Copyright 2001 Springer-Verlag. All rights reserved. This is a draft of an article to appear in "Essays on Programming Methodology", edited by A. McIver and C. Morgan, to be published by Springer-Verlag. Copies must not be offered for sale.

What do Types Mean? — From Intrinsic to Extrinsic Semantics

John C. Reynolds

ABSTRACT A definition of a typed language is said to be "intrinsic" if it assigns meanings to typings rather than arbitrary phrases, so that ill-typed phrases are meaningless. In contrast, a definition is said to be "extrinsic" if all phrases have meanings that are independent of their typings, while typings represent properties of these meanings.

For a simply typed lambda calculus, extended with integers, recursion, and conditional expressions, we give an intrinsic denotational semantics and a denotational semantics of the underlying untyped language. We then establish a logical relations theorem between these two semantics, and show that the logical relations can be "bracketed" by retractions between the domains of the two semantics. From these results, we derive an extrinsic semantics that uses partial equivalence relations.

There are two very different ways of giving denotational semantics to a programming language (or other formal language) with a nontrivial type system. In an *intrinsic* semantics, only phrases that satisfy typing judgements have meanings. Indeed, meanings are assigned to the typing judgements, rather than to the phrases themselves, so that a phrase that satisfies several judgements will have several meanings.

For example, consider $\lambda x.\, x$ (in a simply typed functional language). Corresponding to the typing judgement $\vdash \lambda x.\, x : \mathbf{int} \to \mathbf{int}$, its intrinsic meaning is the identity function on the integers, while corresponding to the judgement $\vdash \lambda x.\, x : \mathbf{bool} \to \mathbf{bool}$, its intrinsic meaning is the identity function on truth values. On the other hand, $\lambda x.\, x x$, which does not satisfy any typing judgement, does not have any intrinsic meaning.

In contrast, in an *extrinsic* semantics, the meaning of each phrase is the same as it would be in a untyped language, regardless of its typing properties. In this view, a typing judgement is an assertion that the meaning of a phrase possesses some property.

For example, the extrinsic meaning of $\lambda x.\, x$ is the identity function on the universe of all values that can occur in a computation. In the simple case where integers and booleans can be regarded as members of this universe, the judgement $\vdash \lambda x.\, x : \mathbf{int} \to \mathbf{int}$ asserts that this function maps each integer into an integer, and the judgement $\vdash \lambda x.\, x : \mathbf{bool} \to \mathbf{bool}$ asserts that the same function maps each truth value into a truth value.

The terms "intrinsic" and "extrinsic" are recent coinages by the author

[1, Chapter 15], but the concepts are much older. The intrinsic view is associated with Alonzo Church, and has been called "ontological" by Leivant [2]. The extrinsic view is associated with Haskell Curry, and has been called "semantical" by Leivant.

In this paper, we will consider the denotational semantics of a very simple typed call-by-name language with integers as a primitive type, functions, conditional expressions, and recursion definitions of values (but not of types). First, we will give an intrinsic semantics and an untyped semantics, which we will relate by a logical relations theorem. Then we will define embedding-retraction pairs between the domain specified for each type in the intrinsic semantics and the universal domain used in the untyped semantics, and we will show that these pairs "bracket" the logical relations. Finally, we will use this result to derive an extrinsic semantics in which each type denotes a partial equivalence relation on the universal domain.

In the course of this paper, we will use a variety of notations for functions. When f is a function, we write $\mathrm{dom}\, f$ for its domain. When $\iota_1, \ldots, \iota_n$ are distinct, we write $[f \mid \iota_1 : x_1 \mid \ldots \mid \iota_n : x_n]$ for the function with domain $\mathrm{dom}\, f \cup \{\iota_1, \ldots, \iota_n\}$ that maps each ι_k into x_k and all other arguments ι' into $f\,\iota'$; in the special case where f is the empty function, we write $[\iota_1 : x_1 \mid \ldots \mid \iota_n : x_n]$.

We write $f\,;g$ for the composition of functions f and g in diagrammatic order, and I_D for the identity function on the domain D. We assume that function application is left-associative, e.g., that $f\,x\,y$ abbreviates $(f\,x)y$.

1 Syntax and Typing Rules

In defining the syntax and type system of our illustrative language, we will use the following metavariables, sometimes with decorations, to range over denumerably infinite sets of syntactic entities:

ι: variables (sometimes called identifiers)

p: phrases

θ: types

π: type assignments.

We write I to denote the set of all variables.

Since our language is an extension of the lambda calculus, a phrase may be a *variable*, an *abstraction*, or an *application*:

$$p ::= \iota \mid \lambda\iota.\, p' \mid p'\, p''$$

In addition, there will be a *fixed-point* expression for defining a value by recursion, and a *conditional* expression that branches on whether a number

is positive to choose between evaluating different subexpressions:

$$p ::= \mathbf{Y}\, p' \mid \mathbf{if}\ p' > 0\ \mathbf{then}\ p''\ \mathbf{else}\ p'''$$

Finally, as primitives, we will have typical constants and operations for integers:

$$p ::= 0 \mid 1 \mid 2 \mid \cdots \mid p' + p''$$

(Additional integer operators would be treated the same way as $+$.)

Types and type assignments can also be defined by an abstract grammar:

$$\theta ::= \mathbf{int} \mid \theta_1 \to \theta_2$$
$$\pi ::= \iota_1{:}\theta_1, \ldots, \iota_n{:}\theta_n$$

Here, $\iota_1{:}\theta_1, \ldots, \iota_n{:}\theta_n$ is a concrete representation of a type assignment that, abstractly, is the function on the set $\{\iota_1, \ldots, \iota_n\}$ that maps each variable ι_k into the type θ_k. This implies that $\iota_1, \ldots, \iota_n$ must be distinct, and that permuting the pairs $\iota_k{:}\theta_k$ will not change the type assignment.

Informally, the primitive type **int** denotes the set of integers, while $\theta_1 \to \theta_2$ denotes the set of functions that map values of type θ_1 into values of type θ_2.

If π is a type assignment, p is a phrase, and θ is a type, then the formula $\pi \vdash p : \theta$ is a *typing judgement*, or more briefly a *typing*, which is read "p has type θ under π". The valid typing judgements are defined by inference rules. (In the first two rules, we rely on the fact that type assignments are functions on variables.)

$$\frac{}{\pi \vdash \iota : \pi\,\iota} \quad \text{when } \iota \in \mathrm{dom}\,\pi$$

$$\frac{[\,\pi \mid \iota{:}\theta_1\,] \vdash p' : \theta_2}{\pi \vdash \lambda\iota.\, p' : \theta_1 \to \theta_2} \qquad \frac{\pi \vdash p' : \theta_1 \to \theta_2 \qquad \pi \vdash p'' : \theta_1}{\pi \vdash p'\,p'' : \theta_2}$$

$$\frac{\pi \vdash p' : \theta \to \theta}{\pi \vdash \mathbf{Y}\,p' : \theta}$$

$$\frac{\pi \vdash p' : \mathbf{int} \qquad \pi \vdash p'' : \theta \qquad \pi \vdash p''' : \theta}{\pi \vdash \mathbf{if}\ p' > 0\ \mathbf{then}\ p''\ \mathbf{else}\ p''' : \theta}$$

$$\frac{}{\pi \vdash n : \mathbf{int}} \quad \text{when } n \text{ is a numeral} \qquad \frac{\pi \vdash p' : \mathbf{int} \qquad \pi \vdash p'' : \mathbf{int}}{\pi \vdash p' + p'' : \mathbf{int}.}$$

2 An Intrinsic Semantics

To give an intrinsic denotational semantics to our illustrative language, we must define the meanings of types, type assignments, and typing judgements. Specifically, we must give:

- for each type θ, a domain $[\![\theta]\!]$ of *values* appropriate to θ,
- for each type assignment π, a domain $[\![\pi]\!]^*$ of *environments* appropriate to π,
- for each valid typing judgement $\pi \vdash p : \theta$, a continuous function $[\![\pi \vdash p : \theta]\!]$ from $[\![\pi]\!]^*$ to $[\![\theta]\!]$, called the *meaning of p with respect to π and θ*.

We define a *predomain* to be a poset with least upper bounds of all increasing chains, and a *domain* to be a predomain with a least element, which we will denote by $\bot$. (In fact, all of the domains we will use are Scott domains, i.e., nonempty partially ordered sets that are directed complete, bounded complete, algebraic, and countably based, but we will not make use of this fact.) A *continuous* function is one that preserves least upper bounds of all increasing chains; it is *strict* if it also preserves least elements.

In what follows, we will write $\mathbf{Z}$ for the sets of integers and $\mathbf{Z}_\bot$ for the flat domain obtained by discretely ordering the integers and then adding a least element $\bot$ below them. When P is a predomain and D is a domain, we write $P \Rightarrow D$ for the pointwise-ordered domain of continuous functions from P to D.

The meanings of types and type assignments are defined by induction on their structure:

Definition 2.1 *For types θ and type assignments π, the domains $[\![\theta]\!]$ and $[\![\pi]\!]^*$ are such that*

$$[\![\mathbf{int}]\!] = \mathbf{Z}_\bot$$

$$[\![\theta_1 \to \theta_2]\!] = [\![\theta_1]\!] \Rightarrow [\![\theta_2]\!]$$

$$[\![\iota_1 : \theta_1, \ldots, \iota_n : \theta_n]\!]^* = \Big\{ [\,\iota_1 : x_1 \mid \ldots \mid \iota_n : x_n\,] \;\Big|\; x_1 \in [\![\theta_1]\!], \ldots, x_n \in [\![\theta_n]\!] \Big\}.$$

(The set on the right of the final equation is a Cartesian product, indexed by the variables $\iota_1, \ldots, \iota_n$, that becomes a domain when ordered componentwise.)

On the other hand, the meanings of typing judgements are defined by induction on the structure of proofs of these judgements. Specifically, for each inference rule, we give a semantic equation that expresses the meaning of a proof in which the final inference is an instance of that rule, in terms of the meanings of its immediate subproofs.

To write such equations succinctly, we write $\mathcal{P}(J)$ to denote a proof of the judgement J. For example, corresponding to the inference rule for function application,

$$\frac{\pi \vdash p' : \theta_1 \to \theta_2 \qquad \pi \vdash p'' : \theta_1}{\pi \vdash p'\,p'' : \theta_2},$$

we have the semantic equation

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2) \qquad \mathcal{P}(\pi \vdash p'' : \theta_1)}{\pi \vdash p'\,p'' : \theta_2} \right]\!\!\right] \eta$$

$$= [\![\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2)]\!]\eta\Big([\![\mathcal{P}(\pi \vdash p'' : \theta_1)]\!]\eta\Big),$$

which asserts that the meaning of a proof of $\pi \vdash p'\,p'' : \theta_2$ in which the final inference is an instance of the function-appplication rule, is a function that, when applied to an environment η, gives $(m'\,\eta)(m''\,\eta)$, where m' and m'' are the meanings of the immediate subproofs of the proof of $\pi \vdash p'\,p'' : \theta_2$.

Before proceeding further, we must warn the reader of a subtle complication in this way of defining semantics. For a sufficiently rich language, there may be many different proofs of the same typing judgement, and there is no a priori guarantee that these proofs will have the same meaning. If our intrinsic semantics is to make sense, so that we can take the meaning $[\![\mathcal{P}(J)]\!]$ of any proof of a judgement J to be the meaning of J itself, then we must have the property of coherence:

Definition 2.2 *An intrinsic semantics is said to be* coherent *if all proofs of the same judgement have the same meaning.*

In fact, as we will see in Section 5, our intrinsic semantics is coherent.

The following are the semantic equations for the intrinsic semantics of our illustrative language:

$$\left[\!\!\left[\frac{}{\pi \vdash \iota : \pi\,\iota} \right]\!\!\right] \eta = \eta\,\iota$$

$$\left[\!\!\left[\frac{\mathcal{P}([\,\pi \mid \iota{:}\theta_1\,] \vdash p' : \theta_2)}{\pi \vdash \lambda\iota.\,p' : \theta_1 \to \theta_2} \right]\!\!\right] \eta = \lambda x \in [\![\theta_1]\!].\ [\![\mathcal{P}([\,\pi \mid \iota{:}\theta_1\,] \vdash p' : \theta_2)]\!][\,\eta \mid \iota{:}x\,]$$

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2) \qquad \mathcal{P}(\pi \vdash p'' : \theta_1)}{\pi \vdash p'\,p'' : \theta_2} \right]\!\!\right] \eta$$

$$= [\![\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2)]\!]\eta\Big([\![\mathcal{P}(\pi \vdash p'' : \theta_1)]\!]\eta\Big)$$

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p' : \theta \to \theta)}{\pi \vdash \mathbf{Y}\,p' : \theta} \right]\!\!\right] \eta = \bigsqcup_{n=0}^{\infty} \Big([\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta\Big)^n \bot$$

$$\left[\!\left[\frac{\mathcal{P}(\pi \vdash p' : \mathbf{int}) \qquad \mathcal{P}(\pi \vdash p'' : \theta) \qquad \mathcal{P}(\pi \vdash p''' : \theta)}{\pi \vdash \mathbf{if}\ p' > 0\ \mathbf{then}\ p''\ \mathbf{else}\ p''' : \theta}\right]\!\right] \eta$$

$$= \begin{cases} \bot & \text{when } [\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta = \bot \\ [\![\mathcal{P}(\pi \vdash p'' : \theta)]\!]\eta & \text{when } [\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta > 0 \\ [\![\mathcal{P}(\pi \vdash p''' : \theta)]\!]\eta & \text{when } [\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta \leq 0 \end{cases}$$

$$\left[\!\left[\frac{}{\pi \vdash n : \mathbf{int}}\right]\!\right] \eta = n$$

$$\left[\!\left[\frac{\mathcal{P}(\pi \vdash p' : \mathbf{int}) \quad \mathcal{P}(\pi \vdash p'' : \mathbf{int})}{\pi \vdash p' + p'' : \mathbf{int}}\right]\!\right] \eta$$

$$= [\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta + [\![\mathcal{P}(\pi \vdash p'' : \mathbf{int})]\!]\eta.$$

3 An Untyped Semantics

Next, we consider the untyped semantics of our illustrative language. Here, independently of the type system, each phrase p possesses a unique meaning that is a mapping from environments to values, where environments map variables into values, and values range over a "universal" domain U:

$$[\![p]\!] \in E \Rightarrow U \qquad \text{where } E = (I \Rightarrow U).$$

It is vital that this untyped semantics be call-by-name, and that U be rich enough to contain "representations" of all the typed values used in the intrinsic semantics of the previous section. These conditions, however, do not fully determine the untyped semantics. To be general, therefore, rather than specify a particular universal domain, we simply state properties of U that will be sufficient for our development. (In fact, these properties hold for a variety of untyped call-by-name models of our illustrative language.)

Specifically, we require the domains $\mathbf{Z}_\bot$ of integers (viewed as primitive values) and $U \Rightarrow U$ of continuous functions (viewed as functional values) to be embeddable in U by pairs of continuous functions:

$$\mathbf{Z}_\bot \underset{\Psi_\mathrm{p}}{\overset{\Phi_\mathrm{p}}{\rightleftarrows}} U \qquad\qquad U \Rightarrow U \underset{\Psi_\mathrm{f}}{\overset{\Phi_\mathrm{f}}{\rightleftarrows}} U,$$

where each Φ_i, Ψ_i is an *embedding-retraction* pair, i.e., each composition $\Phi_i \,;\, \Psi_i$ is an identity function on the embedded domain.

Using these embedding-retraction pairs, it is straightforward to give semantic equations defining the untyped semantics of our language:

$$[\![\iota]\!]\,\varepsilon = \varepsilon\,\iota$$

$$[\![\lambda\iota.\,p']\!]\,\varepsilon = \Phi_{\mathrm{f}}(\lambda y \in U.\,[\![p']\!][\,\varepsilon \mid \iota\colon y\,])$$

$$[\![p'\,p'']\!]\,\varepsilon = \Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)([\![p'']\!]\,\varepsilon)$$

$$[\![\mathbf{Y}\,p']\!]\,\varepsilon = \bigsqcup_{n=0}^{\infty}\big(\Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)\big)^n\,\bot$$

$$[\![\mathbf{if}\ p' > 0\ \mathbf{then}\ p''\ \mathbf{else}\ p'''']\!]\,\varepsilon = \begin{cases} \bot & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) = \bot \\ [\![p'']\!]\,\varepsilon & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) > 0 \\ [\![p''']\!]\,\varepsilon & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) \le 0 \end{cases}$$

$$[\![n]\!]\varepsilon = \Phi_{\mathrm{p}}\,n$$

$$[\![p' + p'']\!]\varepsilon = \Phi_{\mathrm{p}}(\Psi_{\mathrm{p}}([\![p']\!]\varepsilon) + \Psi_{\mathrm{p}}([\![p'']\!]\varepsilon)).$$

4 Logical Relations

Our next task is to connect the intrinsic and untyped semantics by means of a type-indexed family ρ of relations such that

$$\rho[\theta] \subseteq [\![\theta]\!] \times U.$$

The members $\rho[\theta]$ of this family are called *logical relations*. Informally, $\langle x, y\rangle \in \rho[\theta]$ means that the value x of type θ is represented by the untyped value y.

(Logical relations [3] are most often used to connect two intrinsic typed semantics, but the idea works just as well to connect an intrinsic and an untyped semantics.)

The logical relations are defined by induction on the structure of types:

Definition 4.1 *For types θ, the* logical relations $\rho[\theta] \subseteq [\![\theta]\!] \times U$ *are such that:*

$$\langle x\,, y\rangle \in \rho[\mathbf{int}] \quad \textit{iff} \quad x = \Psi_{\mathrm{p}} y$$

$$\langle f, g\rangle \in \rho[\theta_1 \to \theta_2] \quad \textit{iff} \quad \forall\langle x, y\rangle \in \rho[\theta_1].\,\langle f\,x\,, \Psi_{\mathrm{f}}\,g\,y\rangle \in \rho[\theta_2].$$

To explicate the logical relations, we begin with two domain-theoretic properties that will be necessary to deal with recursion:

Definition 4.2 *A relation r between domains is*

- strict *iff* $\langle \bot, \bot \rangle \in r$,
- chain-complete *iff, whenever* $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$ *and* $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$ *are increasing sequences such that each* $\langle x_i, y_i \rangle \in r$,

$$\langle \bigsqcup_{i=0}^{\infty} x_i, \bigsqcup_{i=0}^{\infty} y_i \rangle \in r.$$

Lemma 4.3 *For all types θ, the logical relations $\rho[\theta]$ are strict and chain-complete.*

Proof. We first note that, if Φ, Ψ is any embedding-retraction pair, then $\bot \sqsubseteq \Phi\bot$, and since Ψ is monotone (since it is continuous) and $\Phi\,;\Psi$ is an identity, $\Psi\bot \sqsubseteq \Psi(\Phi\bot) = \bot$. Then since $\bot \sqsubseteq \Psi\bot$, we have $\Psi\bot = \bot$, i.e., Ψ is a strict function.

The main proof is by induction on the structure of θ.

- Suppose θ is **int**. Since Ψ_{p} is strict, we have $\bot = \Psi_{\mathrm{p}}\bot$, so that $\langle \bot, \bot \rangle \in \rho[\mathbf{int}]$.

  Now suppose that x_i and y_i are increasing sequences, in $\mathbf{Z}_\bot$ and U respectively, such that $\langle x_i, y_i \rangle \in \rho[\mathbf{int}]$. Then $x_i = \Psi_{\mathrm{p}} y_i$ for $i \geq 0$, so that $\bigsqcup_{i=0}^{\infty} x_i = \bigsqcup_{i=0}^{\infty} \Psi_{\mathrm{p}} y_i$, and since Ψ_{p} is continuous, $\bigsqcup_{i=0}^{\infty} x_i = \Psi_{\mathrm{p}}(\bigsqcup_{i=0}^{\infty} y_i)$. Thus $\langle \bigsqcup_{i=0}^{\infty} x_i, \bigsqcup_{i=0}^{\infty} y_i \rangle \in \rho[\mathbf{int}]$.

- Suppose θ is $\theta_1 \to \theta_2$. Let $f = \bot$ and $g = \bot$, so that $\Psi_{\mathrm{f}}\, g = \bot$, since Ψ_{f} is strict. Then, for any $\langle x, y \rangle \in \rho[\theta_1]$, since the least element of a domain of functions is the constant function yielding $\bot$, $f\, x = \bot$ and $\Psi_{\mathrm{f}}\, g\, y = \bot$. By the induction hypothesis for θ_2, $\langle f\, x, \Psi_{\mathrm{f}}\, g\, y \rangle = \langle \bot, \bot \rangle \in \rho[\theta_2]$, and since this holds for all $\langle x, y \rangle \in \rho[\theta_1]$, we have $\langle f, g \rangle = \langle \bot, \bot \rangle \in \rho[\theta_1 \to \theta_2]$.

  Now suppose that f_i and g_i are increasing sequences, in $[\![\theta_1 \to \theta_2]\!]$ and U respectively, such that $\langle f_i, g_i \rangle \in \rho[\theta_1 \to \theta_2]$. Let $\langle x, y \rangle \in \rho[\theta_1]$. Then $\langle f_i\, x, \Psi_{\mathrm{f}}\, g_i\, y \rangle \in \rho[\theta_2]$ for $i \geq 0$, and since function application and Ψ_{f} are monotone, $f_i\, x$ and $\Psi_{\mathrm{f}}\, g_i\, y$ are increasing sequences. Then, since Ψ_{f} is continuous, and a least upper bound of functions distributes through application, the induction hypothesis for θ_2 gives

  $$\langle (\bigsqcup_{i=0}^{\infty} f_i)\, x, \Psi_{\mathrm{f}} (\bigsqcup_{i=0}^{\infty} g_i)\, y \rangle = \langle \bigsqcup_{i=0}^{\infty} f_i\, x, \bigsqcup_{i=0}^{\infty} \Psi_{\mathrm{f}}\, g_i\, y \rangle \in \rho[\theta_2],$$

  and since this holds for all $\langle x, y \rangle \in \rho[\theta_1]$, we have $\langle \bigsqcup_{i=0}^{\infty} f_i, \bigsqcup_{i=0}^{\infty} g_i \rangle \in \rho[\theta_1 \to \theta_2]$.

END OF PROOF

Now we can establish our central result. Essentially, it asserts that, in related environments, the typed and untyped meanings of the same expression give related values, where the type-dependent notion of "related" is given by the logical relations:

Theorem 4.4 (The Logical Relations Theorem) *Suppose* $\mathcal{P}(\pi \vdash p : \theta)$ *is a proof of the typing judgement* $\pi \vdash p : \theta$, *and*

$$\forall \iota \in \operatorname{dom} \pi.\ \eta\,\iota \in [\![\pi\,\iota]\!]$$
$$\varepsilon \in I \Rightarrow U \tag{A}$$
$$\forall \iota \in \operatorname{dom} \pi.\ \langle \eta\,\iota, \varepsilon\,\iota\rangle \in \rho[\pi\,\iota],$$

Then

$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \in \rho[\theta]. \tag{B}$$

Proof. By induction on the structure of the proof $\mathcal{P}(\pi \vdash p : \theta)$. More precisely, we prove by induction on n that, for all π, η, ε, p, θ, and $\mathcal{P}(\pi \vdash p : \theta)$, if the depth of $\mathcal{P}(\pi \vdash p : \theta)$ is at most n, and the assumptions (A) hold, then (B) holds. However, in all of the following cases except that for abstractions (i.e., $\lambda\iota.\ p$), the induction hypotheses are only applied for the same values of π, η, and ε as in the theorem being proved. (Abstractions are the only exception because they are the only binding construction in our language.)

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is

  $$\frac{}{\pi \vdash \iota : \pi\,\iota,}$$

  so that

  $$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = \eta\,\iota$$
  $$[\![p]\!]\varepsilon = \varepsilon\,\iota.$$

  Assume (A). Then

  $$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle = \langle \eta\,\iota, \varepsilon\,\iota\rangle \in \rho[\pi\,\iota] = \rho[\theta].$$

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is

  $$\frac{\mathcal{P}([\,\pi \mid \iota{:}\theta_1\,] \vdash p' : \theta_2)}{\pi \vdash \lambda\iota.\ p' : \theta_1 \to \theta_2,}$$

  so that

  $$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = \lambda x \in [\![\theta_1]\!].\ [\![\mathcal{P}([\,\pi \mid \iota{:}\theta_1\,] \vdash p' : \theta_2)]\!][\,\eta \mid \iota{:}x\,]$$
  $$[\![p]\!]\varepsilon = \Phi_{\mathrm{f}}(\lambda y \in U.\ [\![p']\!][\,\varepsilon \mid \iota{:}y\,]).$$

  Assume (A), and suppose $\langle x, y\rangle \in \rho[\theta_1]$. Then

  $$\forall \iota' \in \operatorname{dom}[\,\pi \mid \iota{:}\theta_1\,].\ [\,\eta \mid \iota{:}x\,]\,\iota' \in [\![[\,\pi \mid \iota{:}\theta_1\,]\,\iota']\!]$$
  $$[\,\varepsilon \mid \iota{:}y\,] \in I \Rightarrow U \tag{A$'$}$$
  $$\forall \iota' \in \operatorname{dom}[\,\pi \mid \iota{:}\theta_1\,].\ \langle [\,\eta \mid \iota{:}x\,]\,\iota', [\,\varepsilon \mid \iota{:}y\,]\,\iota'\rangle \in \rho[[\,\pi \mid \iota{:}\theta_1\,]\,\iota'],$$

so that

$$\begin{aligned}&\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta\, x, \Psi_{\mathrm{f}}([\![p]\!]\varepsilon)\, y\rangle\\ &\quad = \langle [\![\mathcal{P}([\,\pi \mid \iota\!:\!\theta_1\,] \vdash p' : \theta_2)]\!][\,\eta \mid \iota\!:\!x\,], \Psi_{\mathrm{f}}(\Phi_{\mathrm{f}}(\lambda y.\ [\![p']\!][\,\varepsilon \mid \iota\!:\!y\,]))y\rangle\\ &\quad = \langle [\![\mathcal{P}([\,\pi \mid \iota\!:\!\theta_1\,] \vdash p' : \theta_2)]\!][\,\eta \mid \iota\!:\!x\,], [\![p']\!][\,\varepsilon \mid \iota\!:\!y\,]\rangle\\ &\quad \in \rho[\theta_2],\end{aligned}$$

where the last step follows from the induction hypothesis for $\mathcal{P}([\,\pi \mid \iota\!:\!\theta_1\,] \vdash p' : \theta_2)$, taking (A) to be (A′). Then, since this holds for all $\langle x, y\rangle \in \rho[\theta_1]$, the definition (4.1) of $\rho[\theta_1 \to \theta_2]$ gives

$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \in \rho[\theta_1 \to \theta_2].$$

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is

$$\frac{\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2) \qquad \mathcal{P}(\pi \vdash p'' : \theta_1)}{\pi \vdash p'\, p'' : \theta_2\,,}$$

so that

$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = [\![\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2)]\!]\eta\Big([\![\mathcal{P}(\pi \vdash p'' : \theta_1)]\!]\eta\Big)$$

$$[\![p]\!]\varepsilon = \Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)([\![p'']\!]\,\varepsilon).$$

Assume (A). Then the induction hypothesis for $\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2)$ gives

$$\langle [\![\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2)]\!]\eta, [\![p']\!]\,\varepsilon\rangle \in \rho[\theta_1 \to \theta_2],$$

and the induction hypothesis for $\mathcal{P}(\pi \vdash p'' : \theta_1)$ gives

$$\langle [\![\mathcal{P}(\pi \vdash p'' : \theta_1)]\!]\eta, [\![p'']\!]\,\varepsilon\rangle \in \rho[\theta_1],$$

so that

$$\begin{aligned}&\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle\\ &\quad = \langle [\![\mathcal{P}(\pi \vdash p' : \theta_1 \to \theta_2)]\!]\eta([\![\mathcal{P}(\pi \vdash p'' : \theta_1)]\!]\eta), \Psi_{\mathrm{f}}([\![p']\!]\varepsilon)([\![p'']\!]\varepsilon)\rangle\\ &\quad \in \rho[\theta_2],\end{aligned}$$

by the definition (4.1) of $\rho[\theta_1 \to \theta_2]$.

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is

$$\frac{\mathcal{P}(\pi \vdash p' : \theta \to \theta)}{\pi \vdash \mathbf{Y}\, p' : \theta,}$$

so that

$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = \bigsqcup_{n=0}^{\infty} \big([\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta\big)^n \bot$$

$$[\![p]\!]\varepsilon = \bigsqcup_{n=0}^{\infty} \big(\Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)\big)^n \bot.$$

Assume (A). By the induction hypothesis, we have

$$\langle [\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta, [\![p']\!]\,\varepsilon\rangle \in \rho[\theta \to \theta].$$

Next we can show, by induction on n, that

$$\Big\langle \big([\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta\big)^n \bot, \big(\Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)\big)^n \bot \Big\rangle \in \rho[\theta].$$

The case for $n = 0$ follows since, by Lemma 4.3, $\rho[\theta]$ is strict. For the induction step, we have

$$\begin{aligned}
&\Big\langle \big([\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta\big)^{n+1} \bot, \big(\Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)\big)^{n+1} \bot \Big\rangle \\
&\quad = \Big\langle [\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta\Big(\big([\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta\big)^n \bot\Big), \\
&\qquad\qquad \Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)\Big(\big(\Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)\big)^n \bot\Big)\Big\rangle \\
&\quad \in \rho[\theta],
\end{aligned}$$

by the induction hypothesis for n and the definition (4.1) of $\rho[\theta \to \theta]$.

Finally,

$$\begin{aligned}
&\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \\
&\quad = \Big\langle \bigsqcup_{n=0}^{\infty} \big([\![\mathcal{P}(\pi \vdash p' : \theta \to \theta)]\!]\eta\big)^n \bot, \bigsqcup_{n=0}^{\infty} \big(\Psi_{\mathrm{f}}([\![p']\!]\,\varepsilon)\big)^n \bot \Big\rangle \\
&\quad \in \rho[\theta],
\end{aligned}$$

since $\rho[\theta]$ is chain-complete by Lemma 4.3.

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is

$$\frac{\mathcal{P}(\pi \vdash p' : \mathbf{int}) \qquad \mathcal{P}(\pi \vdash p'' : \theta) \qquad \mathcal{P}(\pi \vdash p''' : \theta)}{\pi \vdash \mathbf{if}\ p' > 0\ \mathbf{then}\ p''\ \mathbf{else}\ p''' : \theta,}$$

so that

$$[\![\mathcal{P}(\pi \vdash p\colon\theta)]\!]\eta = \begin{cases} \bot & \text{when } [\![\mathcal{P}(\pi \vdash p'\colon\mathbf{int})]\!]\eta = \bot \\ [\![\mathcal{P}(\pi \vdash p''\colon\theta)]\!]\eta & \text{when } [\![\mathcal{P}(\pi \vdash p'\colon\mathbf{int})]\!]\eta > 0 \\ [\![\mathcal{P}(\pi \vdash p'''\colon\theta)]\!]\eta & \text{when } [\![\mathcal{P}(\pi \vdash p'\colon\mathbf{int})]\!]\eta \leq 0 \end{cases}$$

$$[\![p]\!]\varepsilon = \begin{cases} \bot & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) = \bot \\ [\![p'']\!]\,\varepsilon & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) > 0 \\ [\![p''']\!]\,\varepsilon & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) \leq 0. \end{cases}$$

Assume (A). By the induction hypothesis for $\mathcal{P}(\pi \vdash p' : \mathbf{int})$,

$$\langle [\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta, [\![p']\!]\varepsilon\rangle \in \rho[\mathbf{int}],$$

so that $[\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta = \Psi_{\mathrm{p}}([\![p']\!]\varepsilon)$.

There are three cases:

- If $\Psi_{\mathrm{p}}[\![p']\!]\varepsilon = \bot$, then

  $$\langle [\![\mathcal{P}(\pi \vdash p\colon\theta)]\!]\eta, [\![p]\!]\varepsilon\rangle = \langle \bot, \bot\rangle,$$

  which belongs to $\rho[\theta]$ since $\rho[\theta]$ is strict.
- If $\Psi_{\mathrm{p}}[\![p']\!]\varepsilon > 0$, then

  $$\langle [\![\mathcal{P}(\pi \vdash p\colon\theta)]\!]\eta, [\![p]\!]\varepsilon\rangle = \langle [\![\mathcal{P}(\pi \vdash p''\colon\theta)]\!]\eta, [\![p'']\!]\varepsilon\rangle,$$

  which belongs to $\rho[\theta]$ by the induction hypothesis for $\mathcal{P}(\pi \vdash p''\colon\theta)$.
- If $\Psi_{\mathrm{p}}[\![p']\!]\varepsilon \leq 0$, then

  $$\langle [\![\mathcal{P}(\pi \vdash p\colon\theta)]\!]\eta, [\![p]\!]\varepsilon\rangle = \langle [\![\mathcal{P}(\pi \vdash p'''\colon\theta)]\!]\eta, [\![p''']\!]\varepsilon\rangle,$$

  which belongs to $\rho[\theta]$ by the induction hypothesis for $\mathcal{P}(\pi \vdash p'''\colon\theta)$.

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is

  $$\frac{}{\pi \vdash n : \mathbf{int}},$$

  so that

  $$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = n$$

  $$[\![p]\!]\varepsilon = \Phi_{\mathrm{p}}\, n.$$

  Then, since Φ_{p} ; Ψ_{p} is an identity, $[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = \Psi_{\mathrm{p}}(\Phi_{\mathrm{p}}\, n) = \Psi_{\mathrm{p}}([\![p]\!]\varepsilon)$, so that

  $$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \in \rho[\mathbf{int}].$$

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is

$$\frac{\mathcal{P}(\pi \vdash p' : \mathbf{int}) \quad \mathcal{P}(\pi \vdash p'' : \mathbf{int})}{\pi \vdash p' + p'' : \mathbf{int},}$$

so that

$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = [\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta + [\![\mathcal{P}(\pi \vdash p'' : \mathbf{int})]\!]\eta$$

$$[\![p]\!]\varepsilon = \Phi_{\mathrm{p}}(\Psi_{\mathrm{p}}([\![p']\!]\varepsilon) + \Psi_{\mathrm{p}}([\![p'']\!]\varepsilon)).$$

Assume (A). By the induction hypothesis for $\mathcal{P}(\pi \vdash p' : \mathbf{int})$, we have

$$[\![\mathcal{P}(\pi \vdash p' : \mathbf{int})]\!]\eta = \Psi_{\mathrm{p}}([\![p']\!]\varepsilon),$$

and a similar argument holds for p''. Then

$$\begin{aligned}[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta &= \Psi_{\mathrm{p}}([\![p']\!]\varepsilon) + \Psi_{\mathrm{p}}([\![p'']\!]\varepsilon) \\ &= \Psi_{\mathrm{p}}(\Phi_{\mathrm{p}}(\Psi_{\mathrm{p}}([\![p']\!]\varepsilon) + \Psi_{\mathrm{p}}([\![p'']\!]\varepsilon))) \\ &= \Psi_{\mathrm{p}}([\![p]\!]\varepsilon),\end{aligned}$$

since $\Phi_{\mathrm{p}} \,;\, \Psi_{\mathrm{p}}$ is an identity. Thus

$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \in \rho[\mathbf{int}].$$

END OF PROOF

5 Bracketing

Next, we show that the domains $[\![\theta]\!]$ that are the meanings of types can be embedded in the universal domain U by a type-indexed family of function pairs:

$$[\![\theta]\!] \underset{\psi[\theta]}{\overset{\phi[\theta]}{\rightleftarrows}} U.$$

We will show that these are embedding-retraction pairs, and that they are closely related to the logical relations defined in the previous section. (The idea that types denote retractions on a universal domain is due to Scott [4].)

Definition 5.1 *For types θ, the functions $\phi[\theta] \in [\![\theta]\!] \to U$ and $\psi[\theta] \in U \to [\![\theta]\!]$ are such that*

$$\phi[\mathbf{int}]x = \Phi_{\mathrm{p}}\, x$$

$$\psi[\mathbf{int}]y = \Psi_{\mathrm{p}}\, y$$

$$\phi[\theta_1 \to \theta_2]f = \Phi_{\mathrm{f}}(\psi[\theta_1] \,;\, f \,;\, \phi[\theta_2])$$

$$\psi[\theta_1 \to \theta_2]y = \phi[\theta_1] \,;\, \Psi_{\mathrm{f}} y \,;\, \psi[\theta_2].$$

These function pairs are related to the logical relations $\rho[\theta]$ by the following theorem:

Theorem 5.2 (The Bracketing Theorem) *For each type θ:*

1. *For all $x \in [\![\theta]\!]$, $\langle x, \phi[\theta]x\rangle \in \rho[\theta]$.*
2. *For all $\langle x, y\rangle \in \rho[\theta]$, $x = \psi[\theta]y$.*

Proof. The proof is by induction on the structure of θ:

- Suppose θ is **int** and $x \in [\![\mathbf{int}]\!]$. From the fact that $\Phi_{\mathrm{p}} ; \Psi_{\mathrm{p}}$ is an identity, and the definition (5.1) of $\phi[\mathbf{int}]$, we have $x = \Psi_{\mathrm{p}}(\Phi_{\mathrm{p}}\, x) = \Psi_{\mathrm{p}}(\phi[\mathbf{int}]x)$. Then the definition (4.1) of $\rho[\mathbf{int}]$ gives $\langle x, \phi[\mathbf{int}]x\rangle \in \rho[\mathbf{int}]$.

  Now suppose $\langle x, y\rangle \in \rho[\mathbf{int}]$. Then the definitions (4.1) of $\rho[\mathbf{int}]$ and (5.1) of $\psi[\mathbf{int}]$ give $x = \Psi_{\mathrm{p}}\, y = \psi[\mathbf{int}]y$.

- Suppose θ is $\theta_1 \to \theta_2$, and $f \in [\![\theta_1 \to \theta_2]\!]$. Let $\langle x, y\rangle \in \rho[\theta_1]$. By the second part of the induction hypothesis for θ_1, $x = \psi[\theta_1]y$. Then

$$\begin{aligned}
\phi[\theta_2](f\, x) &= \phi[\theta_2](f(\psi[\theta_1]y)) \\
&= (\psi[\theta_1] ; f ; \phi[\theta_2])y \\
&= \Psi_{\mathrm{f}}(\Phi_{\mathrm{f}}(\psi[\theta_1] ; f ; \phi[\theta_2]))y \\
&= \Psi_{\mathrm{f}}(\phi[\theta_1 \to \theta_2]f)y,
\end{aligned}$$

  from the fact that $\Phi_{\mathrm{f}} ; \Psi_{\mathrm{f}}$ is an identity, and the definition (5.1) of $\phi[\theta_1 \to \theta_2]$. Then by the first part of the induction hypothesis for θ_2,

$$\langle f\, x, \Psi_{\mathrm{f}}(\phi[\theta_1 \to \theta_2]f)y\rangle \in \rho[\theta_2],$$

  and since this holds for all $\langle x, y\rangle \in \rho[\theta_1]$, the definition (4.1) of $\rho[\theta_1 \to \theta_2]$ gives

$$\langle f, \phi[\theta_1 \to \theta_2]f\rangle \in \rho[\theta_1 \to \theta_2].$$

  Now suppose $\langle f, g\rangle \in \rho[\theta_1 \to \theta_2]$, and let $x \in [\![\theta_1]\!]$. By the first part of the induction hypothesis for θ_1, $\langle x, \phi[\theta_1]x\rangle \in \rho[\theta_1]$, so by the definition (4.1) of $\rho[\theta_1 \to \theta_2]$, $\langle f\, x, \Psi_{\mathrm{f}}g(\phi[\theta_1]x)\rangle \in \rho[\theta_2]$. Then by the second part of the induction hypothesis for θ_2, $f\, x = \psi[\theta_2](\Psi_{\mathrm{f}}g(\phi[\theta_1]x))$.

  Since this holds for all $x \in [\![\theta_1]\!]$, we have $f = \phi[\theta_1] ; \Psi_{\mathrm{f}}g ; \psi[\theta_2]$, and by the definition (5.1) of $\psi[\theta_1 \to \theta_2]$, $f = \psi[\theta_1 \to \theta_2]g$.

END OF PROOF

An immediate consequence of the bracketing theorem is that

Corollary 5.3 *The $\phi[\theta], \psi[\theta]$ are embedding-retraction pairs.*

Proof. Suppose $x \in [\![\theta]\!]$. By the first part of the bracketing theorem, $\langle x, \phi[\theta]x\rangle \in \rho[\theta]$; then by the second part, $x = \psi[\theta](\phi[\theta]x)$. Since this holds for all $x \in [\![\theta]\!]$, we have $\phi[\theta] \,;\, \psi[\theta] = I_{[\![\theta]\!]}$. END OF PROOF

The name "bracketing theorem" alludes to a more succinct formulation of the theorem as a subset relationship between graphs of functions and relations. Writing $\dagger$ for the reflection of a graph, we can restate the bracketing theorem as:

$$\textit{For each type } \theta,\ \phi[\theta] \subseteq \rho[\theta] \subseteq (\psi[\theta])^{\dagger}.$$

The essence of the bracketing theorem is that it connects the notion of representation provided by the logical relations with the different notion provided by the embedding-retraction pairs. For each typed value $x \in [\![\theta]\!]$, one can regard

- the set $\{\, y \mid x = \psi[\theta]y \,\}$ (i.e., the preimage of x under $\psi[\theta]$) as the set of untyped values that "weakly" represent x,
- the subset $\{\, y \mid \langle x, y\rangle \in \rho[\theta] \,\} \subseteq \{\, y \mid x = \psi[\theta]y \,\}$ as the set of untyped values that represent x,
- the member $\phi[\theta]x \in \{\, y \mid \langle x, y\rangle \in \rho[\theta] \,\}$ as the "best" or "canonical" representation of x.

An essential difference between representation and weak representation is that, since $\psi[\theta]$ is a total function, every untyped value "weakly" represents some typed value.

By combining the bracketing theorem with the logical relations theorem, one can express the intrinsic typed semantics of a phrase in terms of its untyped semantics:

Theorem 5.4 *Suppose* $\pi \vdash p : \theta$ *and* $\eta \in [\![\pi]\!]^*$. *Then*

$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = \psi[\theta]([\![p]\!](\lambda\iota.\ \mathbf{if}\ \iota \in \operatorname{dom}\pi\ \mathbf{then}\ \phi[\pi\,\iota](\eta\,\iota)\ \mathbf{else}\ \bot)).$$

Proof. Let $\varepsilon = \lambda\iota.\ \mathbf{if}\ \iota \in \operatorname{dom}\pi\ \mathbf{then}\ \phi[\pi\,\iota](\eta\,\iota)\ \mathbf{else}\ \bot$. For each $\iota \in \operatorname{dom}\pi$, by this definition of ε and the first part of the bracketing theorem,

$$\langle \eta\,\iota, \varepsilon\,\iota\rangle = \langle \eta\,\iota, \phi[\pi\,\iota](\eta\,\iota)\rangle \in \rho[\pi\,\iota].$$

Then the logical relations theorem gives

$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \in \rho[\theta],$$

and the second part of the bracketing theorem gives

$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = \psi[\theta]([\![p]\!]\varepsilon).$$

END OF PROOF

Theorem 5.4 expresses the meaning of a proof of the typing judgement $\pi \vdash p : \theta$ in terms of the untyped meaning of p and the retraction-embedding pairs associated with θ and the components of π — all of which are determined by the judgement itself rather than its proof. Thus every proof of the same judgement must have the same intrinsic semantics:

Corollary 5.5 *The intrinsic semantics is coherent.*

In fact, the coherence of the simple kind of language we have been considering here has long been known, but we believe this is an unusually elegant proof.

It should also be noted that the above theorem expresses a particular intrinsic semantics in terms of any of a variety of untyped semantics, i.e., all of the untyped semantics meeting the constraints in Section 3, where we permitted variations in the universal domain U and the way in which integers and functions are embedded within it.

For example, one might take the untyped semantics to be one in which η-reduction does or does not preserve meaning [1, Section 10.5], or one in which the fixed-point combinator $\lambda f.\ (\lambda x.\ f(x\,x))(\lambda x.\ f(x\,x))$ is or is not a *least* fixed-point operator [5].

6 An Extrinsic PER Semantics

Suppose we define a type-indexed family of relations between untyped values:

Definition 6.1 *For types θ, the relations $\sigma[\theta] \subseteq U \times U$ are such that*

$$\langle y, y' \rangle \in \sigma[\theta] \text{ iff } \exists x \in [\![\theta]\!].\ \langle x, y \rangle, \langle x, y' \rangle \in \rho[\theta].$$

(More abstractly, using relational composition, $\sigma[\theta] \stackrel{\text{def}}{=} (\rho[\theta])^\dagger\ ;\ \rho[\theta]$.)

Obviously, the $\sigma[\theta]$ are symmetric. Moreover, these relations are transitive:

Theorem 6.2 *If $\langle y, y' \rangle, \langle y', y'' \rangle \in \sigma[\theta]$, then $\langle y, y'' \rangle \in \sigma[\theta]$.*

Proof. If $\langle y, y' \rangle, \langle y', y'' \rangle \in \sigma[\theta]$, then there are $x, x' \in [\![\theta]\!]$ such that $\langle x, y \rangle$, $\langle x, y' \rangle, \langle x', y' \rangle, \langle x', y'' \rangle \in \rho[\theta]$. Then by the second part of the bracketing theorem, $x = \psi[\theta]y'$ and $x' = \psi[\theta]y'$, so that $x = x'$. Then $\langle x, y \rangle, \langle x, y'' \rangle \in \rho[\theta]$, so that $\langle y, y'' \rangle \in \sigma[\theta]$. END OF PROOF

Thus, the $\sigma[\theta]$ are partial equivalence relations (PER's). Although we have chosen to define them in terms of the logical relations, they can also be described directly by induction on the structure of θ:

Theorem 6.3

$$\langle y, y' \rangle \in \sigma[\mathbf{int}] \quad \textit{iff} \quad \Psi_{\mathrm{p}} y = \Psi_{\mathrm{p}} y'$$

$$\langle g, g' \rangle \in \sigma[\theta_1 \to \theta_2] \quad \textit{iff} \quad \forall \langle y, y' \rangle \in \sigma[\theta_1].\ \langle \Psi_{\mathrm{f}}\, g\, y, \Psi_{\mathrm{f}}\, g'\, y' \rangle \in \sigma[\theta_2].$$

Proof. Using the definitions (6.1) of σ, and (4.1) of $\rho[\mathbf{int}]$, we have $\langle y, y' \rangle \in \sigma[\mathbf{int}]$ iff there is an $x \in [\![\mathbf{int}]\!]$ such that $\langle x, y \rangle, \langle x, y' \rangle \in \rho[\mathbf{int}]$, iff there is an $x \in [\![\mathbf{int}]\!]$ such that $x = \Psi_{\mathrm{p}}\, y$ and $x = \Psi_{\mathrm{p}}\, y'$, iff $\Psi_{\mathrm{p}}\, y = \Psi_{\mathrm{p}}\, y'$.

Suppose $\langle g, g' \rangle \in \sigma[\theta_1 \to \theta_2]$ and $\langle y, y' \rangle \in \sigma[\theta_1]$. From the definition (6.1) of σ, there is an $f \in [\![\theta_1 \to \theta_2]\!]$ such that $\langle f, g \rangle, \langle f, g' \rangle \in \rho[\theta_1 \to \theta_2]$ and there is an $x \in [\![\theta_1]\!]$ such that $\langle x, y \rangle, \langle x, y' \rangle \in \rho[\theta_1]$. Then, from the definition (4.1) of $\rho[\theta_1 \to \theta_2]$, we have $\langle f\, x, \Psi_{\mathrm{f}}\, g\, y \rangle \in \rho[\theta_2]$ and $\langle f\, x, \Psi_{\mathrm{f}}\, g'\, y' \rangle \in \rho[\theta_2]$, and from the definition (6.1) of σ, we have $\langle \Psi_{\mathrm{f}}\, g\, y, \Psi_{\mathrm{f}}\, g'\, y' \rangle \in \sigma[\theta_2]$.

On the other hand, suppose

$$\forall \langle y, y' \rangle \in \sigma[\theta_1].\ \langle \Psi_{\mathrm{f}}\, g\, y, \Psi_{\mathrm{f}}\, g'\, y' \rangle \in \sigma[\theta_2],$$

and let $\langle x, y \rangle \in \rho[\theta_1]$. We have $\langle y, y \rangle \in \sigma[\theta_1]$ by the definition (6.1) of σ. Moreover, since the first half of the bracketing theorem gives $\langle x, \phi[\theta_1]x \rangle \in \rho[\theta_1]$, we also have $\langle \phi[\theta_1]x, y \rangle \in \sigma[\theta_1]$ by the definition (6.1) of σ. Then, by the supposition displayed above, we have both $\langle \Psi_{\mathrm{f}}\, g\, y, \Psi_{\mathrm{f}}\, g'\, y \rangle \in \sigma[\theta_2]$ and $\langle \Psi_{\mathrm{f}}\, g(\phi[\theta_1]x), \Psi_{\mathrm{f}}\, g'\, y \rangle \in \sigma[\theta_2]$, and since $\sigma[\theta_2]$ is symmetric and transitive, $\langle \Psi_{\mathrm{f}}\, g(\phi[\theta_1]x), \Psi_{\mathrm{f}}\, g\, y \rangle \in \sigma[\theta_2]$.

Thus we have $\langle \Psi_{\mathrm{f}}\, g(\phi[\theta_1]x), \Psi_{\mathrm{f}}\, \hat{g}\, y \rangle \in \sigma[\theta_2]$, where $\hat{g}$ is either g or g'. By the definition (6.1) of σ, there is a $w \in [\![\theta_2]\!]$ such that $\langle w, \Psi_{\mathrm{f}}\, g(\phi[\theta_1]x) \rangle \in \rho[\theta_2]$ and $\langle w, \Psi_{\mathrm{f}}\, \hat{g}\, y \rangle \in \rho[\theta_2]$. From the first of these inclusions, the second part of the bracketing theorem gives $w = \psi[\theta_2](\Psi_{\mathrm{f}}\, g(\phi[\theta_1]x))$, or $w = f\, x$, where f is the function $\phi[\theta_1]\ ;\ \Psi_{\mathrm{f}}\, g\ ;\ \psi[\theta_2]$. Thus the second inclusion can be written as $\langle f\, x, \Psi_{\mathrm{f}}\, \hat{g}\, y \rangle \in \rho[\theta_2]$. Since this holds for all $\langle x, y \rangle \in \rho[\theta_1]$, the definition (4.1) of $\rho[\theta_1 \to \theta_2]$ gives $\langle f, \hat{g} \rangle \in \rho[\theta_1 \to \theta_2]$, and since this holds when $\hat{g}$ is either g or g', the definition (6.1) of σ gives $\langle g, g' \rangle \in \sigma[\theta_1 \to \theta_2]$.

END OF PROOF

In terms of the notion of representation captured by the logical relations:

- $\langle y, y \rangle \in \sigma[\theta]$ means that y is a representation of some value of type θ.
- $\langle y, y' \rangle \in \sigma[\theta]$ means that y and y' are representions of the same value of type θ.

Thus, for each θ, the PER $\sigma[\theta]$ defines both the set $\{\, y \mid \langle y, y \rangle \in \sigma[\theta] \,\}$ of "representations" and also, when restricted to this set, an equivalence relation of "representing the same thing".

In early extrinsic denotational semantics, such as the Sethi-MacQueen model [7, 8], types played only the first of these roles; the insight that they denote PER's (rather than subsets) on a universal domain of untyped values is due to Scott [4, Section 7], [9, Section 5], [10].

In fact, the basic idea that types represent equivalence relations on subsets of some universe of "realizers" is much older. Two examples are described by Troelstra: the "hereditarily effective operations" (HEO) [11, Section 2.4.11], [12, Section 3.3], where the realizers are natural numbers (used as Gödel numbers), and the "extensional model of hereditarily continuous functionals" (ECF) [11, Section 2.6.5], [12, Section 3.9], where the realizers are functions from natural numbers to natural numbers. Troelstra attributes HEO to Kreisel [13, Section 4.2], and ECF to both Kreisel [13] and, independently, Kleene [14].

The common thread behind all these systems is that, to be continuous (or computable) a typed value must be represented by some realizer. An overview of realizability is given by Amadio and Curien [15, Chapter 15].

The combination of our untyped semantics with the PER's $\sigma[\theta]$ gives what we have called an *extrinsic* semantics. The essential connection between these entities is that, when a phrase satisfies a typing judgement, its untyped meaning respects the type-dependent notion of representation described by the $\sigma[\theta]$. More precisely,

Theorem 6.4 *Suppose* $\pi \vdash p : \theta$, *and* $\varepsilon, \varepsilon' \in I \Rightarrow U$ *satisfy*

$$\forall \iota \in \mathrm{dom}\,\pi.\ \langle \varepsilon\,\iota, \varepsilon'\,\iota \rangle \in \sigma[\pi\,\iota].$$

Then

$$\langle [\![p]\!]\varepsilon, [\![p]\!]\varepsilon' \rangle \in \sigma[\theta].$$

Proof. If ε and ε' are related as supposed, then for each $\iota \in \mathrm{dom}\,\pi$, there must be an $x \in [\![\pi\,\iota]\!]$ such that $\langle x, \varepsilon\,\iota \rangle, \langle x, \varepsilon'\,\iota \rangle \in \rho[\pi\,\iota]$. Thus there must be an environment $\eta \in [\![\pi]\!]^*$ such that

$$\forall \iota \in \mathrm{dom}\,\pi.\ \langle \eta\,\iota, \varepsilon\,\iota \rangle, \langle \eta\,\iota, \varepsilon'\,\iota \rangle \in \rho[\pi\,\iota].$$

Then by two applications of the logical relations theorem,

$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon \rangle, \langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon' \rangle \in \rho[\theta],$$

so that

$$\langle [\![p]\!]\varepsilon, [\![p]\!]\varepsilon' \rangle \in \sigma[\theta].$$

END OF PROOF

7 Further Work and Future Directions

The development we have presented here extends in a straightforward manner to a richer programming language that includes several primitive types, subtyping, and named products (with field-forgetting implicit conversions)

[16]. (Some additional properties of $\rho[\theta]$ and $\sigma[\theta]$ are also proved in this reference.)

Obviously, we would like to treat still richer type systems, such as intersection or polymorphic types. During the last year, we made a strenuous attempt to conquer intersection types, but we were unable to find a semantics for which we could prove the bracketing theorem. (This work was described in a talk at the Workshop on Intersection Types and Related Systems [17].)

It is also of interest to try to move in the opposite direction, from extrinsic to intrinsic semantics. In a sense this is straightforward: Given $\sigma[\theta]$, one simply takes $[\![\theta]\!]$ to be the set of equivalence classes of $\sigma[\theta]$. (More precisely, one takes the semantic category to be a category of PER's [15, Chapter 15].) In general, however, there may be no sensible way to order the set of equivalence classes to make $[\![\theta]\!]$ into a domain.

8 Acknowledgements

This research was supported in part by National Science Foundation Grant CCR-9804014. Much of the research was carried out during two delightful and productive visits to BRICS[1] in Aarhus, Denmark, September to November 1999 and May to June 2000.

9 References

[1] John C. Reynolds. *Theories of Programming Languages.* Cambridge University Press, Cambridge, England, 1998.

[2] Daniel Leivant. Typing and computational properties of lambda expressions. *Theoretical Computer Science*, 44(1):51–68, 1986.

[3] Gordon D. Plotkin. Lambda-definability and logical relations. Memorandum SAI–RM–4, University of Edinburgh, Edinburgh, Scotland, October 1973.

[4] Dana S. Scott. Data types as lattices. *SIAM Journal on Computing*, 5(3):522–587, September 1976.

[5] David M. R. Park. The Y-combinator in Scott's lambda-calculus models. Symposium on Theory of Programming, University of Warwick, unpublished; cited in [6], 1970.

[1] Basic Research in Computer Science (`http://www.brics.dk/`), Centre of the Danish National Research Foundation.

[6] Christopher P. Wadsworth. The relation between computational and denotational properties for Scott's D_∞-models of the lambda-calculus. *SIAM Journal on Computing*, 5(3):488–521, September 1976.

[7] David B. MacQueen and Ravi Sethi. A semantic model of types for applicative languages. In *Conference Record of the 1982 ACM Symposium on LISP and Functional Programming*, pages 243–252, New York, 1982. ACM.

[8] David B. MacQueen, Gordon D. Plotkin, and Ravi Sethi. An ideal model for recursive polymorphic types. *Information and Control*, 71(1–2):95–130, October–November 1986.

[9] Dana S. Scott. Lambda calculus: Some models, some philosophy. In Jon Barwise, H. Jerome Keisler, and Kenneth Kunen, editors, *The Kleene Symposium*, volume 101 of *Studies in Logic and the Foundations of Mathematics*, pages 223–265, Amsterdam, 1980. North-Holland.

[10] Andrej Bauer, Lars Birkedal, and Dana S. Scott. Equilogical spaces. To appear in Theoretical Computer Science, 2000.

[11] Anne Sjerp Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1973.

[12] Anne Sjerp Troelstra. Realizability. In Samuel R. Buss, editor, *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 407–473. Elsevier, Amsterdam, 1998.

[13] Georg Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In Arend Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North-Holland, Amsterdam, 1959.

[14] S. C. Kleene. Countable functionals. In Arend Heyting, editor, *Constructivity in Mathematics*, pages 81–100. North-Holland, Amsterdam, 1959.

[15] Roberto M. Amadio and Pierre-Louis Curien. *Domains and Lambda-Calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, England, 1998.

[16] John C. Reynolds. The meaning of types — from intrinsic to extrinsic semantics. Research Series RS–00–32, BRICS, DAIMI, Department of Computer Science, University of Aarhus, December 2000.

[17] John C. Reynolds. An intrinsic semantics of intersection types (abstract of invited lecture). In *Proceedings of the Workshop on Intersection Types and Related Systems*, 2000. The slides for this lecture are available at ftp://ftp.cs.cmu.edu/user/jcr/intertype.ps.gz.