \texttt{ else } \ldots$$
$$\texttt{if } (P\ x) = k \texttt{ then } (M_k\ x) \texttt{ else } \Omega).$$

Then $[\![M]\!] = g$.

- $t = (t_1 \Rightarrow t_2)$. For ($\Leftarrow$), suppose $M : (s \Rightarrow (t_1 \Rightarrow t_2))$. There are two cases: either $[\![M]\!]$ is undefined on every argument, or $[\![M]\!]$ is defined on at least one argument. In the first case, since $\{\}$ is a path set in S^w, it is not hard to see that $[\![M]\!] \in [\![t_1 \Rightarrow t_2]\!](T,S)^w_S$. In the second case, let

$$M' = (\lambda x : s.\, (M\ x); \mathbf{1}).$$

By the definition of S^w, M' defines a path set consisting of one element; call it $\{w'\}$. We use this as our path set.

  1. Obviously, $[\![M]\!](d)\downarrow$ iff $d \in w'$.
  2. To see $(\underline{\lambda} d \in w'.\, [\![M]\!](d)) \in [\![t_1 \Rightarrow t_2]\!](T,S)^w$, suppose $\varphi : v \xrightarrow{e} w$ and $g \in [\![t_1]\!](T,S)^v$, where $v = [s_1, \ldots, s_{n+k}]$. Furthermore, we let $s' = (s_1 \otimes \ldots \otimes s_{n+k})$. Then, by induction ($\Rightarrow$), there is a closed $N : (s' \Rightarrow t_1)$ such that

$$d \in v' \text{ iff } [\![N]\!](d)\downarrow, \text{ and } g = (\underline{\lambda} d \in v'.\, [\![N]\!](d))$$

     where v' is the domain of g. Consider the term

$$P = (\lambda x : s'.\, (M\ \langle x_1, \ldots, x_n\rangle)\ (N\ x)) : (s' \Rightarrow t_2)$$

     By induction ($\Leftarrow$), $[\![P]\!] \in [\![t_2]\!](T,S)^v_S$. But note that

$$[\![P]\!](d) \simeq (\underline{\lambda} d \in w'.\, [\![M]\!](d))\ (\varphi(d))\ (g(d)).$$

     Thus,

$$(\underline{\lambda} d \in v.\, (\underline{\lambda} d \in w'.\, [\![M]\!](d))\ (\varphi(d))\ (g(d))) \in [\![t_2]\!](T,S)^v_S$$

     as required.

For the converse, suppose $g \in [\![t_1 \Rightarrow t_2]\!](T,S)^w_S$. Then there exists a path set $\{w_1, \ldots, w_k\}$ such that

– $g(d)\downarrow$ iff $d \in w_i$ for some i; and
– $g_i = (\underline{\lambda} d \in w_i.\, g(d)) \in [\![t_1 \Rightarrow t_2]\!](T,S)^w$.

Let $v = [s_1, \ldots, s_n, t_1]$, $\varphi : v \to w$, and $s' = (s_1 \otimes \ldots \otimes s_n \otimes t_1)$, and $N = (\lambda x : s'.\, x_{n+1})$. By induction ($\Leftarrow$), $[\![N]\!] \in [\![t_1]\!](T,S)^v_S$. But note that the T^v-model that makes $[\![N]\!]$ converge is all of v. Thus, there is a path set $\{v_1, \ldots, v_l\} \in S^v$ with $(v_1 \cup \ldots \cup v_l) = v$ and

$$h_j = (\underline{\lambda} d \in v_j.\, [\![N]\!](d)) \in [\![t_1]\!](T,S)^v$$

for all $j \in \{1, \ldots, l\}$. Therefore,

$$h_{i,j} = (\underline{\lambda} d \in v.\, g_i\ (\varphi(d))\ (h_j(d))) \in [\![t_2]\!](T,S)^v_S.$$

By induction ($\Rightarrow$), there exist terms $P_{i,j} : (s' \Rightarrow t_2)$ such that $h_{i,j} = [\![P_{i,j}]\!]$. But note that

$$[\![P_{i,j}]\!]\langle d_1, \ldots, d_n, d\rangle \quad \simeq \quad (g_i\ \langle d_1, \ldots, d_n\rangle\ d).$$

Let Q be the term that defines the path set $\{w_1, \ldots, w_k\}$ and Q_j be the term that defines each v_j. Define

$$\begin{aligned} Q'_i = (\lambda x : s.\, \lambda y : t_1.\, &\mathtt{if}\ Q_1\ \langle x,y\rangle\ \mathtt{then}\ P_{i,1}\ \langle x,y\rangle\ \mathtt{else}\ \ldots \\ &\mathtt{if}\ Q_l\ \langle x,y\rangle\ \mathtt{then}\ P_{i,l}\ \langle x,y\rangle\ \mathtt{else}\ \Omega) \end{aligned}$$

and let

$$\begin{aligned} M = (\lambda x : s.\, &\mathtt{if}\ (Q\ x) = 1\ \mathtt{then}\ (Q'_1\ x)\ \mathtt{else}\ \ldots \\ &\mathtt{if}\ (Q\ x) = k\ \mathtt{then}\ (Q'_k\ x)\ \mathtt{else}\ \Omega) \end{aligned}$$

Then it is not hard to see that $[\![M]\!] = g$.

This completes the induction and hence the proof. ■

Corollary 6.6 *All elements of closed Finite FPC types are definable.*

Proof: Suppose s is a closed Finite FPC type, and $g \in [\![s]\!]$. By concreteness,

$$h = (\underline{\lambda} d \in [\,].\, g) \in [\![s]\!](T,S)^{[\,]}_S.$$

By Lemma 6.5, $h = [\![N]\!]$ for some closed term N of type ($\mathtt{unit} \to s$). Then $g = [\![N\ \langle\rangle]\!]$. ■

6.3 Putting it together

We may now prove the main theorem of the paper.

Theorem 6.7 (Full Abstraction) *Suppose M, N are closed terms of type s. Then $[\![M]\!] \precsim [\![N]\!]$ iff $M \sqsubseteq_{FPC} N$.*

Proof: ($\Leftarrow$) is immediate from Theorem 6.2 and Corollary 6.6, and ($\Rightarrow$) is immediate from Corollary 4.2. ■

7 Discussion

We have shown in this paper how to construct a fully abstract model for call-by-value FPC using logical relations, generalizing the approach taken for PCF (O'Hearn and Riecke, 1995; Sieber, 1992). The model supports a simple form of reasoning for showing that certain values are not definable.

There are other ways to build fully abstract models for FPC. For instance, Riecke and Viswanathan give a dcpo-based model for call-by-value FPC (Riecke and Viswanathan, 1993; Riecke and Viswanathan, 1995), using the syntactic methods of Milner (1977). This construction sheds little light into the structure of FPC, except that the model validates least fixpoint reasoning. Games semantics has also been applied to full abstraction for FPC by McCusker, who builds a model of *call-by-name* FPC (McCusker, 1996; McCusker, 1997). The sums in this model are *separated*: applying the injection operations to the meaning of a divergent computation returns a convergent value (on which a case expression can branch). Game semantics has been recently adapted to the call-by-value setting (Abramsky and McCusker, 1997; Honda and Yoshida, 1997). These papers devise models by loosening the restrictions of the original games semantics (Abramsky et al., 1994; Hyland and Ong, 1995; Nickau, 1994) to include strategies that start with the opponent's *answer* rather than a *question.* Intuitively, this means that the value supplied to a call-by-value function is immediately available without interrogation by the player. The basic definitions are quite different from our logical-relations-based model.

Much of the complexity of our model of FPC lies in the use of Kripke relations. On the one hand, since all examples of reasoning in the model seem to require only the "base" relations, it would be interesting to determine when base relations were sufficient. This kind of result might be analogous to Sieber's result that sequentiality relations suffice for proving facts about PCF up to third-order types (Sieber, 1992). On the other hand, recent results of Ralph Loader suggest that one must go beyond base relations to achieve full abstraction. We conjecture that the following problem is undecidable: given a type in Finite FPC, can one decide how many elements there are in the model of that type? If we remained only with the "base" relations, the problem would be *decidable.* The related decision problem for PCF was first pointed out by Jung and Stoughton (1993); see (Jung et al., 1996; O'Hearn and Riecke, 1995) for a further discussion. Loader shows that the decision problem for PCF over the single boolean base type is undecidable (Loader, 1997). We expect that the proof will carry over to Finite FPC.

The model presented here has some ad hoc features that should be examined more closely. The construction of path sets, for instance, might be better expressed in terms of Grothendieck topologies; recent work by Fiore and Simpson (1999) may be useful. Other recent work by Marz has defined a more general category of "sequential domains" that is suitable for call-by-name (Marz, 1998). Call-by-value can be simulated in Marz's framework through a lifting operation. Marz's conditions on path sets are perhaps easier to understand than our definitions, and also do not rely on the intermediate step of termination theories.

Such a simplification here would make the model easier to construct. It would be useful, among all of these definitions, to settle on a single, well-motivated definition.

We have some hope that the relational account can be adapted to extensions of FPC with other kinds of effects other than simple functional branching, such as continuation-based control operations. We also believe that there is a relationship between "single-threading" of state (O'Hearn and Reynolds, 1997; O'Hearn and Tennent, 1995) and sequentiality; it would be interesting to see if our model can be adapted to model a single-threaded global state. One strength of the current model is its clean separation of values and computations. We conjecture that only the definition of "path theories" must change to reflect the new settings.

Other extensions seem more difficult. For instance, we began by trying to find a similar relation-based model for a *linear* type system, but ran into technical difficulties. The extension of FPC with a notion of *local* state, as in Idealized Algol (Reynolds, 1981) or Standard ML (Milner et al., 1997), also seems to be difficult. One interesting, though non-trivial, direction would be to extend the language with parametric polymorphism. A different kind of relations would be needed in this instance to model parametricity.

Acknowledgements: The first author thanks Glynn Winskel, the organizer of the 1995 Full Abstraction Workshop, and the organizers of the Semantics of Computation program at the Newton Institute for Mathematical Sciences, Cambridge, UK, where this work was begun. The second author thanks Bell Laboratories for offering a very stimulating and pleasant environment during his visit in the summer of 1996. We thank Samson Abramsky, Thomas Hildebrandt, Kohei Honda, Leonid Libkin, Rona Machlin, Peter O'Hearn, Riccardo Pucella, Thomas Streicher, and the anonymous referees of LICS 1997 for comments and encouragement. We also thank the tireless efforts of the referees in improving the exposition and in correcting the definitions with colimits.

References

Abadi, M. and Fiore, M. P. (1996). Syntactic considerations on recursive types. In *Proceedings, Eleventh Annual IEEE Symposium on Logic in Computer Science*, pages 242–252.

Abramsky, S. and Jung, A. (1994). Domain theory. In Abramsky, S., Gabbay, D. M., and Maibaum, T. S. E., editors, *Handbook of Logic in Computer Science*, volume 3, pages 1–168. Clarendon Press.

Abramsky, S., Malacaria, P., and Jagadeesan, R. (1994). Full abstraction for PCF (extended abstract). In Hagiya, M. and Mitchell, J., editors, *Theoretical Aspects of Computer Software*, number 789 in Lect. Notes in Computer Sci., pages 1–15. Springer-Verlag.

Abramsky, S. and McCusker, G. (1997). Call-by-value games. In *Computer Science Logic.* Submitted for publication.

Astesiano, E. and Costa, G. (1980). Nondeterminism and fully abstract models. *RAIRO*, 14(4):323–347.

Curien, P.-L. (1986). *Categorical Combinators, Sequential Algorithms and Functional Programming*. John Wiley & Sons.

Curien, P.-L. and Obtułowicz, A. (1989). Partiality, cartesian closedness, and toposes. *Information and Computation*, 80:50–95.

Fiore, M. and Simpson, A. (1999). Lambda definability with sums via Grothendieck logical relations. In *Typed Lambda Calculi and Applications*, *Lect. Notes in Computer Sci.*. Springer-Verlag. To appear.

Girard, J.-Y. (1971). Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In Fenstad, J. E., editor, *Proceedings of the Second Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*, pages 63–92. North-Holland.

Gunter, C. A. (1992). *Semantics of Programming Languages: Structures and Techniques*. MIT Press.

Gunter, C. A. and Scott, D. S. (1990). Semantic domains. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, volume B, pages 633–674. Elsevier.

Honda, K. and Yoshida, N. (1997). Game theoretic analysis of call-by-value computation. In Degano, P., Gorrieri, R., and Marchetti-Spaccamela, A., editors, *Automata, Languages and Programming: 24th International Colloquium*, volume 1256 of *Lect. Notes in Computer Sci.* Springer-Verlag.

Hyland, J. M. E. and Ong, C.-H. L. (1995). Pi-calculus, dialogue games and PCF. In *7th Annual ACM Conference on Functional Programming Languages and Computer Architecture, La Jolla, California*.

Jung, A., Fiore, M., Moggi, E., O'Hearn, P. W., Riecke, J. G., Rosolini, G., and Stark, I. (1996). Domains and denotational semantics: History, accomplishments, and open problems. *Bulletin of the European Association for Theoretical Computer Science*, pages 227–256.

Jung, A. and Stoughton, A. (1993). Studying the fully abstract model of PCF within its continuous function model. In *Typed Lambda Calculi and Applications*, volume 664 of *Lect. Notes in Computer Sci.*, pages 230–244. Springer-Verlag.

Jung, A. and Tiuryn, J. (1993). A new characterization of lambda definability. In *Typed Lambda Calculi and Applications*, volume 664 of *Lect. Notes in Computer Sci.*, pages 245–257. Springer-Verlag.

Loader, R. (1997). Finitary PCF is not decidable. Unpublished manuscript available from `http://www.dcs.ed.ac.uk/~loader`.

McCusker, G. (1996). Games and full abstraction for FPC. In *Proceedings, Eleventh Annual IEEE Symposium on Logic in Computer Science*, pages 174–183.

McCusker, G. (1997). Games and definability for FPC. *Bulletin of Symbolic Logic*, 3(3):347–362.

Marz, M. (1998) A fully abstract model for sequential computation. Technical Report CSR-98-06, Department of Computer Science, University of Birmingham, 1998.

Meyer, A. R. (1988). Semantical paradigms: Notes for an invited lecture, with two appendices by Stavros S. Cosmadakis. In *Proceedings, Third Annual Symposium on Logic in Computer Science*, pages 236–255. IEEE.

Milner, R. (1977). Fully abstract models of the typed lambda calculus. *Theoretical Computer Sci.*, 4:1–22.

Milner, R., Tofte, M., Harper, R., and MacQueen, D. (1997). *The Definition of Standard ML (Revised).* MIT Press.

Moggi, E. (1991). Notions of computation and monads. *Information and Control*, 93:55–92.

Mulmuley, K. (1987). *Full Abstraction and Semantic Equivalence.* ACM Doctoral Dissertation Award 1986. MIT Press.

Nickau, H. (1994). Hereditarily sequential functionals. In Nerode, A. and Matiyasevich, Y., editors, *Proceedings of the Symposium on Logical Foundations of Computer Science: Logic at St. Petersburg*, volume 813 of *Lect. Notes in Computer Sci.* Springer-Verlag.

O'Hearn, P. W. and Reynolds, J. C. (1997). From Algol to polymorphic linear lambda calculus. Unpublished manuscript available from `http://www.dcs.qmw.ac.uk/~ohearn`.

O'Hearn, P. W. and Riecke, J. G. (1995). Kripke logical relations and PCF. *Information and Computation*, 120(1):107–116.

O'Hearn, P. W. and Tennent, R. D. (1995). Parametricity and local variables. *J. ACM*, 42:658–709.

Pitts, A. M. (1996). Relational properties of domains. *Information and Computation*, 127:66–90.

Plotkin, G. D. (1977). LCF considered as a programming language. *Theoretical Computer Sci.*, 5:223–257.

Plotkin, G. D. (1985). (Towards a) logic for computable functions. Unpublished manuscript, CSLI Summer School Notes.

Reynolds, J. C. (1974). Towards a theory of type structure. In *Proceedings Colloque sur la Programmation*, volume 19 of *Lecture Notes in Computer Science*, pages 408–425, Berlin. Springer-Verlag.

Reynolds, J. C. (1981). The essence of Algol. In de Bakker, J. W. and van Vliet, J. C., editors, *Algorithmic Languages*, pages 345–372. North-Holland, Amsterdam.

Riecke, J. G. (1993). Fully abstract translations between functional languages. *Mathematical Structures in Computer Science*, 3:387–415. Preliminary version appears in *Conference Record of the Eighteenth Annual ACM Symposium on Principles of Programming Languages*, pages 245–254, ACM, 1991.

Riecke, J. G. and Subrahmanyam, R. (1997). Semantic orthogonality of type disciplines. Unpublished manuscript available from `http://www.cs.bell-labs.com/who/riecke/`.

Riecke, J. G. and Viswanathan, R. (1993). Full abstraction for call-by-value sequential languages. Unpublished manuscript.

Riecke, J. G. and Viswanathan, R. (1995). Isolating side effects in sequential languages. In *Conference Record of the Twenty-Second Annual ACM Symposium on Principles of Programming Languages*, pages 1–12. ACM.

Sieber, K. (1992). Reasoning about sequential functions via logical relations. In *Applications of Categories in Computer Science*, volume 177 of *London Mathematical Society Lecture Note Series*. Cambridge University Press.

Stoughton, A. (1988). *Fully Abstract Models of Programming Languages*. Research Notes in Theoretical Computer Science. Pitman/Wiley. Revision of Ph.D thesis, Dept. of Computer Science, Univ. Edinburgh, Report No. CST-40-86, 1986.

Wechler, W. (1992). *Universal Algebra for Computer Scientists*. Number 25 in EATCS Monographs in Theoretical Computer Science. Springer-Verlag.

A Proof of Adequacy

To establish the adequacy of the model, we follow the proof techniques of (Meyer, 1988; Plotkin, 1985; Riecke and Subrahmanyam, 1997). There are two main steps. First, we define a notion of an "adequacy relation" between elements of the model and terms, and show that the meanings of terms and the terms themselves are related by any adequacy relation. Second, we construct a particular adequacy relation. The properties of the adequacy relation allow us to conclude that the model is adequate. Since the proof is not novel, we omit many of the details that are easy to check.

A.1 Adequacy relations

Definition A.1 An **adequacy relation** is a family of relations $\leq_s$, indexed by closed types, between elements of $[\![s]\!]$ and closed terms of type s. The following conditions must also hold.

1. $d \leq_{\mathtt{void}} M$ holds vacuously (since there is no $d \in \mathit{void}$).
2. $\top \leq_{\mathtt{unit}} M$ iff $M \Downarrow \langle\rangle$.
3. $(\mathit{inj}_i\ e) \leq_{s_1 \oplus s_2} M$ iff $M \Downarrow (\mathtt{inj}_i\ V)$ and $(e \leq_{s_i} V)$.

4. $\langle e_1, e_2\rangle \leq_{s_1 \otimes s_2} M$ iff $M \Downarrow \langle V_1, V_2\rangle$ and $(e_i \leq_{s_i} V_i)$.

5. $f \leq_{s_1 \Rightarrow s_2} M$ iff $M \Downarrow (\lambda x : s_1.\, N)$, and $e \leq_{s_1} P$ implies $f(e) \lesssim_{s_2} (M\ P)$.

6. $e \leq_{\texttt{rec}\ \alpha.\ s} M$ iff $M \Downarrow (\texttt{intro}\ V)$ and $\mathit{elim}(e) \leq_{s[\texttt{rec}\ \alpha.\ s/\alpha]} V$.

In the clause for function types, $\mathit{exp} \lesssim_s M$ means either exp is undefined, or it is defined and $\mathit{exp} \leq_s M$.

Lemma A.2 *Suppose $\leq$ is an adequacy relation, $d \leq_s M$, $M \Downarrow V$, and $N \Downarrow V$. Then $d \leq_s N$.*

Proof: By a simple case analysis on s. ■

Lemma A.3 *Suppose $\leq$ is an adequacy relation, $e_i \leq_{s_i} M_i$, $\rho = \langle e_1, \ldots, e_n\rangle$, $\Gamma = (x_1 : s_1, \ldots, x_n : s_n)$, and $\Gamma \vdash M : s$. Then*

$$[\![\Gamma \vdash M : s]\!]\rho \lesssim_s M[M_1, \ldots, M_n/x_1, \ldots, x_n].$$

Proof: By induction on the derivation of $\Gamma \vdash M : s$. We give a few of the cases and leave the others to the reader.

1. $\Gamma \vdash x_i : s_i$. Then $[\![\Gamma \vdash x_i : s_i]\!]\rho = \rho(x_i) \leq_{s_i} M_i$ by the hypothesis.

2. $\Gamma \vdash (N\ P) : t$ where $\Gamma \vdash N : (s \to t)$ and $P : s$. Follows easily from the induction hypothesis.

3. $\Gamma \vdash (\lambda x : s.\, N) : (s \to t)$ where $\Gamma, x : s \vdash N : t$. Suppose $e \leq_s P$, and let

$$M' = ((\lambda x : s.\, N)\ P)[M_1, \ldots, M_n/x_1, \ldots, x_n]$$

   By the induction hypothesis,

$$d' = [\![\Gamma, x : s \vdash N : t]\!]\rho[x \mapsto e] \lesssim_t N[M_1, \ldots, M_n, P/x_1, \ldots, x_n, x] = N'.$$

   Thus, either d' is not defined, or it is defined and $d' \leq_t N'$. In the first case, $d' \lesssim_t M'$. In the second case, $N' \Downarrow V$ for some V. Note then that $M' \Downarrow V$. By Lemma A.2, $d' \leq_t M'$. Thus, by the property of adequacy relations,

$$[\![\Gamma \vdash (\lambda x : s.\, N) : (s \to t)]\!]\rho \lesssim_{s \to t} (\lambda x : s.\, N)[M_1, \ldots, M_n/x_1, \ldots, x_n].$$

   as desired.

4. $\Gamma \vdash (\texttt{elim}\ N) : t[\texttt{rec}\ \alpha.\ t/\alpha]$, where $\Gamma \vdash N : (\texttt{rec}\ \alpha.\ t)$. Follows directly from the induction hypothesis and the definition of adequacy relations.

5. $\Gamma \vdash (\texttt{intro}\ N) : (\texttt{rec}\ \alpha.\ t)$, where $\Gamma \vdash N : t[\texttt{rec}\ \alpha.\ t/\alpha]$. By the induction hypothesis,

$$[\![\Gamma \vdash N : t[\texttt{rec}\ \alpha.\ t/\alpha]]\!]\rho \lesssim_{t[\texttt{rec}\ \alpha.\ t/\alpha]} N[M_1, \ldots, M_n/x_1, \ldots, x_n].$$

Suppose the left side is defined (the other case holds easily). Then

$$N[M_1, \dots, M_n/x_1, \dots, x_n] \Downarrow V \text{ for some } V,$$

and hence

$$(\texttt{intro}\, N)[M_1, \dots, M_n/x_1, \dots, x_n] \Downarrow (\texttt{intro}\, V).$$

Recall that $\mathit{elim} \circ \mathit{intro} = \mathit{id}$. Thus, since $N[M_1, \dots, M_n/x_1, \dots, x_n] \Downarrow V$ and $(\texttt{elim}\,(\texttt{intro}\, V)) \Downarrow V$, it follows by Lemma A.2 that

$$\mathit{elim}([\![\Gamma \vdash (\texttt{intro}\, N) : \texttt{rec}\, \alpha.\, t]\!]\rho) \leq_{t[\texttt{rec}\, \alpha.\, t/\alpha]} V.$$

Therefore, by definition of adequacy relations,

$$[\![\Gamma \vdash (\texttt{intro}\, N) : \texttt{rec}\, \alpha.\, t]\!]\rho \lesssim_{\texttt{rec}\, \alpha.\, t} (\texttt{intro}\, N)[M_1, \dots, M_n/x_1, \dots, x_n].$$

as desired.

This completes the induction and hence the proof. ■

Definition A.4 Suppose s is a closed type. A relation R between elements of $[\![s]\!]$ and closed terms of type s is **directed complete** if for any directed set $\{d_i \mid i \in I\}$ such that for all $i \in I$, $d_i\ R\ M$, then $(\sqcup d_i)\ R\ M$.

Definition A.5 Suppose θ is a substitution from type variables to closed types, and R is a map from type variables to directed-complete relations. We say that R is **compatible with** θ if for all α, $R(\alpha)$ is a binary relation between $[\![\theta(\alpha)]\!]$ and closed terms of type $\theta(\alpha)$.

Definition A.6 For all open types s, define the family

$$(\leq_{s,R} \mid R \text{ compatible with } \theta),$$

where $\leq_{s,R}$ is a relation between elements of $[\![\theta(s)]\!]$ and closed terms of type $\theta(s)$, as follows:

1. $d \leq_{\alpha,R} M$ iff $d\ R(\alpha)\ M$.
2. $d \leq_{\texttt{void},R} M$ holds vacuously.
3. $\top \leq_{\texttt{unit},R} M$ iff $M \Downarrow \langle\rangle$.
4. $(\mathit{inj}_i\ e) \leq_{s_1 \oplus s_2,R} M$ iff $M \Downarrow (\texttt{inj}_i\ V)$ and $(e \leq_{s_i,R} V)$.
5. $\langle e_1, e_2\rangle \leq_{s_1 \otimes s_2,R} M$ iff $M \Downarrow \langle V_1, V_2\rangle$ and $(e_i \leq_{s_i,R} V_i)$.
6. $f \leq_{s_1 \Rightarrow s_2,R} M$ iff
$$M \Downarrow (\lambda x : s_1.\, N), \text{ and } e \leq_{s_1,R} P \text{ implies } f(e) \lesssim_{s_2,R} (M\ P).$$

7. If $s = (\texttt{rec}\ \alpha.\ t)$, then $e \leq_{s,R} M$ iff for all $n \geq 0$, $(\mu^{n,p}_{\theta(s)}\ e) \lesssim^s_{n,R} M$, where $\leq^s_{n,R}$ are the relations defined by

$$\begin{aligned} e \leq^s_{0,R} N &\iff \text{true} \\ e \leq^s_{n+1,R} N &\iff N \Downarrow (\texttt{intro}\ V') \text{ and } e \leq_{t,R[\alpha \mapsto \leq^s_{n,R}]} V'. \end{aligned}$$

(Recall that $\mu^n_{\theta(s)}$ are the maps in the colimiting cocone from Section 3.8.)

Lemma A.7 *Suppose R is compatible with θ. Then for any type s, $\leq_{s,R}$ is a directed-complete relation between elements of $[\![\theta(s)]\!]$ and closed terms of type $\theta(s)$.*

Lemma A.8 $\leq_{t,R[\alpha \mapsto \leq_{s,R}]} = \leq_{t[s/\alpha],R}$

Proof: By induction on t. The only difficult case is $t = (\texttt{rec}\ \beta.\ u)$. Let $R' = R[\alpha \mapsto \leq_{s,R}]$ and $t' = t[s/\alpha]$. Note that $\leq^{t'}_{0,R} = \leq^t_{0,R'}$, and $\leq^{t'}_{n,R} = \leq^t_{n,R'}$ implies that $\leq^{t'}_{n+1,R} = \leq^t_{n+1,R'}$ using the induction hypothesis. Thus, for all n, $\leq^{t'}_{n,R} = \leq^t_{n,R'}$, and hence

$$d \leq_{t,R'} M \iff d \leq_{t',R} M$$

as desired. ■

Definition A.9 If s is a closed type and R is a relation between elements of $[\![s]\!]$ and closed terms of type s, we write $exp\ \widetilde{R}\ M$ if either the expression exp is undefined, or it is defined and $exp\ R\ M$.

Lemma A.10 *Suppose $\pi : \eta \to \eta'$ is a ep-pair environment (i.e., mapping type variables to ep-pairs, not just pre-ep-pairs) and for all α,*

$$\begin{array}{lcl} d\ R(\alpha)\ M & \textit{implies} & \pi(\alpha)^e(d)\ \widetilde{R'(\alpha)}\ M \\ d\ R'(\alpha)\ M & \textit{implies} & \pi(\alpha)^p(d)\ \widetilde{R(\alpha)}\ M \end{array}$$

Then

1. *If $d \leq_{s,R} M$, then $([\![s]\!]\pi)^e(d) \leq_{s,R'} M$.*
2. *If $d \leq_{s,R'} M$, then $([\![s]\!]\pi)^p(d) \lesssim_{s,R} M$.*

Proof: By induction on s, proving both claims simultaneously. Most of the cases are straightforward; the only case that is not is when $s = (\texttt{rec}\ \alpha.\ t)$. For this case, we first prove a claim. Let $\leq^s_{n,R}$ and $\leq^s_{n,R'}$ be the relations defined above. We claim that for all n,

- If $e \leq^s_{n,R} N$, then $p^{n,e}_{s,\pi}(e) \leq^s_{n,R'} N$.
- If $e \leq^s_{n,R'} N$, then $p^{n,p}_{s,\pi}(e) \lesssim^s_{n,R} N$.

We prove the claim by induction on n. The basis is easy, so consider the induction case. Let

$$\pi' = \pi[\alpha \mapsto p^n_{s,\pi}] : \eta[\alpha \mapsto T^n_{s,\eta}] \to \eta'[\alpha \mapsto T^n_{s,\eta'}]$$

Since $p^n_{s,\pi}$ is an ep-pair (by a simple induction on the definition of $p^n_{s,\pi}$), π' is an ep-pair environment. To see the second part of the claim, suppose $e \leq^s_{n+1,R'} N$. Then $N \Downarrow (\texttt{intro}\ V)$ and $e \leq_{t,R'[\alpha\mapsto\leq^s_{n,R'}]} V$. By the induction hypothesis,

$$p^{n+1,p}_{s,\pi}(e) = (\llbracket t \rrbracket \pi')^p(e) \lesssim_{t,R[\alpha\mapsto\leq^s_{n,R}]} V$$

and hence $p^{n+1,p}_{s,\pi}(e) \lesssim^s_{n+1,R} N$. The first part holds analogously, completing the proof of the claim.

Using the claim, we can now prove the lemma. We prove only the second part and leave the other to the reader. Suppose $d \leq_{s,R'} M$. Then for all n, $\mu^{n,p}_{s,\eta'}(d) \lesssim^s_{n,R'} M$. By the claim, $p^{n,p}_{s,\pi}(\mu^{n,p}_{s,\eta'}(d)) \lesssim^s_{n,R} M$. By general facts about the colimit, $p^{n,p}_{s,\pi} \circ \mu^{n,p}_{s,\eta'} = \mu^{n,p}_{s,\eta} \circ (\llbracket s \rrbracket \pi)^p$, so for all n,

$$\mu^{n,p}_{s,\eta}\ ((\llbracket s \rrbracket \pi)^p\ (d)) \lesssim^s_{n,R}\ M.$$

Thus, $(\llbracket s \rrbracket \pi)^p\ (d) \lesssim_{s,R} M$ as desired. ■

Lemma A.11 *Suppose $s = (\texttt{rec}\ \alpha.\ t)$ and $\leq^s_{n,R}$ are the relations defined above. If $d \leq^s_{k,R} M$, then $\mu^{k,e}_{s,\eta}(d) \leq_{s,R} M$.*

Proof: First, we claim that the following two statements hold:

- If $d \leq^s_{n,R} M$, then $f^{n,e}_{s,\eta}(d) \leq^s_{n+1,R} M$.
- If $d \leq^s_{n+1,R} M$, then $f^{n,p}_{s,\eta}(d) \lesssim^s_{n,R} M$.

We prove the claim by induction on n. The base case is straightforward. For the induction case, we prove the first statement and leave the second to the reader. Now, assume that $d \leq^s_{n,R} M$ implies $f^{n,e}_{s,\eta}(d) \leq_{n+1,R} M$. Recall that $f^{n+1,e}_{s,\eta} = (\llbracket t \rrbracket (id[\alpha \mapsto f^n_{s,\eta}]))^e$. Therefore, for any β,

$$d\ R[\alpha \mapsto \leq^s_{n,R}](\beta)\ M \text{ implies } (id[\alpha \mapsto f^n_{s,\eta}](\beta))^e(d)\ R[\alpha \mapsto \leq^s_{n+1,R}](\beta)\ M.$$

So suppose $d \leq^s_{n+1,R} M$; then $M \Downarrow (\texttt{intro}\ V)$ and $d \leq_{t,R[\alpha\mapsto\leq_{n,R}]} V$. By Lemma A.10 (1),

$$(\llbracket t \rrbracket (id[\alpha \mapsto f^n_{s,\eta}]))^e(d) \leq_{t,R[\alpha\mapsto\leq_{n+1,R}]} V$$

and so $f^{n+1,e}_{s,\eta}(d) \leq^s_{n+2,R} M$.

We can now prove the lemma. Suppose $d \leq^s_{k,R} M$. We show that for all $n \geq 0$, $\mu^{n,p}_{s,\eta}\ (\mu^{k,e}_{s,\eta}\ (d)) \lesssim^s_{n,R} M$. There are two cases:

- If $n < k$, then

  $$\mu^{n,p}_{s,\eta}\ (\mu^{k,e}_{s,\eta}\ (d)) = f^{n,p}_{s,\eta}\ (f^{n+1,p}_{s,\eta}\ (\ldots f^{k,p}_{s,\eta}\ (d)\ldots)).$$

  By repeated uses of the second part of the claim,

  $$f^{n,p}_{s,\eta}\ (f^{n+1,p}_{s,\eta}\ (\ldots f^{k,p}_{s,\eta}\ (d)\ldots)) \lesssim^{s}_{n,R} M.$$

- If $n \geq k$, we proceed by induction on n. In the basis, $n = k$, and $\mu^{n,p}_{s,\eta}\ (\mu^{k,e}_{s,\eta}\ (d)) = d \leq^{s}_{n,R} M$ by hypothesis. For the induction case, note that

  $$\mu^{n+1,p}_{s,\eta}\ (\mu^{k,e}_{s,\eta}\ (d)) = f^{n,e}_{s,\eta}\ (\mu^{n,p}_{s,\eta}\ (\mu^{k,e}_{s,\eta}\ (d))).$$

  By the induction hypothesis, $\mu^{n,p}_{s,\eta}\ (\mu^{k,e}_{s,\eta}\ (d)) \leq^{s}_{n,R} M$. By the first part of the claim,

  $$f^{n,e}_{s,\eta}\ (\mu^{n,p}_{s,\eta}\ (\mu^{k,e}_{s,\eta}\ (d))) \leq^{s}_{n+1,R} M$$

  as desired.

This completes the proof. ■

Lemma A.12 *Let R be compatible with θ. Then the family of relations $\leq_{s,R}$ for all closed types s forms an adequacy relation.*

Proof: The only tricky part then is to verify condition (6) of the definition of adequacy relations. Suppose $s = (\texttt{rec}\ \alpha.\ t)$ is a closed type, and pick any η. To see one direction of (6), suppose $d\ \leq_{s,R}\ M$. Then for all n,

$$d_n = \mu^{n,p}_{s,\eta}\ (d) \lesssim^{s}_{n,R} M.$$

In particular, $M \Downarrow (\texttt{intro}\ V)$ and for all n,

$$d_{n+1} \lesssim_{t,R[\alpha \mapsto \leq^{s}_{n,R}]} V.$$

By Lemmas A.10 and A.11,

$$([\![t]\!](id[\alpha \mapsto \mu^{n}_{s,\eta}]))^{e}(d_{n+1}) \lesssim_{t,R[\alpha \mapsto \leq_{s,R}]} V.$$

But note that $([\![t]\!](id[\alpha \mapsto \mu^{n}_{s,\eta}]))^{e} = \mathit{elim} \circ \mu^{n+1,e}_{s,\eta}$. Thus, for all n,

$$\mathit{elim}(\mu^{n+1,e}_{s,\eta}(d_{n+1})) \lesssim_{t,R[\alpha \mapsto \leq_{s,R}]} V$$

and hence by Lemma A.7,

$$\bigsqcup_{n \geq 0} \mathit{elim}(\mu^{n+1,e}_{s,\eta}(d_{n+1})) = \mathit{elim}(d) \lesssim_{t,R[\alpha \mapsto \leq_{s,R}]} V$$

as desired. The converse holds by a similar argument. ■

The Adequacy Theorem 4.1 now follows directly from Lemmas A.3 and A.12.

B Proof of Lemma 6.1

The proof of Lemma 6.1 uses techniques from (Riecke and Subrahmanyam, 1997). In outline, we show how to construct, from the syntax of the language, pre-ep-pairs which mimic the semantic pre-ep-pairs in the construction of the model. We use these to construct the desired terms $\mathtt{clm}^e_{s,n}$ and $\mathtt{clm}^p_{s,n}$.

Given type expressions s and s', a pair of closed terms

$$M : s \Rightarrow s' \text{ and } M' : s' \Rightarrow s$$

is called a **syntactic pre-ep-pair** from s to s' (written $\langle M, M'\rangle : s \hookrightarrow s'$). Some notation on syntactic pre-ep-pairs will be helpful. If P is a syntactic pre-ep-pair, P^e and P^p represent its first and second components respectively, and P^{-1} denotes the pair $\langle P^p, P^e\rangle$. There are also several constructions for building complex syntactic pre-ep-pairs from simple ones. Let $M : s \hookrightarrow t$, $N : t \hookrightarrow u$ and $P : s' \hookrightarrow t'$. Define

$$\begin{gathered}
\mathtt{id}_s : s \hookrightarrow s = \langle \lambda x : s.\, x, \lambda x : s.\, x\rangle \qquad !_t : \mathtt{void} \hookrightarrow t = \langle \Omega, \Omega\rangle \\
\mathtt{iso}_{\mathtt{rec}\ \alpha.\ t} : t[\mathtt{rec}\ \alpha.\ t/\alpha] \hookrightarrow \mathtt{rec}\ \alpha.\ t = \langle \mathtt{intro}_{\mathtt{rec}\ \alpha.\ t}, \mathtt{elim}_{\mathtt{rec}\ \alpha.\ t}\rangle \\
\mathtt{iso}^{-1}_{\mathtt{rec}\ \alpha.\ t} : (\mathtt{rec}\ \alpha.\ t) \hookrightarrow t[\mathtt{rec}\ \alpha.\ t/\alpha] = \langle \mathtt{elim}_{\mathtt{rec}\ \alpha.\ t}, \mathtt{intro}_{\mathtt{rec}\ \alpha.\ t}\rangle \\
(N \circ M) : s \hookrightarrow u = \langle N^e \circ M^e, M^p \circ N^p\rangle
\end{gathered}$$

$$\begin{aligned}
&(M \oplus P) : (s \oplus s') \hookrightarrow (t \oplus t') = \\
&\qquad\qquad \langle \lambda x : s \oplus s'.\ \mathtt{case}\ x\ \mathtt{of}\ \mathtt{inj}_1(y).(M^e\ y)\ \mathtt{or}\ \mathtt{inj}_2(y).(P^e\ y), \\
&\qquad\qquad\ \lambda x : t \oplus t'.\ \mathtt{case}\ x\ \mathtt{of}\ \mathtt{inj}_1(y).(M^p\ y)\ \mathtt{or}\ \mathtt{inj}_2(y).(P^p\ y)\rangle \\
&(M \otimes P) : (s \otimes s') \hookrightarrow (t \otimes t') = \ \langle \lambda x : s \otimes s'.\ \langle M^e\ (\mathtt{proj}_1\ x), P^e\ (\mathtt{proj}_2\ x)\rangle, \\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \lambda x : t \otimes t'.\ \langle M^p\ (\mathtt{proj}_1\ x), P^p\ (\mathtt{proj}_2\ x)\rangle\rangle \\
&(M \Rightarrow P) : (s \Rightarrow s') \hookrightarrow (t \Rightarrow t') = \\
&\qquad\qquad\qquad \langle \lambda x : s \Rightarrow s'.\ P^e \circ x \circ M^p, \lambda x : t \Rightarrow t'.\ P^p \circ x \circ M^e\rangle
\end{aligned}$$

where, abusing notation, $(Q \circ Q')$ stands for the term $(\lambda x.\ Q\ (Q'\ x))$.

We use these constructions in defining terms $\langle \mathtt{F}^e_s, \mathtt{F}^p_s\rangle$ that syntactically represent the action, on *syntactic* pre-ep-pairs, of the functor $[\![s]\!]$ on *semantic* pre-ep-pairs. The terms are defined by induction on the structure of s. As with the functors, the syntactic terms take in a map, usually denoted $\mathtt{pi}$, from type variables to syntactic pre-ep-pairs. These maps go between **type substitutions** θ, i.e., maps from type variables to closed types. More precisely, we write $\mathtt{pi} : \theta \hookrightarrow \theta'$ if θ, θ' are type substitutions such that for all α, $\mathtt{pi}(\alpha) : \theta(\alpha) \hookrightarrow \theta'(\alpha)$. We write $\overline{\mathtt{id}} : \theta \hookrightarrow \theta$ to denote the map where for all type variables α, $\overline{\mathtt{id}}(\alpha) = \mathtt{id}_{\theta(\alpha)}$.

To define the syntactic representation of the functor, let s be a type, and $\mathtt{pi} : \theta \hookrightarrow \theta'$. Let $\theta(\tau)$ denote the type produced from applying the type substitution θ to τ. The term $\mathtt{F}_s(\mathtt{pi}) : \theta(s) \hookrightarrow \theta'(s)$ is a syntactic pre-ep-pair defined by induction on s:

$$\begin{array}{rcl}
\mathtt{F}_\alpha(\mathtt{pi}) & = & \mathtt{pi}(\alpha) \\
\mathtt{F}_{\mathtt{unit}}(\mathtt{pi}) & = & \mathtt{id}_{\mathtt{unit}} \\
\mathtt{F}_{s\Rightarrow t}(\mathtt{pi}) & = & \mathtt{F}_s(\mathtt{pi}) \;\Rightarrow\; \mathtt{F}_t(\mathtt{pi}) \\
\mathtt{F}_{\mathtt{rec}\ \alpha.\ t}(\mathtt{pi}) & = & \mathtt{fix}\ (\lambda f.\ \mathtt{iso}_{\theta'(\mathtt{rec}\ \alpha.\ t)} \circ \mathtt{F}_t(\mathtt{pi}[\alpha \mapsto f]) \circ \mathtt{iso}^{-1}_{\theta(\mathtt{rec}\ \alpha.\ t)})
\end{array}$$

A similar definition appears elsewhere (Abadi and Fiore, 1996). In the definition, the term `fix` in the last line is shorthand for the term $\lambda g.\ \Delta\ (\mathtt{intro}\ \Delta)$, where

$$\Delta = \lambda x.\ \langle \lambda y.\ (\mathtt{proj}_1\ (g\ ((\mathtt{elim}\, x)\ x)))\ y, \lambda z.\ (\mathtt{proj}_2\ (g\ ((\mathtt{elim}\, x)\ x)))\ z\rangle.$$

The term `fix` is a simple modification of the call-by-value recursion combinator (see (Gunter, 1992)); it defines a pair $\langle \mathtt{F}^e_{\mathtt{rec}\ \alpha.\ t}(\mathtt{pi}), \mathtt{F}^p_{\mathtt{rec}\ \alpha.\ t}(\mathtt{pi})\rangle$ by mutual recursion.

We will show that the semantic functor $[\![s]\!]$ coincides with the syntactic functor $\mathtt{F}_s$. In order to do this, we must have a way to connect semantic pre-ep-environments π with syntactic pre-ep-environments `pi`. Suppose π is a morphism in $\mathcal{E}$ (see previous subsection) and η is a type environment. Then π **matches** $\mathtt{pi} : \theta \hookrightarrow \theta'$ if for all type variables α, $[\![\mathtt{pi}(\alpha)]\!] = \pi(\alpha)$.

Suppose $\mathtt{pi} : \theta \hookrightarrow \theta'$ and $s = (\mathtt{rec}\ \alpha.\ t)$. For any $\mathtt{g} : \theta(s) \hookrightarrow \theta'(s)$, define $\mathtt{J}_s(\mathtt{pi}, \mathtt{g}) : \theta(s) \hookrightarrow \theta'(s)$ by

$$\mathtt{J}_s(\mathtt{pi}, \mathtt{g}) = \mathtt{iso}_{\theta'(s)} \circ \mathtt{F}_t(\mathtt{pi}[\alpha \mapsto \mathtt{g}]) \circ \mathtt{iso}^{-1}_{\theta(s)}.$$

Define

$$\begin{array}{rcl}
\mathtt{J}^0_s(\mathtt{pi}, \mathtt{g}) & = & \mathtt{g} \\
\mathtt{J}^{n+1}_s(\mathtt{pi}, \mathtt{g}) & = & \mathtt{J}_s(\mathtt{pi}, \mathtt{J}^n_s(\mathtt{pi}, \mathtt{g}))
\end{array}$$

Lemma B.1 $[\![\mathtt{F}_s(\mathtt{pi})]\!] = \bigsqcup_{n\geq 0}[\![\mathtt{J}^n_s(\mathtt{pi}, \Omega)]\!]$.

Theorem B.2 (Correspondence) *Suppose π matches* $\mathtt{pi} : \theta_1 \hookrightarrow \theta_2$. *Then* $[\![s]\!]\pi = [\![\mathtt{F}_s(\mathtt{pi})]\!]$.

Proof: By induction on the structure of s. We consider four of the cases here.

1. $s = \alpha$. Then $[\![\mathtt{F}_\alpha(\mathtt{pi})]\!] = [\![\mathtt{pi}(\alpha)]\!] = \pi(\alpha) = [\![\alpha]\!](\pi)$ by the hypothesis that π matches `pi`.

2. $s = \mathtt{unit}, \mathtt{void}$. Then $[\![\mathtt{F}_s(\mathtt{pi})]\!] = \mathtt{id}_s = [\![s]\!]\pi$.

3. $s = (s_1 \Rightarrow s_2)$. Then

$$\begin{array}{rcl}
[\![\mathtt{F}_s(\mathtt{pi})]\!] & = & [\![(\mathtt{F}_{s_1}(\mathtt{pi}) \Rightarrow \mathtt{F}_{s_2}(\mathtt{pi}))]\!] \\
& = & ([\![\mathtt{F}_{s_1}(\mathtt{pi})]\!] \Rightarrow [\![\mathtt{F}_{s_2}(\mathtt{pi})]\!]) \\
& = & [\![s_1]\!]\pi \Rightarrow [\![s_2]\!]\pi \\
& = & [\![s]\!]\pi
\end{array}$$

4. $s = (\texttt{rec}\ \alpha.\ t)$. Define type environments η_1 and η_2 by $\eta_1(\alpha) = [\![\theta_1(\alpha)]\!]$ and $\eta_2(\alpha) = [\![\theta_2(\alpha)]\!]$. We claim that $[\![\texttt{F}_s(\texttt{pi})]\!]$ is the unique mediating map from $[\![s]\!]\eta_1 = T_{s,\eta_1}$ to $[\![s]\!]\eta_2 = T_{s,\eta_2}$. The proof of the claim is quite similar to the proof of the existence of Freyd's minimal invariants from (Abramsky and Jung, 1994; Pitts, 1996), but we present the argument in detail because of the slight difference in setting with type environments.

   Recall the definitions of $\mu^n_{s,\eta}$ and $p^n_{s,\pi}$ and the definition of $\texttt{J}^n_s$ above. To prove the claim, we note three facts:

   - For any type environment η, $\mu^{n+1}_{s,\eta} = [\![\texttt{iso}]\!] \circ [\![t]\!](id[\alpha \mapsto \mu^n_{s,\eta}])$.
   - $[\![\texttt{J}^n_s(\texttt{pi},\Omega)]\!]$ is a pair of partial functions.
   - $[\![\texttt{J}^n_s(\texttt{pi},\Omega)]\!] \circ \mu^n_{s,\eta_1} = \mu^n_{s,\eta_2} \circ p^n_{s,\pi}$.

   The first fact holds from general principles about the colimit. The second may be proved by induction on n, using the induction hypothesis of the theorem. The proof of the third proceeds by induction on n. The base case, when $n = 0$, is easy. For the induction step,

$$\begin{aligned}
&[\![\texttt{J}^{n+1}_s(\texttt{pi},\Omega) \circ \mu^{n+1}_{s,\eta_1}]\!] \\
&\quad = [\![\texttt{iso}]\!] \circ [\![\texttt{F}_t(\texttt{pi}[\alpha \mapsto \texttt{J}^n_s(\texttt{pi},\Omega)])]\!] \circ [\![\texttt{iso}^{-1}]\!] \circ \mu^{n+1}_{s,\eta_1} \\
&\quad = [\![\texttt{iso}]\!] \circ [\![t]\!](\pi[\alpha \mapsto [\![\texttt{J}^n_s(\texttt{pi},\Omega)]\!]]) \circ [\![t]\!](id[\alpha \mapsto \mu^n_{s,\eta_1}]) \\
&\quad = [\![\texttt{iso}]\!] \circ [\![t]\!](\pi[\alpha \mapsto [\![\texttt{J}^n_s(\texttt{pi},\Omega)]\!] \circ \mu^n_{s,\eta_1}]) \\
&\quad = [\![\texttt{iso}]\!] \circ [\![t]\!](\pi[\alpha \mapsto \mu^n_{s,\eta_2} \circ p^n_{s,\pi}]) \\
&\quad = [\![\texttt{iso}]\!] \circ [\![t]\!](id[\alpha \mapsto \mu^n_{s,\eta_2}]) \circ [\![t]\!](\pi[\alpha \mapsto p^n_{s,\pi}]) \\
&\quad = \mu^{n+1}_{s,\eta_2} \circ p^{n+1}_{s,\pi}
\end{aligned}$$

   as desired, where the second line follows by the induction hypothesis of the theorem (the fact that the semantic functor $[\![s]\!]$ coincides with $\texttt{F}_s$) , and the fourth line follows by the induction hypothesis.

   These facts can be used to prove the claim. Any mediating map must be unique, since $\langle [\![s]\!]\eta_1, \mu^n_{s,\eta_1} \mid n \geq 0 \rangle$ is a colimiting cocone for the chain $\langle T^i_{s,\eta_1}, f^i_{s,\eta_1} \mid i \geq 0 \rangle$ and $\langle [\![s]\!]\eta_2, \mu^n_{s,\eta_2} \circ p^n_{s,\pi} \mid n \geq 0 \rangle$ is a cocone for the same chain. Thus, all we need to do is to show that $\texttt{F}_s(\texttt{pi})$ is a mediating map, i.e., for all $n \geq 0$,

$$[\![\texttt{F}_s(\texttt{pi})]\!] \circ \mu^n_{s,\eta_1} = \mu^n_{s,\eta_2} \circ p^n_{s,\pi}.$$

   To see this, for any type s and type environment η, define $f^{n\Rightarrow m}_{s,\eta}$ to be the composition of the maps from $T^n_{s,\eta}$ to $T^m_{s,\eta}$; then for any $n \leq m$,

$$\begin{aligned}
[\![\texttt{J}^m_s(\texttt{pi},\Omega)]\!] \circ \mu^n_{s,\eta_1} &= [\![\texttt{J}^m_s(\texttt{pi},\Omega)]\!] \circ \mu^m_{s,\eta_1} \circ f^{n\Rightarrow m}_{s,\eta_1} \\
&= \mu^m_{s,\eta_2} \circ p^m_{s,\pi} \circ f^{n\Rightarrow m}_{s,\eta_1} \\
&= \mu^m_{s,\eta_2} \circ f^{n\Rightarrow m}_{s,\eta_2} \circ p^n_{s,\pi} \\
&= \mu^n_{s,\eta_2} \circ p^n_{s,\pi}
\end{aligned}$$

and hence

$$\begin{aligned}[\![\mathtt{F}_s(\mathtt{pi})]\!] \circ \mu^n_{s,\eta_1} &= (\bigsqcup_{m\geq 0} [\![\mathtt{J}^m_s(\mathtt{pi},\Omega)]\!]) \circ \mu^n_{s,\eta_1} \\ &= \bigsqcup_{m\geq 0} \mu^n_{s,\eta_2} \circ p^n_{s,\pi} = \mu^n_{s,\eta_2} \circ p^n_{s,\pi}\end{aligned}$$

as desired.

This completes the induction and hence the proof. ■

Define two families of syntactic pre-ep-pairs

$$\mathtt{unwnd}_{s,i} : s^i \hookrightarrow s^{i+1} \text{ and } \mathtt{clm}_{s,i} : s^i \hookrightarrow s$$

by induction on s. Most of the definition is straightforward—the main difficulty lies when s is a recursive type.

1. For type variables, $\mathtt{unwnd}_{\alpha,n} = \mathtt{id}_\alpha$ and $\mathtt{clm}_{\alpha,n} = \mathtt{id}_\alpha$.
2. For the base types $s = \mathtt{void}, \mathtt{unit}$, let $\mathtt{unwnd}_{s,n} = \mathtt{id}_s$ and $\mathtt{clm}_{s,n} = \mathtt{id}_s$.
3. For sum, product, and function types, let

$$\begin{aligned}\mathtt{unwnd}_{s_1\Rightarrow s_2,n} &= \mathtt{unwnd}_{s_1,n} \Rightarrow \mathtt{unwnd}_{s_2,n} \\ \mathtt{clm}_{s_1\Rightarrow s_2,n} &= \mathtt{clm}_{s_1,n} \Rightarrow \mathtt{clm}_{s_2,n}\end{aligned}$$

   and similarly for sums and products.

4. For recursive types, suppose $s = (\mathtt{rec}\ \alpha.\ t)$. Define the family of types $T^n_{s,k}$, for $n, k \geq 0$, by

$$\begin{aligned}T^0_{s,k} &= \mathtt{void} \\ T^{n+1}_{s,k} &= t_k[T^n_{s,k}/\alpha].\end{aligned}$$

   Notice that, for any n, $T^n_{s,n}$ is the same as s_n defined above. Define two families of syntactic pre-ep-pairs

$$\mathtt{f}^n_{s,k} : T^n_{s,k} \hookrightarrow T^n_{s,k+1} \text{ and } \mathtt{g}^n_{s,k} : T^n_{s,k} \hookrightarrow T^{n+1}_{s,k}$$

   by

$$\begin{array}{lllllll}\mathtt{f}^0_{s,k} &=& !_{\mathtt{void}} & \mathtt{f}^{n+1}_{s,k} &=& \mathtt{F}_{t_{k+1}}(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{f}^n_{s,k}]) \circ \mathtt{unwnd}_{t,k}[T^n_{s,k}/\alpha] \\ \mathtt{g}^0_{s,k} &=& !_{T^1_{s,k}} & \mathtt{g}^{n+1}_{s,k} &=& \mathtt{F}_{t_k}(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{g}^n_{s,k}])\end{array}$$

A diagram clarifies these types and functions:

$$\begin{tikzcd}
\vdots & \vdots & \vdots & \\
T^2_{s,0} \arrow[u, "\mathtt{g}^0_{s,2}"'] \arrow[r, "\mathtt{f}^2_{s,0}"'] & T^2_{s,1} \arrow[u, "\mathtt{g}^1_{s,2}"'] \arrow[r, "\mathtt{f}^2_{s,1}"'] & T^2_{s,2} \arrow[u, "\mathtt{g}^2_{s,2}"'] \arrow[r, "\mathtt{f}^2_{s,2}"'] & \cdots \\
T^1_{s,0} \arrow[u, "\mathtt{g}^0_{s,1}"'] \arrow[r, "\mathtt{f}^1_{s,0}"'] & T^1_{s,1} \arrow[u, "\mathtt{g}^1_{s,1}"'] \arrow[r, "\mathtt{f}^1_{s,1}"'] & T^1_{s,2} \arrow[u, "\mathtt{g}^2_{s,1}"'] \arrow[r, "\mathtt{f}^1_{s,2}"'] & \cdots \\
\mathtt{void} \arrow[u, "\mathtt{g}^0_{s,0}"'] \arrow[r, "\mathtt{f}^0_{s,0}"'] & \mathtt{void} \arrow[u, "\mathtt{g}^1_{s,0}"'] \arrow[r, "\mathtt{f}^0_{s,1}"'] & \mathtt{void} \arrow[u, "\mathtt{g}^2_{s,0}"'] \arrow[r, "\mathtt{f}^0_{s,2}"'] & \cdots
\end{tikzcd}$$

Then define the maps $\mathtt{unwnd}_{s,n}$ along the main diagonal:

$$\mathtt{unwnd}_{s,n} \quad = \quad \mathtt{f}^{n+1}_{s,n} \circ \mathtt{g}^{n}_{s,n}$$

For the colimiting maps, define $\mathtt{clm}^n_{s,k} : T^n_{s,k} \hookrightarrow s$ by

$$\begin{array}{rcl}
\mathtt{clm}^0_{s,k} & = & !_s \\
\mathtt{clm}^{n+1}_{s,k} & = & \mathtt{iso}_s \circ \mathtt{F}_t(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{clm}^n_{s,k}]) \circ \mathtt{clm}_{t,k}[T^n_{s,k}/\alpha]
\end{array}$$

and set $\mathtt{clm}_{s,n} = \mathtt{clm}^n_{s,n}$.

In order to prove Lemma 6.1, we need to break up the definition of the colimiting maps on recursive types into two pieces. Suppose $s = (\mathtt{rec}\ \alpha.\ t)$. Define the syntactic pre-ep-pairs $\mathtt{h}^n_{s,k} : T^n_{s,k} \hookrightarrow T^n_s$ and $\mathtt{mu}^n_s : T^n_s \hookrightarrow s$, where

$$\begin{array}{rclrcl}
T^0_s & = & \mathtt{void} & T^{n+1}_s & = & t[T^n_s/\alpha] \\
\mathtt{h}^0_{s,k} & = & \mathtt{id}_{\mathtt{void}} & \mathtt{h}^{n+1}_{s,k} & = & \mathtt{F}_t(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}^n_{s,k}]) \circ \mathtt{clm}_{t,k}[T^n_{s,k}/\alpha] \\
\mathtt{mu}^0_s & = & !_s & \mathtt{mu}^{n+1}_s & = & \mathtt{iso}_s \circ \mathtt{F}_t(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{mu}^n_s])
\end{array}$$

Lemma B.3 $[\![\theta(\mathtt{mu}^n_s)]\!] = \mu^n_{\theta(s)}$.

The proof follows from the Correspondence Theorem and the properties of colimits.

Lemma B.4 $[\![\theta(\mathtt{clm}^n_{s,k})]\!] = [\![\mathtt{mu}^n_s \circ \mathtt{h}^n_{s,k}]\!]$.

Proof: By induction on n; to simplify notation, we omit meaning brackets below. The base case is trivial; the induction step is as follows:

$$\begin{aligned}
&\theta(\mathtt{mu}_s^{n+1} \circ \mathtt{h}_{s,k}^{n+1}) \\
&\quad = \theta(\mathtt{iso}_s \circ \mathtt{F}_t(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{mu}_s^n]) \circ \mathtt{F}_t(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^n]) \circ \mathtt{clm}_{t,k}[T_{s,k}^n/\alpha]) \\
&\quad = \theta(\mathtt{iso}_s \circ \mathtt{F}_t(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{mu}_s^n \circ \mathtt{h}_{s,k}^n]) \circ \mathtt{clm}_{t,k}[T_{s,k}^n/\alpha]) \\
&\quad = \theta(\mathtt{iso}_s \circ \mathtt{F}_t(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{clm}_{s,k}^n]) \circ \mathtt{clm}_{t,k}[T_{s,k}^n/\alpha]) \\
&\quad = \theta(\mathtt{clm}_{s,k}^{n+1})
\end{aligned}$$

where the first equality follows from the definition of $\mathtt{mu}_s^{n+1}, \mathtt{h}_{s,k}^{n+1}$, the second from syntactic functoriality, the third from the induction hypothesis, and the forth from the definition of $\mathtt{clm}_{s,k}^{n+1}$. This completes the induction step and hence the proof. ■

Lemma B.5 *For any s and θ, $\bigsqcup_{n\geq 0}[\![\theta(\mathtt{clm}_{s,n}^e \circ \mathtt{clm}_{s,n}^p)]\!] = id_{[\![\theta(s)]\!]}$.*

Proof: We proceed by induction on s, omitting meaning brackets to simplify the notation. We give only the case of $s = (\mathtt{rec}\ \alpha.\ t)$ and leave the others to the reader. We claim

$$\bigsqcup_{k\geq 0} \theta(\mathtt{h}_{s,k}^{n,e} \circ \mathtt{h}_{s,k}^{n,p}) = id.$$

We prove the claim by induction on n. The basis, $n = 0$, is easy, so for the induction step,

$$\begin{aligned}
&\bigsqcup_{k\geq 0} \theta(\mathtt{h}_{s,k}^{n+1,e} \circ \mathtt{h}_{s,k}^{n+1,p}) \\
&= \bigsqcup_{k\geq 0} \theta(\mathtt{F}_t^e(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}]) \circ \mathtt{clm}_{t,k}^e[T_{s,k}^n/\alpha] \circ \mathtt{clm}_{t,k}^p[T_{s,k}^n/\alpha] \circ \mathtt{F}_t^p(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}])) \\
&= \bigsqcup_{k,m\geq 0} \theta(\mathtt{F}_t^e(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}]) \circ \mathtt{clm}_{t,m}^e[T_{s,k}^n/\alpha] \circ \mathtt{clm}_{t,m}^p[T_{s,k}^n/\alpha] \circ \mathtt{F}_t^p(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}])) \\
&= \bigsqcup_{k,m\geq 0} \theta(\mathtt{F}_t^e(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}]) \circ (\mathtt{clm}_{t,m}^e \circ \mathtt{clm}_{t,m}^p)[T_{s,k}^n/\alpha] \circ \mathtt{F}_t^p(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}])) \\
&= \bigsqcup_{k\geq 0} \theta(\mathtt{F}_t^e(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}]) \circ (\bigsqcup_{m\geq 0} \mathtt{clm}_{t,m}^e \circ \mathtt{clm}_{t,m}^p)[T_{s,k}^n/\alpha] \circ \mathtt{F}_t^p(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}])) \\
&= \bigsqcup_{k\geq 0} \theta(\mathtt{F}_t^e(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}]) \circ \mathtt{F}_t^p(\overline{\mathtt{id}}[\alpha \mapsto \mathtt{h}_{s,k}^{n,e}])) \\
&= id
\end{aligned}$$

where the second line follows from Bekič's Lemma, the fifth from the global induction hypothesis, and the last by a simple induction on t, using the local induction hypothesis. Taking advantage of this fact,

$$\begin{aligned}
\bigsqcup_{n\geq 0} \theta(\mathtt{clm}^{e}_{s,n} \circ \mathtt{clm}^{p}_{s,n}) &= \bigsqcup_{n\geq 0} \theta(\mathtt{clm}^{n,e}_{s,n} \circ \mathtt{clm}^{n,p}_{s,n}) \\
&= \bigsqcup_{k,n\geq 0} \theta(\mathtt{clm}^{n,e}_{s,k} \circ \mathtt{clm}^{n,p}_{s,k}) \\
&= \bigsqcup_{k,n\geq 0} \theta(\mathtt{mu}^{n,e}_{s} \circ \mathtt{h}^{n,e}_{s,k} \circ \mathtt{h}^{n,p}_{s,k} \circ \mathtt{mu}^{n,p}_{s}) \\
&= \bigsqcup_{n\geq 0} \theta(\mathtt{mu}^{n,e}_{s} \circ (\bigsqcup_{k\geq 0} \mathtt{h}^{n,e}_{s,k} \circ \mathtt{h}^{n,p}_{s,k}) \circ \mathtt{mu}^{n,p}_{s}) \\
&= \bigsqcup_{n\geq 0} \theta(\mathtt{mu}^{n,e}_{s} \circ \mathtt{mu}^{n,p}_{s}) \\
&= \mathtt{id}_{\theta(s)}
\end{aligned}$$

where the third line follows from Lemma B.4, the fifth from the claim, and the last by Lemma B.3 and the properties of colimits. This completes the induction and hence the proof. ■

Lemma 6.1 now follows directly from Lemma B.5.