BRICS
Basic Research in Computer Science

BRICS RS-00-32 J. C. Reynolds: The Meaning of Types — From Intrinsic to Extrinsic Semantics

The Meaning of Types

From Intrinsic to Extrinsic Semantics

John C. Reynolds

BRICS Report Series RS-00-32
ISSN 0909-0878 December 2000

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

http://www.brics.dk
ftp://ftp.brics.dk
This document in subdirectory RS/00/32/

The Meaning of Types — From Intrinsic to Extrinsic Semantics *†

John C. Reynolds
Department of Computer Science
Carnegie Mellon University

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 recursion, subtypes, and named products, 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 $\lambda \mathsf{x}.\, \mathsf{x}$ (in a simply typed functional language). Corresponding to the typing judgement $\vdash \lambda \mathsf{x}.\, \mathsf{x} : \mathbf{int} \to \mathbf{int}$, its intrinsic meaning is the identity function on the integers, while corresponding to

*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 (Basic Research in Computer Science, `http://www.brics.dk/`, Centre of the Danish National Research Foundation) in Aarhus, Denmark, September to November 1999 and May to June 2000.

†A shorter and simpler version of this report, in which products and subtyping are omitted and there is only a single primitive type, will appear in "Essays on Programming Methodology", edited by Annabelle McIver and Carroll Morgan (copyright 2001 Springer-Verlag, all rights reserved).

the judgement $\vdash \lambda \mathsf{x}.\, \mathsf{x} : \mathbf{bool} \to \mathbf{bool}$, its intrinsic meaning is the identity function on truth values. On the other hand, $\lambda \mathsf{x}.\, \mathsf{x}\,\mathsf{x}$, 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 $\lambda \mathsf{x}.\, \mathsf{x}$ 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 $\vdash \lambda \mathsf{x}.\, \mathsf{x} : \mathbf{int} \to \mathbf{int}$ asserts that this function maps each integer into an integer, and the judgement $\vdash \lambda \mathsf{x}.\, \mathsf{x} : \mathbf{bool} \to \mathbf{bool}$ 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 report, we will consider the denotational semantics of a typed call-by-name language with several primitive types, functions, named products, subtyping, 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 report, 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 $[\, f \mid \iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n \,]$ 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 $[\, \iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n \,]$.

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:

$$\begin{aligned}&\iota\text{: identifiers}\\&p\text{: phrases}\\&\delta\text{: primitive types}\\&\theta\text{: types}\\&\pi\text{: type assignments.}\end{aligned}$$

Identifiers will be used both as variables and field names. We write I to denote the set of all identifiers.

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

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

We will also have operations for constructing a *record* (or *named tuple*), and for selecting the *field* of a record corresponding to a *field name*:

$$p ::= \langle \iota_1\colon p_1, \ldots, \iota_n\colon p_n\rangle \mid p'.\iota$$

Here, $\langle \iota_1\colon p_1, \ldots, \iota_n\colon p_n\rangle$ is a concrete representation of a phrase that, abstractly, is the function on the set $\{\iota_1, \ldots, \iota_n\}$ of field names that maps each ι_k into the subphrase p_k. This implies that the field names must be distinct, and that permuting the pairs $\iota_k\colon p_k$ does not change the phrase.

In addition, there will be a *fixed-point* expression for defining a value by recursion, and a *conditional* expression that branches on a truth value to choose between evaluating different subexpressions:

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

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

$$\begin{aligned}p ::= {}& 0 \mid 1 \mid 2 \mid \cdots\\ & \mid \mathbf{true} \mid \mathbf{false}\\ & \mid p' + p'' \mid p' \times p'' \mid p' - p'' \mid p' = p'' \mid p' < p'' \mid \neg\, p'\end{aligned}$$

Primitive types, types, and type assignments can also be defined by an abstract grammar:

$$\begin{aligned}\delta &::= \mathbf{int} \mid \mathbf{nat} \mid \mathbf{bool}\\ \theta &::= \delta \mid \theta_1 \to \theta_2 \mid \mathbf{rcd}(\pi)\\ \pi &::= \iota_1\colon \theta_1, \ldots, \iota_n\colon \theta_n\end{aligned}$$

Abstractly, a type assignment, like a record constructor, is a function whose domain is the set $\{\iota_1, \ldots, \iota_n\}$; in this case each identifier ι_k is mapped into the type θ_k. Again, the identifiers must be distinct, and permuting the pairs $\iota_k{:}\,\theta_k$ will not change the type assignment.

Informally, the primitive types **int**, **nat**, and **bool** denote the sets of integers, natural numbers (nonnegative integers), and truth values respectively, $\theta_1 \to \theta_2$ denotes the set of functions that map values of type θ_1 into values of type θ_2, and $\mathbf{rcd}(\iota_1{:}\,\theta_1, \ldots, \iota_n{:}\,\theta_n)$ denotes the set of records with fields named $\iota_1, \ldots, \iota_n$ such that the field named ι_k has a value of type θ_k.

If θ and θ' are types, then the formula $\theta \leq \theta'$ is a *subtype judgement*, which is read "θ is a subtype of θ'". The valid subtype judgements are defined by inference rules (i.e., they are the judgements that can be proved by the use of these rules).

First, we have rules asserting that $\leq$ is a preorder:

$$\frac{}{\theta \leq \theta} \qquad\qquad \frac{\theta \leq \theta'' \qquad \theta'' \leq \theta'}{\theta \leq \theta'.}$$

Next there are rules for the primitive types:

$$\frac{}{\mathbf{nat} \leq \mathbf{int}} \qquad\qquad \frac{}{\mathbf{nat} \leq \mathbf{bool}.}$$

Informally, there is an implicit conversion of natural numbers into integers that is an identity injection, and there is an implicit conversion of natural numbers into truth values that maps zero into false and all positive numbers into true. (We do not recommend this subtyping for the primitive types of a real programming language; we use it in this report to illustrate the variety of implicit conversions that are possible. For instance, the conversion from **nat** to **bool** is not injective.)

For function types, we have

$$\frac{\theta_1' \leq \theta_1 \qquad \theta_2 \leq \theta_2'}{\theta_1 \to \theta_2 \leq \theta_1' \to \theta_2'.}$$

In other words, the type constructor $\to$ is antimonotone in its left argument and monotone in its right argument.

The rule for record types describes an implicit conversion in which fields can be forgotten, and the remaining fields can be implicitly converted:

$$\frac{\theta_1 \leq \theta_1' \quad \cdots \quad \theta_m \leq \theta_m'}{\mathbf{rcd}(\iota_1{:}\,\theta_1, \ldots, \iota_n{:}\,\theta_n) \leq \mathbf{rcd}(\iota_1{:}\,\theta_1', \ldots, \iota_m{:}\,\theta_m')} \qquad \text{when } 0 \leq m \leq n.$$

Notice that, since the pairs $\iota_k{:}\,\theta_k$ can be permuted, $\{\iota_1, \ldots, \iota_m\}$ can be any subset of $\{\iota_1, \ldots, \iota_n\}$.

For example, the following is an (unnecessarily complex) proof of a subtype judgement, written as a tree of inferences:

$$\dfrac{\dfrac{}{\begin{array}{l}\mathbf{rcd}(\mathsf{k}\colon \mathbf{nat}, \mathsf{b}\colon \mathbf{bool}, \mathsf{f}\colon \mathbf{int}\to\mathbf{nat})\\ \quad\le \mathbf{rcd}(\mathsf{f}\colon\mathbf{int}\to\mathbf{nat},\mathsf{k}\colon\mathbf{nat})\end{array}} \qquad \dfrac{\dfrac{\dfrac{}{\mathbf{nat}\le\mathbf{int}} \quad \dfrac{}{\mathbf{nat}\le\mathbf{bool}}}{\mathbf{int}\to\mathbf{nat}\le\mathbf{nat}\to\mathbf{bool}} \quad \dfrac{}{\mathbf{nat}\le\mathbf{int}}}{\begin{array}{l}\mathbf{rcd}(\mathsf{f}\colon\mathbf{int}\to\mathbf{nat},\mathsf{k}\colon\mathbf{nat})\\ \quad\le\mathbf{rcd}(\mathsf{f}\colon\mathbf{nat}\to\mathbf{bool},\mathsf{k}\colon\mathbf{int})\end{array}}}{\begin{array}{l}\mathbf{rcd}(\mathsf{k}\colon \mathbf{nat}, \mathsf{b}\colon \mathbf{bool}, \mathsf{f}\colon \mathbf{int}\to\mathbf{nat})\\ \quad\le\mathbf{rcd}(\mathsf{f}\colon\mathbf{nat}\to\mathbf{bool},\mathsf{k}\colon\mathbf{int}).\end{array}}$$

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.

The *subsumption* rule captures the syntactic essence of subtyping: When θ is a subtype of θ', any phrase of type θ can be used in a context requiring a phrase of type θ':

$$\frac{\pi \vdash p : \theta \qquad \theta \le \theta'}{\pi \vdash p : \theta'.}$$

For the lambda calculus, record operations, fixed-point expressions, and conditional expressions, we have standard inference rules. (In the first two rules, we rely on the fact that type assignments are functions on identifiers.)

$$\frac{}{\pi \vdash \iota : \pi\,\iota} \qquad \text{when } \iota \in \operatorname{dom}\pi$$

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

$$\frac{\pi \vdash p_1 : \theta_1 \quad \cdots \quad \pi \vdash p_n : \theta_n}{\pi \vdash \langle \iota_1\colon p_1, \ldots, \iota_n\colon p_n\rangle : \mathbf{rcd}(\iota_1\colon\theta_1, \ldots, \iota_n\colon\theta_n)}$$

$$\frac{\pi \vdash p' : \mathbf{rcd}(\iota_1\colon\theta_1, \ldots, \iota_n\colon\theta_n)}{\pi \vdash p'.\iota_k : \theta_k} \qquad \text{when } 1 \le k \le n$$

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

$$\frac{\pi \vdash p' : \mathbf{bool} \qquad \pi \vdash p'' : \theta \qquad \pi \vdash p''' : \theta}{\pi \vdash \mathbf{if}\ p'\ \mathbf{then}\ p''\ \mathbf{else}\ p''' : \theta.}$$

There are also a large number of rules for primitive constants and operations, which all have the form

$$\frac{\pi \vdash p_1 : \delta_1 \qquad \cdots \qquad \pi \vdash p_n : \delta_n}{\pi \vdash op(p_1, \ldots, p_n) : \delta.}$$

(To treat all of these rules uniformly, we use prefix form for the primitive operations, and regard constants as zero-ary operations.) For each rule of the above form, we say that $\delta_1, \ldots, \delta_n \to \delta$ is a *signature* of the operator *op*. Then, instead of giving the individual rules explicitly, it is enough to list each operator and its signatures:

Operator	Signatures
$0, 1, 2, \ldots$	$\to \mathbf{nat}$
$\mathbf{true}, \mathbf{false}$	$\to \mathbf{bool}$
$+, \times$	$\mathbf{nat}, \mathbf{nat} \to \mathbf{nat}$
	$\mathbf{int}, \mathbf{int} \to \mathbf{int}$
	$\mathbf{bool}, \mathbf{bool} \to \mathbf{bool}$
$-$	$\mathbf{int}, \mathbf{int} \to \mathbf{int}$
$=, <$	$\mathbf{int}, \mathbf{int} \to \mathbf{bool}$
$\neg$	$\mathbf{bool} \to \mathbf{bool}.$

Here we have "overloaded" $+$ and $\times$ to act on truth values as well as numbers; the intent is that $+$ will act as "or" and $\times$ as "and". At the other extreme, $-$ does not act on truth values, and does not, in general, map natural numbers into natural numbers.

2 An Intrinsic Semantics

To give an intrinsic denotational semantics to our illustrative language, we must define the meanings of types, type assignments, subtype judgements, 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 subtype judgement $\theta \leq \theta'$, a strict continuous function $[\![\theta \leq \theta']\!]$ from $[\![\theta]\!]$ to $[\![\theta']\!]$, called the *implicit conversion from θ to θ'*,
- 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 $\perp$. (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 $\mathbf{Z}$, $\mathbf{N}$, and $\mathbf{B}$ for the sets of integers, natural numbers, and truth values respectively; denumerable sets such as these will be also be regarded as discretely ordered predomains. We write $P_\perp$ for the domain obtained from a predomain P by adding a new least element. 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*

$$\begin{aligned}
[\![\mathbf{int}]\!] &= \mathbf{Z}_\perp \\
[\![\mathbf{nat}]\!] &= \mathbf{N}_\perp \\
[\![\mathbf{bool}]\!] &= \mathbf{B}_\perp \\
[\![\theta_1 \to \theta_2]\!] &= [\![\theta_1]\!] \Rightarrow [\![\theta_2]\!] \\
[\![\mathbf{rcd}(\pi)]\!] &= [\![\pi]\!]^* \\
[\![\iota_1\!:\theta_1, \ldots, \iota_n\!:\theta_n]\!]^* &= \Big\{ [\, \iota_1\!:x_1 \mid \ldots \mid \iota_n\!:x_n \,] \;\Big|\; x_1 \in [\![\theta_1]\!], \ldots, x_n \in [\![\theta_n]\!] \Big\}.
\end{aligned}$$

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

On the other hand, the meanings of subtype and 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 subsumption, we have the semantic equation

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p : \theta) \qquad \mathcal{P}(\theta \leq \theta')}{\pi \vdash p : \theta'} \right]\!\!\right] = [\![\mathcal{P}(\pi \vdash p : \theta)]\!] \, ; [\![\mathcal{P}(\theta \leq \theta')]\!],$$

which asserts that the meaning of a proof of $\pi \vdash p : \theta'$, in which the final inference is an instance of the subsumption rule, is the functional composition of the meanings of its immediate subproofs.

Before proceeding further, we must warn the reader that our illustrative language is rich enough that a judgement may have several significantly different proofs, e.g.,

$$\dfrac{\dfrac{\dfrac{}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{m} : \mathbf{nat}} \qquad \dfrac{}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{n} : \mathbf{nat}}}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{m} \times \mathsf{n} : \mathbf{nat}}}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{m} \times \mathsf{n} : \mathbf{bool}}$$

or

$$\dfrac{\dfrac{\dfrac{}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{m} : \mathbf{nat}}}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{m} : \mathbf{bool}} \qquad \dfrac{\dfrac{}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{n} : \mathbf{nat}}}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{n} : \mathbf{bool}}}{\mathsf{m}{:}\,\mathbf{nat},\mathsf{n}{:}\,\mathbf{nat} \vdash \mathsf{m} \times \mathsf{n} : \mathbf{bool}.}$$

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. However, this fact depends critically on the details of the language design, especially of the semantics of implicit conversions and "overloaded" operators with more than one signature [3],[1, Chapter 16].

The meanings of the subtype judgements are defined by the following semantic equations:

$$\left[\!\!\left[\dfrac{}{\theta \leq \theta}\right]\!\!\right] = I_{[\![\theta]\!]}$$

$$\left[\!\!\left[\dfrac{\mathcal{P}(\theta \leq \theta'') \qquad \mathcal{P}(\theta'' \leq \theta')}{\theta \leq \theta'}\right]\!\!\right] = [\![\mathcal{P}(\theta \leq \theta'')]\!] \,;\, [\![\mathcal{P}(\theta'' \leq \theta')]\!]$$

$$\left[\!\!\left[\dfrac{}{\mathbf{nat} \leq \mathbf{int}}\right]\!\!\right] n = n$$

$$\left[\!\!\left[\dfrac{}{\mathbf{nat} \leq \mathbf{bool}}\right]\!\!\right] n = \begin{cases} \perp & \text{when } n = \perp \\ \mathbf{true} & \text{when } n > 0 \\ \mathbf{false} & \text{when } n = 0 \end{cases}$$

$$\left[\!\!\left[\dfrac{\mathcal{P}(\theta_1' \leq \theta_1) \qquad \mathcal{P}(\theta_2 \leq \theta_2')}{\theta_1 \to \theta_2 \leq \theta_1' \to \theta_2'}\right]\!\!\right] f = [\![\mathcal{P}(\theta_1' \leq \theta_1)]\!] \,;\, f \,;\, [\![\mathcal{P}(\theta_2 \leq \theta_2')]\!]$$

$$\left[\!\!\left[\frac{\mathcal{P}(\theta_1 \le \theta'_1) \quad \cdots \quad \mathcal{P}(\theta_m \le \theta'_m)}{\mathbf{rcd}(\iota_1{:}\,\theta_1, \ldots, \iota_n{:}\,\theta_n) \le \mathbf{rcd}(\iota_1{:}\,\theta'_1, \ldots, \iota_m{:}\,\theta'_m)} \right]\!\!\right] [\, \iota_1{:}\, x_1 \mid \ldots \mid \iota_n{:}\, x_n \,]$$
$$= [\, \iota_1{:}\, [\![\mathcal{P}(\theta_1 \le \theta'_1)]\!] x_1 \mid \ldots \mid \iota_m{:}\, [\![\mathcal{P}(\theta_m \le \theta'_m)]\!] x_m \,].$$

The semantic equations for typing judgements about variables, functions, records, fixed-point expressions, and conditional expressions (as well as the equation for subsumption given earlier) give meanings that are standard for a call-by-name language:

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

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

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

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p_1 : \theta_1) \quad \cdots \quad \mathcal{P}(\pi \vdash p_n : \theta_n)}{\pi \vdash \langle \iota_1{:}\, p_1, \ldots, \iota_n{:}\, p_n \rangle : \mathbf{rcd}(\iota_1{:}\,\theta_1, \ldots, \iota_n{:}\,\theta_n)} \right]\!\!\right] \eta$$
$$= [\, \iota_1{:}\, [\![\mathcal{P}(\pi \vdash p_1{:}\, \theta_1)]\!] \eta \mid \ldots \mid \iota_n{:}\, [\![\mathcal{P}(\pi \vdash p_n{:}\, \theta_n)]\!] \eta \,]$$

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p' : \mathbf{rcd}(\iota_1{:}\,\theta_1, \ldots, \iota_n{:}\,\theta_n))}{\pi \vdash p'.\iota_k : \theta_k} \right]\!\!\right] \eta$$
$$= \big([\![\mathcal{P}(\pi \vdash p' : \mathbf{rcd}(\iota_1{:}\,\theta_1, \ldots, \iota_n{:}\,\theta_n))]\!] \eta \big) \iota_k$$

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

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p' : \mathbf{bool}) \qquad \mathcal{P}(\pi \vdash p'' : \theta) \qquad \mathcal{P}(\pi \vdash p''' : \theta)}{\pi \vdash \mathbf{if}\ p'\ \mathbf{then}\ p''\ \mathbf{else}\ p''' : \theta} \right]\!\!\right] \eta$$
$$= \begin{cases} \perp & \text{when } [\![\mathcal{P}(\pi \vdash p' : \mathbf{bool})]\!] \eta = \perp \\ [\![\mathcal{P}(\pi \vdash p'' : \theta)]\!] \eta & \text{when } [\![\mathcal{P}(\pi \vdash p' : \mathbf{bool})]\!] \eta = \mathbf{true} \\ [\![\mathcal{P}(\pi \vdash p''' : \theta)]\!] \eta & \text{when } [\![\mathcal{P}(\pi \vdash p' : \mathbf{bool})]\!] \eta = \mathbf{false}. \end{cases}$$

The semantic equations for primitive constants and operations have the general form

$$\left[\!\!\left[\frac{\mathcal{P}(\pi \vdash p_1 : \delta_1) \qquad \cdots \qquad \mathcal{P}(\pi \vdash p_n : \delta_n)}{\pi \vdash op(p_1, \ldots, p_n) : \delta} \right]\!\!\right] \eta$$

$$= \mathcal{I}_{op}^{\delta_1,\ldots,\delta_n \to \delta}([\![\mathcal{P}(\pi \vdash p_1 : \delta_1)]\!]\eta, \ldots, [\![\mathcal{P}(\pi \vdash p_n : \delta_n)]\!]\eta),$$

where

$$\mathcal{I}_{op}^{\delta_1,\ldots,\delta_n \to \delta} \in [\![\delta_1]\!] \times \cdots \times [\![\delta_n]\!] \to [\![\delta]\!].$$

Now suppose that $S_1, \ldots, S_n$ are sets, D is a domain, and f is a function from $S_1 \times \cdots \times S_n$ to some subset of D. Then the function f' from $(S_1)_\perp \times \cdots \times (S_n)_\perp$ to D such that $f'\langle x_1, \ldots, x_n \rangle = \perp_D$ when any x_i is $\perp$, and $f'\langle x_1, \ldots, x_n \rangle = f\langle x_1, \ldots, x_n \rangle$ otherwise, is called a *componentwise strict extension* of f. (In the special case where $n = 0$, so that f is a constant function on the singleton domain $\{\langle\rangle\}$, f' is the same as f.)

In particular, the interpretations of the primitive constants and operations are all componentwise strict extensions of standard functions:

The function:	is the componentwise strict extension of:
$\mathcal{I}_0^{\to \mathbf{nat}}$	$\lambda x \in \{\langle\rangle\}.\ 0$
$\mathcal{I}_1^{\to \mathbf{nat}}$	$\lambda x \in \{\langle\rangle\}.\ 1$
$\vdots$	$\vdots$
$\mathcal{I}_{\mathbf{true}}^{\to \mathbf{bool}}$	$\lambda x \in \{\langle\rangle\}.\ \mathbf{true}$
$\mathcal{I}_{\mathbf{false}}^{\to \mathbf{bool}}$	$\lambda x \in \{\langle\rangle\}.\ \mathbf{false}$
$\mathcal{I}_{+}^{\mathbf{nat} \times \mathbf{nat} \to \mathbf{nat}}$	addition of natural numbers
$\mathcal{I}_{+}^{\mathbf{int} \times \mathbf{int} \to \mathbf{int}}$	addition of integers
$\mathcal{I}_{+}^{\mathbf{bool} \times \mathbf{bool} \to \mathbf{bool}}$	disjunction of truth values
$\mathcal{I}_{\times}^{\mathbf{nat} \times \mathbf{nat} \to \mathbf{nat}}$	multiplication of natural numbers
$\mathcal{I}_{\times}^{\mathbf{int} \times \mathbf{int} \to \mathbf{int}}$	multiplication of integers
$\mathcal{I}_{\times}^{\mathbf{bool} \times \mathbf{bool} \to \mathbf{bool}}$	conjunction of truth values
$\mathcal{I}_{-}^{\mathbf{int} \times \mathbf{int} \to \mathbf{int}}$	subtraction of integers
$\mathcal{I}_{=}^{\mathbf{int} \times \mathbf{int} \to \mathbf{bool}}$	equality of integers
$\mathcal{I}_{<}^{\mathbf{int} \times \mathbf{int} \to \mathbf{bool}}$	ordering of integers
$\mathcal{I}_{\neg}^{\mathbf{bool} \to \mathbf{bool}}$	negation of truth values.

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 (i.e., identifiers) 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 $\mathbf{Z}_\perp$ of integers (viewed as primitive values), $U \Rightarrow U$ of continuous functions (viewed as functional values), and E of environments (viewed as record values) to be embeddable in U by pairs of continuous functions:

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

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 variables, functions, records, and fixed points:

$$\begin{aligned}
[\![\iota]\!]\,\varepsilon &= \varepsilon\,\iota \\
[\![\lambda\iota.\ p']\!]\,\varepsilon &= \Phi_\mathrm{f}(\lambda y \in U.\ [\![p']\!][\,\varepsilon \mid \iota\colon y\,]) \\
[\![p'\,p'']\!]\,\varepsilon &= \Psi_\mathrm{f}([\![p']\!]\,\varepsilon)([\![p'']\!]\,\varepsilon) \\
[\![\langle\iota_1\colon p_1,\ldots,\iota_n\colon p_n\rangle]\!]\,\varepsilon &= \Phi_\mathrm{r}([\,\lambda\iota \in I.\ \perp \mid \iota_1\colon [\![p_1]\!]\,\varepsilon \mid \ldots \mid \iota_n\colon [\![p_n]\!]\,\varepsilon\,]) \\
[\![p'.\iota]\!]\,\varepsilon &= \Psi_\mathrm{r}([\![p']\!]\,\varepsilon)\,\iota \\
[\![\mathbf{Y}\,p']\!]\,\varepsilon &= \bigsqcup_{n=0}^{\infty} \big(\Psi_\mathrm{f}([\![p']\!]\,\varepsilon)\big)^n \perp.
\end{aligned}$$

(Note that records with a finite number of fields are "represented" by records with an infinite number of fields, almost all of which are $\perp$.)

When we come to conditional expressions, however, we encounter a problem. Since, from the untyped viewpoint, all primitive values are integers, to describe how a conditional expression branches on its first argument we must understand how integers are used to represent truth values. In fact (as we will formalize in the next section), **false** will be represented by zero, **true** will be represented by any positive integer; and no truth value will be represented by any negative integer. This leads to the semantic equation

$$\begin{aligned}&[\![\mathbf{if}\ p'\ \mathbf{then}\ p''\ \mathbf{else}\ p''']\!]\,\varepsilon\\ &\quad= \begin{cases}\perp & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) = \perp\\ [\![p'']\!]\,\varepsilon & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) > 0\\ [\![p''']\!]\,\varepsilon & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) = 0\\ \mathsf{anyval}([\![p']\!]\,\varepsilon, [\![p'']\!]\,\varepsilon, [\![p''']\!]\varepsilon) & \text{when } \Psi_{\mathrm{p}}([\![p']\!]\,\varepsilon) < 0.\end{cases}\end{aligned}$$

Here anyval can be any continuous function from $U \times U \times U$ to U. (Again, we do not want to constrain our untyped semantics more than will be necessary to establish a proper relationship to our intrinsic typed semantics.)

The semantic equations for primitive constants and operations have the general form

$$[\![op(p_1, \ldots, p_n)]\!]\varepsilon = \Phi_{\mathrm{p}}(\mathcal{I}^U_{op}(\Psi_{\mathrm{p}}([\![p_1]\!]\varepsilon), \ldots, \Psi_{\mathrm{p}}([\![p_n]\!]\varepsilon))),$$

where $\mathcal{I}^U_{op} \in (\mathbf{Z}_\perp)^n \to \mathbf{Z}_\perp$. In particular,

The function:	is the componentwise strict extension of:
$\mathcal{I}^U_0$	$\lambda x \in \{\langle\rangle\}.\ 0$
$\mathcal{I}^U_1$	$\lambda x \in \{\langle\rangle\}.\ 1$
$\vdots$	$\vdots$
$\mathcal{I}^U_{\mathbf{true}}$	$\lambda x \in \{\langle\rangle\}.\ \mathsf{trueint}$
$\mathcal{I}^U_{\mathbf{false}}$	$\lambda x \in \{\langle\rangle\}.\ 0$
$\mathcal{I}^U_+$	addition of integers
$\mathcal{I}^U_\times$	multiplication of integers
$\mathcal{I}^U_-$	subtraction of integers
$\mathcal{I}^U_=$	eqint
$\mathcal{I}^U_<$	$\mathsf{lessint}$
$\mathcal{I}^U_\neg$	notint.

As with anyval, we state the necessary properties of $\mathsf{trueint} \in \mathbf{Z}_\perp$, eqint, $\mathsf{lessint} \in \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}_\perp$, and $\mathsf{notint} \in \mathbf{Z} \to \mathbf{Z}_\perp$, without being overspecific:

$$\begin{aligned}
\mathsf{trueint} &> 0 & & \\[1ex]
\mathsf{eqint}(i_1, i_2) &> 0 & &\text{when } i_1 = i_2 \\
\mathsf{eqint}(i_1, i_2) &= 0 & &\text{when } i_1 \neq i_2 \\[1ex]
\mathsf{lessint}(i_1, i_2) &> 0 & &\text{when } i_1 < i_2 \\
\mathsf{lessint}(i_1, i_2) &= 0 & &\text{when } i_1 \geq i_2 \\[1ex]
\mathsf{notint}(i_1) &> 0 & &\text{when } i_1 = 0 \\
\mathsf{notint}(i_1) &= 0 & &\text{when } i_1 > 0.
\end{aligned}$$

(Note that $z > 0$ does not hold when $z = \perp$.)

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 [4] 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 relations $\rho[\theta]$ will be defined by induction on the structure of the type θ. To sequester the effects of our particular choice of primitive types, however, it is useful first to define *primitive logical relations*:

Definition 4.1 *For primitive types δ, the* primitive logical relations $\widetilde{\rho}\,[\delta] \subseteq [\![\delta]\!] \times \mathbf{Z}_\perp$ *are such that:*

$$\begin{aligned}
\langle x, z\rangle \in \widetilde{\rho}\,[\mathbf{int}] \;\; &\textit{iff} \;\; x = z \\[1ex]
\langle n, z\rangle \in \widetilde{\rho}\,[\mathbf{nat}] \;\; &\textit{iff} \;\; n = z \\[1ex]
\langle b, z\rangle \in \widetilde{\rho}\,[\mathbf{bool}] \;\; &\textit{iff} \;\; (b = \perp \textit{ and } z = \perp) \textit{ or } (b = \mathbf{true} \textit{ and } z > 0) \\
&\qquad\qquad \textit{or } (b = \mathbf{false} \textit{ and } z = 0).
\end{aligned}$$

Then:

Definition 4.2 *For types θ, the* logical relations $\rho[\theta] \subseteq [\![\theta]\!] \times U$ *are such that:*

$$\langle x, y \rangle \in \rho[\delta] \;\; \textit{iff} \;\; \langle x, \Psi_{\mathrm{p}} y \rangle \in \widetilde{\rho}\,[\delta]$$

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

$$\langle [\,\iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n\,], y \rangle \in \rho[\mathbf{rcd}(\iota_1 \colon \theta_1, \ldots, \iota_n \colon \theta_n)]$$
$$\textit{iff} \;\; \langle x_1, \Psi_{\mathrm{r}}\,y\,\iota_1 \rangle \in \rho[\theta_1] \;\; \textit{and} \;\; \cdots \;\; \textit{and} \, \langle x_n, \Psi_{\mathrm{r}}\,y\,\iota_n \rangle \in \rho[\theta_n].$$

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

Definition 4.3 *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 $\langle x_i, y_i \rangle \in r$,*

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

Lemma 4.4 *For all primitive types δ, the primitive logical relations $\widetilde{\rho}\,[\delta]$ are strict and chain-complete.*

Proof. By the definition (4.1) of the $\widetilde{\rho}\,[\delta]$, it is immediate that $\langle \bot, \bot \rangle \in \widetilde{\rho}\,[\delta]$.

Suppose $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$ and $z_0 \sqsubseteq z_1 \sqsubseteq \cdots$ are increasing sequences, in $[\![\delta]\!]$ and $\mathbf{Z}_\bot$ respectively, such that each $\langle x_i, z_i \rangle \in \widetilde{\rho}\,[\delta]$. Since $[\![\delta]\!]$ and $\mathbf{Z}_\bot$ are both flat domains, there is a sufficiently large n that $\bigsqcup_{i=0}^{\infty} x_i = x_n$ and $\bigsqcup_{i=0}^{\infty} z_i = z_n$. Then $\langle \bigsqcup_{i=0}^{\infty} x_i, \bigsqcup_{i=0}^{\infty} z_i \rangle = \langle x_n, z_n \rangle \in \widetilde{\rho}\,[\delta]$. END OF PROOF

Lemma 4.5 *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 a primitive type δ. Then $\langle x, y \rangle \in \rho[\delta]$ iff $\langle x, \Psi_{\mathrm{p}} y \rangle \in \widetilde{\rho}\,[\delta]$. Since Ψ_{p} is a strict function and $\widetilde{\rho}\,[\delta]$ is a strict relation, $\langle \bot, \Psi_{\mathrm{p}}\bot \rangle = \langle \bot, \bot \rangle \in \widetilde{\rho}\,[\delta]$, so that $\langle \bot, \bot \rangle \in \rho[\delta]$.

  Now suppose that x_i and y_i are increasing sequences, in $[\![\delta]\!]$ and U respectively, such that $\langle x_i, y_i \rangle \in \rho[\delta]$. Then $\langle x_i, \Psi_{\mathrm{p}} y_i \rangle \in \widetilde{\rho}\,[\delta]$ for $i \geq 0$, and since Ψ_{p} is monotone, the $\Psi_{\mathrm{p}} y_i$ are an increasing sequence in $Z_\bot$.

Then, since Ψ_{p} is continuous, and $\widetilde{\rho}\,[\delta]$ is chain-complete,

$$\langle \bigsqcup_{i=0}^{\infty} x_i, \Psi_{\mathrm{p}}\big(\bigsqcup_{i=0}^{\infty} y_i\big)\rangle = \langle \bigsqcup_{i=0}^{\infty} x_i, \bigsqcup_{i=0}^{\infty} \Psi_{\mathrm{p}} y_i\rangle \in \widetilde{\rho}\,[\delta],$$

so that $\langle \bigsqcup_{i=0}^{\infty} x_i, \bigsqcup_{i=0}^{\infty} y_i\rangle \in \rho[\delta]$.

- 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 $\langle x, y\rangle \in \rho[\theta_1]$, $\langle \bigsqcup_{i=0}^{\infty} f_i, \bigsqcup_{i=0}^{\infty} g_i\rangle \in \rho[\theta_1 \to \theta_2]$.

- Suppose θ is $\mathbf{rcd}(\iota_1\colon \theta_1, \ldots, \iota_n\colon \theta_n)$. Let $[\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,] = \bot$ and $y = \bot$. Then, for each k between one and n, since products are ordered componentwise, $x_k = \bot$, and since Ψ_{r} is strict, $\Psi_{\mathrm{r}}\, y\, \iota_k = \bot$, so that the induction hypothesis for θ_k gives $\langle x_k, \Psi_{\mathrm{r}}\, y\, \iota_k\rangle = \langle \bot, \bot\rangle \in \rho[\theta_k]$. Then, since this holds for all k, we have $\langle [\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,], y\rangle = \langle \bot, \bot\rangle \in \rho[\mathbf{rcd}(\iota_1\colon \theta_1, \ldots, \iota_n\colon \theta_n)]$.

  Now suppose that $[\,\iota_1\colon x_{1,i} \mid \ldots \mid \iota_n\colon x_{n,i}\,]$ and y_i are increasing sequences (in i) in $[\![\mathbf{rcd}(\iota_1\colon \theta_1, \ldots, \iota_n\colon \theta_n)]\!]$ and U respectively, such that $\langle [\,\iota_1\colon x_{1,i} \mid \ldots \mid \iota_n\colon x_{n,i}\,], y_i\rangle \in \rho[\mathbf{rcd}(\iota_1\colon \theta_1, \ldots, \iota_n\colon \theta_n)]$. Let k be between one and n. Then $\langle x_{k,i}, \Psi_{\mathrm{r}}\, y_i\, \iota_k\rangle \in \rho[\theta_k]$ for $i \geq 0$, and since component selection, Ψ_{r}, and function application are monotone, $x_{k,i}$ and $\Psi_{\mathrm{r}}\, y_i\, \iota_k$ are increasing sequences. Then, since Ψ_{r} is continuous, and a least upper bound of functions distributes through application, the induction hypothesis for θ_k gives

  $$\langle \bigsqcup_{i=0}^{\infty} x_{k,i}, \Psi_{\mathrm{r}}(\bigsqcup_{i=0}^{\infty} y_i)\iota_k\rangle = \langle \bigsqcup_{i=0}^{\infty} x_{k,i}, \bigsqcup_{i=0}^{\infty} \Psi_{\mathrm{r}}\, y_i\, \iota_k\rangle \in \rho[\theta_k].$$

  Finally, since this holds for all k, and since least upper bounds of products are taken componentwise,

  $$\begin{aligned} &\langle \bigsqcup_{i=0}^{\infty} [\,\iota_1\colon x_{1,i} \mid \ldots \mid \iota_n\colon x_{n,i}\,], \bigsqcup_{i=0}^{\infty} y_i\rangle \\ &\quad = \langle [\,\iota_1\colon \bigsqcup_{i=0}^{\infty} x_{1,i} \mid \ldots \mid \iota_n\colon \bigsqcup_{i=0}^{\infty} x_{n,i}\,], \bigsqcup_{i=0}^{\infty} y_i\rangle \\ &\quad \in \rho[\mathbf{rcd}(\iota_1\colon \theta_1, \ldots, \iota_n\colon \theta_n)]. \end{aligned}$$

END OF PROOF

Next, we establish the connection between subtyping and the logical relations:

Theorem 4.6 *If $\mathcal{P}(\theta \leq \theta')$ is a proof of the subtype judgement $\theta \leq \theta'$, and $\langle x, y\rangle \in \rho[\theta]$, then $\langle [\![\mathcal{P}(\theta \leq \theta')]\!]x, y\rangle \in \rho[\theta']$.*

Proof. By induction on the structure of the proof $\mathcal{P}(\theta \leq \theta')$.

- Suppose $\mathcal{P}(\theta \leq \theta')$ is
$$\frac{}{\theta \leq \theta,}$$
so that $[\![\mathcal{P}(\theta \leq \theta')]\!] = I_{[\![\theta]\!]}$. If $\langle x, y\rangle \in \rho[\theta]$, then $\langle [\![\mathcal{P}(\theta \leq \theta')]\!]x, y\rangle = \langle x, y\rangle \in \rho[\theta] = \rho[\theta']$.

- Suppose $\mathcal{P}(\theta \leq \theta')$ is
$$\frac{\mathcal{P}(\theta \leq \theta'') \qquad \mathcal{P}(\theta'' \leq \theta')}{\theta \leq \theta',}$$
so that $[\![\mathcal{P}(\theta \leq \theta')]\!] = [\![\mathcal{P}(\theta \leq \theta'')]\!] \mathbin{;} [\![\mathcal{P}(\theta'' \leq \theta')]\!]$. If $\langle x, y\rangle \in \rho[\theta]$, then $\langle [\![\mathcal{P}(\theta \leq \theta'')]\!]x, y\rangle \in \rho[\theta'']$ by the induction hypothesis for $\mathcal{P}(\theta \leq \theta'')$, and then $\langle [\![\mathcal{P}(\theta \leq \theta')]\!]x, y\rangle = \langle [\![\mathcal{P}(\theta'' \leq \theta')]\!]([\![\mathcal{P}(\theta \leq \theta'')]\!]x), y\rangle \in \rho[\theta']$ by the induction hypothesis for $\mathcal{P}(\theta'' \leq \theta')$.

- Suppose $\mathcal{P}(\theta \leq \theta')$ is
$$\frac{}{\mathbf{nat} \leq \mathbf{int},}$$
so that $[\![\mathcal{P}(\theta \leq \theta')]\!]n = n$. If $\langle n, y\rangle \in \rho[\mathbf{nat}]$, then $n = \Psi_{\mathrm{p}}y$, so that
$$\langle [\![\mathcal{P}(\theta \leq \theta')]\!]n, \Psi_{\mathrm{p}}y\rangle = \langle n, \Psi_{\mathrm{p}}y\rangle = \langle \Psi_{\mathrm{p}}y, \Psi_{\mathrm{p}}y\rangle,$$
which satisfies the definition (4.1) of a member of $\widetilde{\rho}\,[\mathbf{int}]$. It follows that $\langle [\![\mathcal{P}(\theta \leq \theta')]\!]n, y\rangle \in \rho[\mathbf{int}]$.

- Suppose $\mathcal{P}(\theta \leq \theta')$ is
$$\frac{}{\mathbf{nat} \leq \mathbf{bool},}$$
so that
$$[\![\mathcal{P}(\theta \leq \theta')]\!]n = \begin{cases} \perp & \text{when } n = \perp \\ \mathbf{true} & \text{when } n > 0 \\ \mathbf{false} & \text{when } n = 0. \end{cases}$$
If $\langle n, y\rangle \in \rho[\mathbf{nat}]$, then $n = \Psi_{\mathrm{p}}y$, so that the equation displayed above implies that
$$\langle [\![\mathcal{P}(\theta \leq \theta')]\!]n, \Psi_{\mathrm{p}}y\rangle = \begin{cases} \langle \perp, \Psi_{\mathrm{p}}y\rangle & \text{when } \Psi_{\mathrm{p}}y = \perp \\ \langle \mathbf{true}, \Psi_{\mathrm{p}}y\rangle & \text{when } \Psi_{\mathrm{p}}y > 0 \\ \langle \mathbf{false}, \Psi_{\mathrm{p}}y\rangle & \text{when } \Psi_{\mathrm{p}}y = 0, \end{cases}$$
which satisfies the definition (4.1) of a member of $\widetilde{\rho}\,[\mathbf{bool}]$. It follows that $\langle [\![\mathcal{P}(\theta \leq \theta')]\!]n, y\rangle \in \rho[\mathbf{bool}]$.

- Suppose $\mathcal{P}(\theta \leq \theta')$ is

$$\frac{\mathcal{P}(\theta_1' \leq \theta_1) \qquad \mathcal{P}(\theta_2 \leq \theta_2')}{\theta_1 \to \theta_2 \leq \theta_1' \to \theta_2',}$$

so that

$$[\![\mathcal{P}(\theta \leq \theta')]\!] f = [\![\mathcal{P}(\theta_1' \leq \theta_1)]\!] \,;\, f \,;\, [\![\mathcal{P}(\theta_2 \leq \theta_2')]\!].$$

Suppose $\langle f, g\rangle \in \rho[\theta_1 \to \theta_2]$, and let $\langle x, y\rangle \in \rho[\theta_1']$. By the induction hypothesis for $\mathcal{P}(\theta_1' \leq \theta_1)$,

$$\langle [\![\mathcal{P}(\theta_1' \leq \theta_1)]\!] x, y\rangle \in \rho[\theta_1],$$

and since $\langle f, g\rangle \in \rho[\theta_1 \to \theta_2]$,

$$\langle f([\![\mathcal{P}(\theta_1' \leq \theta_1)]\!] x), \Psi_{\mathrm{f}}\, g\, y\rangle \in \rho[\theta_2].$$

Then, by the induction hypothesis for $\mathcal{P}(\theta_2 \leq \theta_2')$,

$$\langle [\![\mathcal{P}(\theta_2 \leq \theta_2')]\!](f([\![\mathcal{P}(\theta_1' \leq \theta_1)]\!] x)), \Psi_{\mathrm{f}}\, g\, y\rangle \in \rho[\theta_2'],$$

and since this holds for all $\langle x, y\rangle \in \rho[\theta_1']$,

$$\langle [\![\mathcal{P}(\theta \leq \theta')]\!] f, g\rangle = \langle [\![\mathcal{P}(\theta_1' \leq \theta_1)]\!] \,;\, f \,;\, [\![\mathcal{P}(\theta_2 \leq \theta_2')]\!], g\rangle \in \rho[\theta_1' \to \theta_2'].$$

- Suppose $\mathcal{P}(\theta \leq \theta')$ is

$$\frac{\mathcal{P}(\theta_1 \leq \theta_1') \quad \cdots \quad \mathcal{P}(\theta_m \leq \theta_m')}{\mathbf{rcd}(\iota_1{:}\, \theta_1, \ldots, \iota_n{:}\, \theta_n) \leq \mathbf{rcd}(\iota_1{:}\, \theta_1', \ldots, \iota_m{:}\, \theta_m'),}$$

so that

$$\begin{aligned}&[\![\mathcal{P}(\theta \leq \theta')]\!][\,\iota_1{:}\, x_1 \mid \ldots \mid \iota_n{:}\, x_n\,] \\ &\quad = [\,\iota_1{:}\, [\![\mathcal{P}(\theta_1 \leq \theta_1')]\!] x_1 \mid \ldots \mid \iota_m{:}\, [\![\mathcal{P}(\theta_m \leq \theta_m')]\!] x_m\,].\end{aligned}$$

Suppose $\langle [\,\iota_1{:}\, x_1 \mid \ldots \mid \iota_n{:}\, x_n\,], y\rangle \in \rho[\mathbf{rcd}(\iota_1{:}\, \theta_1, \ldots, \iota_n{:}\, \theta_n)]$. Then each $\langle x_k, \Psi_{\mathrm{r}}\, y\, \iota_k\rangle \in \rho[\theta_k]$, so that, for $1 \leq k \leq m$, the induction hypothesis for $\mathcal{P}(\theta_k \leq \theta_k')$ gives

$$\langle [\![\mathcal{P}(\theta_k \leq \theta_k')]\!] x_k, \Psi_{\mathrm{r}}\, y\, \iota_k\rangle \in \rho[\theta_k'].$$

Since this holds for each $k \leq m$, we have

$$\begin{aligned}&\langle [\![\mathcal{P}(\theta \leq \theta')]\!][\,\iota_1{:}\, x_1 \mid \ldots \mid \iota_n{:}\, x_n\,], y\rangle \\ &\qquad = \langle [\,\iota_1{:}\, [\![\mathcal{P}(\theta_1 \leq \theta_1')]\!] x_1 \mid \ldots \mid \iota_m{:}\, [\![\mathcal{P}(\theta_m \leq \theta_m')]\!] x_m\,], y\rangle \\ &\qquad \in \rho[\mathbf{rcd}(\iota_1{:}\, \theta_1', \ldots, \iota_m{:}\, \theta_m')].\end{aligned}$$

END OF PROOF

Next, we establish that, for any signature of any primitive operator, applying the typed and untyped interpretations of the operator to related arguments will yield related results, where the type-dependent notion of "related" is given by the primitive logical relations:

Lemma 4.7 *Suppose $\delta_1, \ldots, \delta_n \to \delta$ is a signature of op, and $\langle x_1, z_1\rangle \in \widetilde{\rho}\,[\delta_1]$, $\ldots$, $\langle x_n, z_n\rangle \in \widetilde{\rho}\,[\delta_n]$. Then*

$$\langle \mathcal{I}_{op}^{\delta_1,\ldots,\delta_n\to\delta}(x_1, \ldots, x_n), \mathcal{I}_{op}^{U}(z_1, \ldots, z_n)\rangle \in \widetilde{\rho}\,[\delta].$$

The proof is a case analysis over each signature of each operator; we leave the tedious details to the reader. The only nontriviality is the connection between arithmetic and boolean operations that justifies the cases where $+$ and $\times$ have the signature **bool**, **bool** $\to$ **bool**: Suppose $z_1 \geq 0$ and $z_2 \geq 0$; then $z_1 + z_2 > 0$ holds iff $z_1 > 0$ or $z_2 > 0$, and $z_1 \times z_2 > 0$ holds iff $z_1 > 0$ and $z_2 > 0$.

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

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

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 that the final inference of $\mathcal{P}(\pi \vdash p : \theta)$ is an instance of the subsumption rule, i.e., $\mathcal{P}(\pi \vdash p : \theta)$ is

$$\frac{\mathcal{P}(\pi \vdash p : \theta') \qquad \mathcal{P}(\theta' \leq \theta)}{\pi \vdash p : \theta,}$$

so that

$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!] = [\![\mathcal{P}(\pi \vdash p : \theta')]\!] \mathbin{;} [\![\mathcal{P}(\theta' \leq \theta)]\!].$$

(We have renamed the metavariables θ and θ' by interchanging them.) Assume (A). By the induction hypothesis for $\mathcal{P}(\pi \vdash p : \theta')$, we have

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

Then,

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

by Theorem 4.6.

- 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

$$\forall \iota' \in \mathrm{dom}[\, \pi \mid \iota{:}\, \theta_1 \,].\ [\, \eta \mid \iota{:}\, x \,]\, \iota' \in [\![[\, \pi \mid \iota{:}\, \theta_1 \,]\, \iota']\!]$$

$$[\, \varepsilon \mid \iota{:}\, y \,] \in I \Rightarrow U \tag{A$'$}$$

$$\forall \iota' \in \mathrm{dom}[\, \pi \mid \iota{:}\, \theta_1 \,].\ \langle [\, \eta \mid \iota{:}\, x \,]\, \iota', [\, \varepsilon \mid \iota{:}\, y \,]\, \iota' \rangle \in \rho[[\, \pi \mid \iota{:}\, \theta_1 \,]\, \iota'],$$

so that

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

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 (4.2) 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

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

by the definition (4.2) of $\rho[\theta_1 \to \theta_2]$.

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

$$\frac{\mathcal{P}(\pi \vdash p_1 : \theta_1) \quad \cdots \quad \mathcal{P}(\pi \vdash p_n : \theta_n)}{\pi \vdash \langle \iota_1{:}\,p_1, \ldots, \iota_n{:}\,p_n\rangle : \mathbf{rcd}(\iota_1{:}\,\theta_1, \ldots, \iota_n{:}\,\theta_n),}$$

so that

$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = [\,\iota_1{:}\,[\![\mathcal{P}(\pi \vdash p_1 : \theta_1)]\!]\eta \mid \ldots \mid \iota_n{:}\,[\![\mathcal{P}(\pi \vdash p_n : \theta_n)]\!]\eta\,]$$

$$[\![p]\!]\varepsilon = \Phi_{\mathrm{r}}([\,\lambda\iota \in I.\,\bot \mid \iota_1{:}\,[\![p_1]\!]\,\varepsilon \mid \ldots \mid \iota_n{:}\,[\![p_n]\!]\,\varepsilon\,]).$$

Assume (A). Then, for each k between one and n,

$$\begin{aligned}
&\langle [\![\mathcal{P}(\pi \vdash p_k : \theta_k)]\!]\eta, \Psi_{\mathrm{r}}(\Phi_{\mathrm{r}}([\,\lambda\iota \in I.\,\bot \mid \iota_1{:}\,[\![p_1]\!]\,\varepsilon \mid \ldots \mid \iota_n{:}\,[\![p_n]\!]\,\varepsilon\,]))\,\iota_k\rangle \\
&\qquad = \langle [\![\mathcal{P}(\pi \vdash p_k : \theta_k)]\!]\eta, [\,\lambda\iota \in I.\,\bot \mid \iota_1{:}\,[\![p_1]\!]\,\varepsilon \mid \ldots \mid \iota_n{:}\,[\![p_n]\!]\,\varepsilon\,]\,\iota_k\rangle \\
&\qquad = \langle [\![\mathcal{P}(\pi \vdash p_k : \theta_k)]\!]\eta, [\![p_k]\!]\varepsilon\rangle \\
&\qquad \in \rho[\theta_k],
\end{aligned}$$

where the last step follows from the induction hypothesis for $\mathcal{P}(\pi \vdash p_k : \theta_k)$. Thus the definition (4.2) of $\rho[\mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n)]$ gives

$$\begin{aligned}
&\langle[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \\
&\qquad = \langle[\,\iota_1\colon[\![\mathcal{P}(\pi \vdash p_1 : \theta_1)]\!]\eta \mid \ldots \mid \iota_n\colon[\![\mathcal{P}(\pi \vdash p_n : \theta_n)]\!]\eta\,], \\
&\qquad\qquad \Phi_{\mathrm{r}}([\,\lambda\iota \in I.\,\bot \mid \iota_1\colon[\![p_1]\!]\,\varepsilon \mid \ldots \mid \iota_n\colon[\![p_n]\!]\,\varepsilon\,])\rangle \\
&\qquad \in \rho[\mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n)].
\end{aligned}$$

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

$$\frac{\mathcal{P}(\pi \vdash p' : \mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n))}{\pi \vdash p'.\iota_k : \theta_k,}$$

so that

$$\begin{aligned}
[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta &= \big([\![\mathcal{P}(\pi \vdash p' : \mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n))]\!]\eta\big)\iota_k \\
[\![p]\!]\varepsilon &= \Psi_{\mathrm{r}}([\![p']\!]\,\varepsilon)\,\iota_k.
\end{aligned}$$

Assume (A). Then the induction hypothesis gives

$$\langle[\![\mathcal{P}(\pi \vdash p'\colon\mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n))]\!]\eta, [\![p']\!]\varepsilon\rangle \in \rho[\mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n)],$$

so that

$$\begin{aligned}
&\langle[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon\rangle \\
&\qquad = \langle([\![\mathcal{P}(\pi \vdash p' : \mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n))]\!]\eta)\iota_k, \Psi_{\mathrm{r}}([\![p']\!]\,\varepsilon)\,\iota_k\rangle \\
&\qquad \in \rho[\theta_k],
\end{aligned}$$

by the definition (4.2) of $\rho[\mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n)]$.

- 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

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

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

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

The case for $n = 0$ follows since, by Lemma 4.5, $\rho[\theta]$ is strict. For the induction step, we have

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

by the induction hypothesis for n and the definition (4.2) of $\rho[\theta\to\theta]$.

Finally,

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

since $\rho[\theta]$ is chain-complete by Lemma 4.5.

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

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

so that

$$\llbracket\mathcal{P}(\pi\vdash p{:}\,\theta)\rrbracket\eta = \begin{cases}\bot & \text{when } \llbracket\mathcal{P}(\pi\vdash p'{:}\,\mathbf{bool})\rrbracket\eta = \bot\\ \llbracket\mathcal{P}(\pi\vdash p''{:}\,\theta)\rrbracket\eta & \text{when } \llbracket\mathcal{P}(\pi\vdash p'{:}\,\mathbf{bool})\rrbracket\eta = \mathbf{true}\\ \llbracket\mathcal{P}(\pi\vdash p'''{:}\,\theta)\rrbracket\eta & \text{when } \llbracket\mathcal{P}(\pi\vdash p'{:}\,\mathbf{bool})\rrbracket\eta = \mathbf{false}\end{cases}$$

$$\llbracket p\rrbracket\varepsilon = \begin{cases}\bot & \text{when } \Psi_{\mathrm{p}}(\llbracket p'\rrbracket\,\varepsilon) = \bot\\ \llbracket p''\rrbracket\,\varepsilon & \text{when } \Psi_{\mathrm{p}}(\llbracket p'\rrbracket\,\varepsilon) > 0\\ \llbracket p'''\rrbracket\,\varepsilon & \text{when } \Psi_{\mathrm{p}}(\llbracket p'\rrbracket\,\varepsilon) = 0\\ \mathsf{anyval}(\llbracket p'\rrbracket\,\varepsilon, \llbracket p''\rrbracket\,\varepsilon, \llbracket p'''\rrbracket\,\varepsilon) & \text{when } \Psi_{\mathrm{p}}(\llbracket p'\rrbracket\,\varepsilon) < 0.\end{cases}$$

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

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

so that

$$\langle\llbracket\mathcal{P}(\pi\vdash p' : \mathbf{bool})\rrbracket\eta, \Psi_{\mathrm{p}}(\llbracket p'\rrbracket\varepsilon)\rangle \in \widetilde{\rho}\,[\mathbf{bool}].$$

By the definition (4.1) of $\widetilde{\rho}\,[\mathbf{bool}]$, there are three cases:

- $[\![\mathcal{P}(\pi \vdash p' : \mathbf{bool})]\!]\eta = \bot$ and $\Psi_{\mathrm{p}}[\![p']\!]\varepsilon = \bot$. Then
$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon \rangle = \langle \bot, \bot \rangle,$$
which belongs to $\rho[\theta]$ since $\rho[\theta]$ is strict.
- $[\![\mathcal{P}(\pi \vdash p' : \mathbf{bool})]\!]\eta = \mathbf{true}$ and $\Psi_{\mathrm{p}}[\![p']\!]\varepsilon > 0$. Then
$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon \rangle = \langle [\![\mathcal{P}(\pi \vdash p'' : \theta)]\!]\eta, [\![p'']\!]\varepsilon \rangle,$$
which belongs to $\rho[\theta]$ by the induction hypothesis for $\mathcal{P}(\pi \vdash p'' : \theta)$.
- $[\![\mathcal{P}(\pi \vdash p' : \mathbf{bool})]\!]\eta = \mathbf{false}$ and $\Psi_{\mathrm{p}}[\![p']\!]\varepsilon = 0$. Then
$$\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon \rangle = \langle [\![\mathcal{P}(\pi \vdash p''' : \theta)]\!]\eta, [\![p''']\!]\varepsilon \rangle,$$
which belongs to $\rho[\theta]$ by the induction hypothesis for $\mathcal{P}(\pi \vdash p''' : \theta)$.

- Suppose $\mathcal{P}(\pi \vdash p : \theta)$ is
$$\frac{\mathcal{P}(\pi \vdash p_1 : \delta_1) \qquad \cdots \qquad \mathcal{P}(\pi \vdash p_n : \delta_n)}{\pi \vdash op(p_1, \ldots, p_n) : \delta,}$$
so that
$$[\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta = \mathcal{I}_{op}^{\delta_1,\ldots,\delta_n \to \delta}([\![\mathcal{P}(\pi \vdash p_1 : \delta_1)]\!]\eta, \ldots, [\![\mathcal{P}(\pi \vdash p_n : \delta_n)]\!]\eta)$$
$$[\![p]\!]\varepsilon = \Phi_{\mathrm{p}}(\mathcal{I}_{op}^{U}(\Psi_{\mathrm{p}}([\![p_1]\!]\varepsilon), \ldots, \Psi_{\mathrm{p}}([\![p_n]\!]\varepsilon))).$$
Assume (A). By the induction hypothesis, we have, for $1 \le k \le n$,
$$\langle [\![\mathcal{P}(\pi \vdash p_k : \delta_k)]\!]\eta, [\![p_k]\!]\varepsilon \rangle \in \rho[\delta_k],$$
so that
$$\langle [\![\mathcal{P}(\pi \vdash p_k : \delta_k)]\!]\eta, \Psi_{\mathrm{p}}([\![p_k]\!]\varepsilon) \rangle \in \widetilde{\rho}\,[\delta_k].$$
Then
$$\begin{aligned}
&\langle \mathcal{I}_{op}^{\delta_1,\ldots,\delta_n \to \delta}([\![\mathcal{P}(\pi \vdash p_1 : \delta_1)]\!]\eta, \ldots, [\![\mathcal{P}(\pi \vdash p_n : \delta_n)]\!]\eta),\\
&\qquad \Psi_{\mathrm{p}}(\Phi_{\mathrm{p}}(\mathcal{I}_{op}^{U}(\Psi_{\mathrm{p}}([\![p_1]\!]\varepsilon), \ldots \Psi_{\mathrm{p}}([\![p_n]\!]\varepsilon)))) \rangle\\
&\quad = \langle \mathcal{I}_{op}^{\delta_1,\ldots,\delta_n \to \delta}([\![\mathcal{P}(\pi \vdash p_1 : \delta_1)]\!]\eta, \ldots, [\![\mathcal{P}(\pi \vdash p_n : \delta_n)]\!]\eta),\\
&\qquad \mathcal{I}_{op}^{U}(\Psi_{\mathrm{p}}([\![p_1]\!]\varepsilon), \ldots \Psi_{\mathrm{p}}([\![p_n]\!]\varepsilon)) \rangle\\
&\quad \in \widetilde{\rho}\,[\delta],
\end{aligned}$$
by Lemma 4.7, so that
$$\begin{aligned}
&\langle [\![\mathcal{P}(\pi \vdash p : \theta)]\!]\eta, [\![p]\!]\varepsilon \rangle\\
&\quad = \langle \mathcal{I}_{op}^{\delta_1,\ldots,\delta_n \to \delta}([\![\mathcal{P}(\pi \vdash p_1 : \delta_1)]\!]\eta, \ldots, [\![\mathcal{P}(\pi \vdash p_n : \delta_n)]\!]\eta),\\
&\qquad \Phi_{\mathrm{p}}(\mathcal{I}_{op}^{U}(\Psi_{\mathrm{p}}([\![p_1]\!]\varepsilon), \ldots \Psi_{\mathrm{p}}([\![p_n]\!]\varepsilon))) \rangle\\
&\quad \in \rho[\delta].
\end{aligned}$$

END OF PROOF

Finally, we can show that the logical relations possess a property that is plausibly called *convexity*:

Definition 4.9 *A relation r between a set and a domain is said to be* convex *iff $\langle x, y' \rangle \in r$ holds whenever $\langle x, y \rangle$, $\langle x, y'' \rangle \in r$ and $y \sqsubseteq y' \sqsubseteq y''$.*

Lemma 4.10 *The primitive logical relations $\widetilde{\rho}\,[\delta]$ are convex.*

Proof. For each primitive type δ, the domain $[\![\delta]\!]$ is flat, so that $y \sqsubseteq y' \sqsubseteq y''$ implies that $y' = y$ or $y' = y''$. END OF PROOF

Theorem 4.11 *The logical relations $\rho[\theta]$ are convex.*

Proof. By induction on the structure of θ.

- Suppose θ is a primitive type δ, and that $\langle x, y \rangle$, $\langle x, y'' \rangle \in \rho[\delta]$ and $y \sqsubseteq y' \sqsubseteq y''$. Then $\langle x, \Psi_{\mathrm{p}} y \rangle$, $\langle x, \Psi_{\mathrm{p}} y'' \rangle \in \widetilde{\rho}\,[\delta]$ and, since Ψ_{p} is monotone, $\Psi_{\mathrm{p}} y \sqsubseteq \Psi_{\mathrm{p}} y' \sqsubseteq \Psi_{\mathrm{p}} y''$. Since $\widetilde{\rho}\,[\delta]$ is convex, $\langle x, \Psi_{\mathrm{p}} y' \rangle \in \widetilde{\rho}\,[\delta]$, so that $\langle x, y' \rangle \in \rho[\delta]$.

- Suppose θ is $\theta_1 \to \theta_2$, and that $\langle f, g \rangle$, $\langle f, g'' \rangle \in \rho[\theta_1 \to \theta_2]$ and $g \sqsubseteq g' \sqsubseteq g''$. Let $\langle x, y \rangle \in \rho[\theta_1]$. Then $\langle f\,x, \Psi_{\mathrm{f}}\, g\, y \rangle$, $\langle f\,x, \Psi_{\mathrm{f}}\, g''\, y \rangle \in \rho[\theta_2]$ and, since Ψ_{f} and function application are monotone, $\Psi_{\mathrm{f}}\, g\, y \sqsubseteq \Psi_{\mathrm{f}}\, g'\, y \sqsubseteq \Psi_{\mathrm{f}}\, g''\, y$. Then the induction hypothesis for $\rho[\theta_2]$ gives $\langle f\,x, \Psi_{\mathrm{f}}\, g'\, y \rangle \in \rho[\theta_2]$, and since this holds for all $\langle x, y \rangle \in \rho[\theta_1]$, we have $\langle f, g' \rangle \in \rho[\theta_1 \to \theta_2]$.

- Suppose θ is $\mathbf{rcd}(\iota_1 \colon \theta_1, \ldots, \iota_n \colon \theta_n)$, and that $\langle [\, \iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n \,], y \rangle$, $\langle [\, \iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n \,], y'' \rangle \in \rho[\mathbf{rcd}(\iota_1 \colon \theta_1, \ldots, \iota_n \colon \theta_n)]$ and $y \sqsubseteq y' \sqsubseteq y''$. Then, for each k between one and n, $\langle x_k, \Psi_{\mathrm{r}}\, y\, \iota_k \rangle$, $\langle x_k, \Psi_{\mathrm{r}}\, y''\, \iota_k \rangle \in \rho[\theta_k]$ and, since Ψ_{r} and application to ι_k are monotone, $\Psi_{\mathrm{r}}\, y\, \iota_k \sqsubseteq \Psi_{\mathrm{r}}\, y'\, \iota_k \sqsubseteq \Psi_{\mathrm{r}}\, y''\, \iota_k$. Then the induction hypothesis for $\rho[\theta_k]$ gives $\langle x_k, \Psi_{\mathrm{r}}\, y'\, \iota_k \rangle \in \rho[\theta_k]$, and since this holds for all k, we have $\langle [\, \iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n \,], y' \rangle \in \rho[\mathbf{rcd}(\iota_1 \colon \theta_1, \ldots, \iota_n \colon \theta_n)]$.

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 [5].)

First, however, as with the definition of logical relations, it is useful to define a subsidiary kind of function pair for primitive types:

Definition 5.1 *For primitive types δ, the functions*

$$[\![\delta]\!] \underset{\widetilde{\psi}\,[\delta]}{\overset{\widetilde{\phi}\,[\delta]}{\rightleftarrows}} \mathbf{Z}_\perp$$

are such that

$$\widetilde{\phi}\,[\mathbf{int}]x = x$$

$$\widetilde{\psi}\,[\mathbf{int}]z = z$$

$$\widetilde{\phi}\,[\mathbf{nat}]n = n$$

$$\widetilde{\psi}\,[\mathbf{nat}]z = \begin{cases} \perp & \textit{when}\ z = \perp \\ z & \textit{when}\ z \geq 0 \\ \perp & \textit{when}\ z < 0 \end{cases}$$

$$\widetilde{\phi}\,[\mathbf{bool}]b = \begin{cases} \perp & \textit{when}\ b = \perp \\ 1 & \textit{when}\ b = \mathbf{true} \\ 0 & \textit{when}\ b = \mathbf{false} \end{cases}$$

$$\widetilde{\psi}\,[\mathbf{bool}]z = \begin{cases} \perp & \textit{when}\ z = \perp \\ \mathbf{true} & \textit{when}\ z > 0 \\ \mathbf{false} & \textit{when}\ z = 0 \\ \perp & \textit{when}\ z < 0. \end{cases}$$

¿From this definition, by tedious case analysis, the reader may verify that $\widetilde{\phi}\,[\delta]$ and $\widetilde{\psi}\,[\delta]$ bear the following relationship to the primitive logical relations $\widetilde{\rho}\,[\delta]$:

Lemma 5.2 *For each primitive type δ:*

1. *For all $x \in [\![\delta]\!]$, $\langle x, \widetilde{\phi}\,[\delta]x\rangle \in \widetilde{\rho}\,[\delta]$.*
2. *For all $\langle x, z\rangle \in \widetilde{\rho}\,[\delta]$, $x = \widetilde{\psi}\,[\delta]z$.*

Now we can define the function pairs $\phi[\theta], \psi[\theta]$ by induction on the structure of θ:

Definition 5.3 *For types θ, the functions*

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

are such that

$$\phi[\delta]x = \Phi_{\mathrm{p}}(\widetilde{\phi}\,[\delta]x)$$

$$\psi[\delta]y = \widetilde{\psi}\,[\delta](\Psi_{\mathrm{p}}y)$$

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

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

$$\begin{aligned}\phi[\mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n)][\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,] \\ = \Phi_{\mathrm{r}}([\,\lambda\iota \in I.\ \perp \mid \iota_1\colon\phi[\theta_1]x_1 \mid \ldots \mid \iota_n\colon\phi[\theta_n]x_n\,])\end{aligned}$$

$$\psi[\mathbf{rcd}(\iota_1\colon\theta_1,\ldots,\iota_n\colon\theta_n)]y = [\,\iota_1\colon\psi[\theta_1](\Psi_{\mathrm{r}}y\,\iota_1) \mid \ldots \mid \iota_n\colon\psi[\theta_n](\Psi_{\mathrm{r}}y\,\iota_n)\,].$$

These function pairs are related to the logical relations $\rho[\theta]$ in a way that is analogous to the previous lemma:

Theorem 5.4 (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 a primitive type δ, and $x \in [\![\delta]\!]$. ¿From the fact that $\Phi_{\mathrm{p}}\,;\Psi_{\mathrm{p}}$ is an identity, and the first part of Lemma 5.2, we have

  $$\langle x, \Psi_{\mathrm{p}}(\Phi_{\mathrm{p}}(\widetilde{\phi}\,[\delta]x))\rangle = \langle x, \widetilde{\phi}\,[\delta]x\rangle \in \widetilde{\rho}\,[\delta].$$

  Then the definitions (5.3) of $\phi[\delta]$ and (4.2) of $\rho[\delta]$ give

  $$\langle x, \phi[\delta]x\rangle = \langle x, \Phi_{\mathrm{p}}(\widetilde{\phi}\,[\delta]x)\rangle \in \rho[\delta].$$

  Now suppose $\langle x, y\rangle \in \rho[\delta]$. By the definition (4.2) of $\rho[\delta]$, we have $\langle x, \Psi_{\mathrm{p}}y\rangle \in \widetilde{\rho}\,[\delta]$. Then the second part of Lemma 5.2 and the definition (5.3) of $\psi[\delta]$ give

  $$x = \widetilde{\psi}\,[\delta](\Psi_{\mathrm{p}}y) = \psi[\delta]y.$$

- 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 (5.3) 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 (4.2) 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 (4.2) 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 (5.3) of $\psi[\theta_1 \to \theta_2]$, $f = \psi[\theta_1 \to \theta_2]g$.

- Suppose θ is $\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)$, and $[\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,]$ belongs to $[\![\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)]\!]$. For each k between one and n, the fact that $\Phi_{\mathrm{r}} ; \Psi_{\mathrm{r}}$ is an identity, and the first part of the induction hypothesis for θ_k, give

$$\begin{aligned}\langle x_k, \Psi_{\mathrm{r}}(\Phi_{\mathrm{r}}([\,\lambda\iota.\ \perp \mid \iota_1\colon\phi[\theta_1]x_1 \mid \ldots \mid \iota_n\colon\phi[\theta_n]x_n\,]))\iota_k\rangle &= \langle x_k, \phi[\theta_k]x_k\rangle \\ &\in \rho[\theta_k].\end{aligned}$$

Then the definition (5.3) of $\phi[\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)]$ and the definition (4.2) of $\rho[\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)]$ give

$$\begin{aligned}&\langle[\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,], \phi[\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)][\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,]\rangle \\ &\quad = \langle[\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,], \Phi_{\mathrm{r}}([\,\lambda\iota.\ \perp \mid \iota_1\colon\phi[\theta_1]x_1 \mid \ldots \mid \iota_n\colon\phi[\theta_n]x_n\,])\rangle \\ &\quad \in \rho[\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)].\end{aligned}$$

Now suppose $\langle[\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,], y\rangle \in \rho[\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)]$. By the definition (4.2) of $\rho[\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)]$, for each k between one and n, $\langle x_k, \Psi_{\mathrm{r}}y\,\iota_k\rangle \in \rho[\theta_k]$. Then by the second part of the induction hypothesis for θ_k, $x_k = \psi[\theta_k](\Psi_{\mathrm{r}}y\,\iota_k)$, and since this holds for all k, the definition (5.3) of $\psi[\mathbf{rcd}(\iota_1\colon\theta_1, \cdots, \iota_n\colon\theta_n)]$ gives

$$\begin{aligned}\psi[\mathbf{rcd}(\iota_1\colon\theta_1, \ldots, \iota_n\colon\theta_n)]y &= [\,\iota_1\colon\psi[\theta_1](\Psi_{\mathrm{r}}y\,\iota_1) \mid \ldots \mid \iota_n\colon\psi[\theta_n](\Psi_{\mathrm{r}}y\,\iota_n)\,] \\ &= [\,\iota_1\colon x_1 \mid \ldots \mid \iota_n\colon x_n\,].\end{aligned}$$

END OF PROOF

An immediate consequence of the bracketing theorem is that

Corollary 5.5 *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 $\phi[\theta] \mathbin{;} \psi[\theta] = I_{[\![\theta]\!]}$. 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 † for the reflection of a graph, we can restate the bracketing theorem as:

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

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 Theorem 4.6, one can express implicit conversions in terms of the embedding-retraction pairs:

Theorem 5.6 *When $\theta \leq \theta'$, $[\![\mathcal{P}(\theta \leq \theta')]\!] = \phi[\theta] \mathbin{;} \psi[\theta']$.*

Proof. Suppose $x \in [\![\theta]\!]$. By the first part of the bracketing theorem, $\langle x, \phi[\theta]x \rangle \in \rho[\theta]$. Then, by Theorem 4.6, $\langle [\![\mathcal{P}(\theta \leq \theta')]\!]x, \phi[\theta]x \rangle \in \rho[\theta']$, and by the second part of the bracketing theorem, $[\![\mathcal{P}(\theta \leq \theta')]\!]x = \psi[\theta'](\phi[\theta]x)$.
END OF PROOF

Finally, 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 5.7 *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 5.6 expresses the meaning of a proof of the subtype judgement $\theta \leq \theta'$ as a function of the retraction-embedding pairs associated with θ and θ', which are determined by the judgement itself rather than its proof. Similarly, Theorem 5.7 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 5.8 *The intrinsic semantics is coherent.*

Of course, the coherence of the kind of language we have been considering has been known for some time [6], 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 primitive values, functions, and records are embedded within it, as well as in certain aspects of the primitive operations, such as the choice of **trueint**.

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 [7].

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*

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

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

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. To see this, we first define a subsidiary relations for primitive types:

Definition 6.3 *For primitive types* δ, *the relation* $\widetilde{\sigma}\,[\delta] \subseteq \mathbf{Z}_\perp \times \mathbf{Z}_\perp$ *is such that:*

$$\langle z, z' \rangle \in \widetilde{\sigma}\,[\delta] \textit{ iff } \exists x \in [\![\delta]\!].\ \langle x, z \rangle, \langle x, z' \rangle \in \widetilde{\rho}\,[\delta].$$

The reader may verify from the definition (4.1) of $\widetilde{\rho}\,[\delta]$ that

Lemma 6.4

$$\langle z, z' \rangle \in \widetilde{\sigma}\,[\mathbf{int}] \quad \textit{iff} \quad z = z'$$

$$\langle z, z' \rangle \in \widetilde{\sigma}\,[\mathbf{nat}] \quad \textit{iff} \quad z = z' \textit{ and } (z \geq 0 \textit{ or } z = \perp)$$

$$\langle z, z' \rangle \in \widetilde{\sigma}\,[\mathbf{bool}] \quad \textit{iff} \quad (z = z' = \perp) \textit{ or } (z > 0 \textit{ and } z' > 0) \textit{ or } z = z' = 0.$$

Then the $\sigma[\theta]$ are described by:

Theorem 6.5

$$\langle y, y' \rangle \in \sigma[\delta] \quad \textit{iff} \quad \langle \Psi_{\mathrm{p}} y, \Psi_{\mathrm{p}} y' \rangle \in \widetilde{\sigma}\,[\delta]$$

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

$$\langle y, y' \rangle \in \sigma[\mathbf{rcd}(\iota_1 \colon \theta_1, \ldots, \iota_n \colon \theta_n)]$$
$$\textit{iff} \quad \langle \Psi_{\mathrm{r}}\, y\, \iota_1, \Psi_{\mathrm{r}}\, y'\, \iota_1 \rangle \in \sigma[\theta_1] \textit{ and } \cdots \textit{ and } \langle \Psi_{\mathrm{r}}\, y\, \iota_n, \Psi_{\mathrm{r}}\, y'\, \iota_n \rangle \in \sigma[\theta_n].$$

Proof. Using the definitions (6.1) of σ, (4.2) of $\rho[\delta]$, and (6.3) of $\widetilde{\sigma}$, we have $\langle y, y' \rangle \in \sigma[\delta]$ iff there is an $x \in [\![\delta]\!]$ such that $\langle x, y \rangle, \langle x, y' \rangle \in \rho[\delta]$, iff there is an $x \in [\![\delta]\!]$ such that $\langle x, \Psi_{\mathrm{p}} y \rangle, \langle x, \Psi_{\mathrm{p}} y' \rangle \in \widetilde{\rho}\,[\delta]$, iff $\langle \Psi_{\mathrm{p}} y, \Psi_{\mathrm{p}} y' \rangle \in \widetilde{\sigma}\,[\delta]$.

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 (4.2) 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 (4.2) of $\rho[\theta_1 \rightarrow \theta_2]$ gives $\langle f, \hat{g} \rangle \in \rho[\theta_1 \rightarrow \theta_2]$, and since this holds when $\hat{g}$ is either g or g', the definition (6.1) of σ gives $\langle g, g' \rangle \in \sigma[\theta_1 \rightarrow \theta_2]$.

Finally, we have the case where $\theta = \mathbf{rcd}(\iota_1 \colon \theta_1, \ldots, \iota_n \colon \theta_n)$. Using the definitions (6.1) of σ, (2.1) of $[\![\theta]\!]$, (4.2) of $\rho[\mathbf{rcd}(\iota_1 \colon \theta_1, \ldots, \iota_n \colon \theta_n)]$, and (6.1) of σ, we have that

$$\langle y, y' \rangle \in \sigma[\theta]$$

iff there is a record $r \in [\![\theta]\!]$ such that

$$\langle r, y \rangle, \langle r, y' \rangle \in \rho[\theta],$$

iff there are $x_1 \in [\![\theta_1]\!], \ldots x_n \in [\![\theta_n]\!]$ such that

$$\langle [\, \iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n \,], y \rangle, \langle [\, \iota_1 \colon x_1 \mid \ldots \mid \iota_n \colon x_n \,], y' \rangle \in \rho[\theta],$$

iff there are $x_1 \in [\![\theta_1]\!], \ldots x_n \in [\![\theta_n]\!]$ such that

$$\begin{aligned} &\langle x_1, \Psi_{\mathrm{r}}\, y\, \iota_1 \rangle, \langle x_1, \Psi_{\mathrm{r}}\, y'\, \iota_1 \rangle \in \rho[\theta_1] \text{ and} \\ &\qquad \cdots \text{ and } \langle x_n, \Psi_{\mathrm{r}}\, y\, \iota_n \rangle, \langle x_n, \Psi_{\mathrm{r}}\, y'\, \iota_n \rangle \in \rho[\theta_n], \end{aligned}$$

iff

$$\langle \Psi_{\mathrm{r}}\, y\, \iota_1, \Psi_{\mathrm{r}}\, y'\, \iota_1 \rangle \in \sigma[\theta_1] \text{ and } \cdots \text{ and } \langle \Psi_{\mathrm{r}}\, y\, \iota_n, \Psi_{\mathrm{r}}\, y'\, \iota_n \rangle \in \sigma[\theta_n].$$

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 [9, 10], 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 [5, Section 7], [11, Section 5], [12].

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) [13, Section 2.4.11], [14, Section 3.3], where the realizers are natural numbers (used as Gödel numbers), and the "extensional model of hereditarily continuous functionals" (ECF) [13, Section 2.6.5], [14, Section 3.9], where the realizers are functions from natural numbers to natural numbers. Troelstra attributes HEO to Kreisel [15, Section 4.2], and ECF to both Kreisel [15] and, independently, Kleene [16].

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 [17, 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 6.6 *Suppose $\pi \vdash p : \theta$, and $\varepsilon, \varepsilon' \in I \Rightarrow U$ satisfy*

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

Then

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

Proof. If ε and ε' are related as supposed, then for each $\iota \in \operatorname{dom}\pi$, 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

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

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

Finally, we can connect the $\sigma[\theta]$ with a family of functions defined in terms of the embedding-retraction pairs that bracket the $\rho[\theta]$:

Definition 6.7 *For types θ, the function $\mu[\theta]$ is the composition $\psi[\theta]\,;\phi[\theta] \in U \Rightarrow U$.*

It is easily seen that each $\mu[\theta]$ is idempotent, so that its range is its set of fixed points. Morever, it is well known that this set of fixed points, ordered as a sub-partial-ordering of U, is a domain that is isomorphic to $[\![\theta]\!]$.

The $\mu[\theta]$ are related to the $\sigma[\theta]$ by the following theorem, which plays a role at the level of PER's that is similar to the bracketing theorem:

Theorem 6.8 *For each type θ:*

1. *If $\langle y, y' \rangle \in \sigma[\theta]$, then $\mu[\theta]y = \mu[\theta]y'$.*

2. *$\langle \mu[\theta]y, \mu[\theta]y \rangle \in \sigma[\theta]$.*

3. *If $\langle y, y \rangle \in \sigma[\theta]$, then $\langle y, \mu[\theta]y \rangle \in \sigma[\theta]$.*

Proof.

1. If $\langle y, y' \rangle \in \sigma[\theta]$, then there is an $x \in [\![\theta]\!]$ such that $\langle x, y \rangle$, $\langle x, y' \rangle \in \rho[\theta]$. Then the second part of the bracketing theorem gives $x = \psi[\theta]y = \psi[\theta]y'$, so that $\mu[\theta]y = \mu[\theta]y'$.

2. We have $\langle \psi[\theta]y, \mu[\theta]y \rangle = \langle \psi[\theta]y, \phi[\theta](\psi[\theta]y) \rangle \in \rho[\theta]$, by the first part of the bracketing theorem. Thus, taking x to be $\psi[\theta]y$, the definition of $\sigma[\theta]$ gives $\langle \mu[\theta]y, \mu[\theta]y \rangle \in \sigma[\theta]$.

3. If $\langle y, y \rangle \in \sigma[\theta]$, then there is an $x \in [\![\theta]\!]$ such that $\langle x, y \rangle \in \rho[\theta]$. Then, by the second part of the bracketing theorem, $x = \psi[\theta]y$. Moreover, by the first part of the bracketing theorem, $\langle x, \phi[\theta]x \rangle \in \rho[\theta]$, which in conjunction with $\langle x, y \rangle \in \rho[\theta]$ gives $\langle y, \phi[\theta]x \rangle \in \sigma[\theta]$. Then $x = \psi[\theta]y$ gives $\langle y, \mu[\theta]y \rangle \in \sigma[\theta]$.

END OF PROOF

The reader may verify that as consequences of the parts of this theorem:

1. Every equivalence class of $\sigma[\theta]$ is a subset of the preimage under $\mu[\theta]$ of some fixed point of $\mu[\theta]$.

2. Every fixed point of $\mu[\theta]$ belongs to an equivalence class of $\sigma[\theta]$. (Thus, since each preimage of a fixed point contains that fixed point, every preimage of a fixed point contains at least one equivalence class.)

3. Every equivalence class of $\sigma[\theta]$ contains a fixed point of $\mu[\theta]$. (Thus, since no preimage contains more than one fixed point, no preimage of a fixed point contains more than one equivalence class.)

7 Future Directions

Obviously, we would like to extend our approach to languages with 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 [18].)

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 [17, Chapter 15].) In general, however, there may be no sensible way to order the set of equivalence classes to make $[\![\theta]\!]$ into a domain.

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] John C. Reynolds. Using category theory to design implicit conversions and generic operators. In Neil D. Jones, editor, *Semantics-Directed Compiler Generation*, volume 94 of *Lecture Notes in Computer Science*, pages 211–258, Berlin, 1980. Springer-Verlag.

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

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

[6] Pierre-Louis Curien and Giorgio Ghelli. Coherence of subsumption, minimum typing, and type-checking in $F_{\leq}$. *Mathematical Structures in Computer Science*, 2(1):55–91, March 1992.

[7] David M. R. Park. The Y-combinator in Scott's lambda-calculus models. Symposium on Theory of Programming, University of Warwick, unpublished; cited in [8], 1970.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

Recent BRICS Report Series Publications

RS-00-32 John C. Reynolds. *The Meaning of Types — From Intrinsic to Extrinsic Semantics*. December 2000. 35 pp.

RS-00-31 Bernd Grobauer and Julia L. Lawall. *Partial Evaluation of Pattern Matching in Strings, revisited*. November 2000. 48 pp.

RS-00-30 Ivan B. Damgård and Maciej Koprowski. *Practical Threshold RSA Signatures Without a Trusted Dealer*. November 2000. 14 pp.

RS-00-29 Luigi Santocanale. *The Alternation Hierarchy for the Theory of μ-lattices*. November 2000. 44 pp. Extended abstract appears in *Abstracts from the International Summer Conference in Category Theory*, CT2000, Como, Italy, July 16–22, 2000.

RS-00-28 Luigi Santocanale. *Free μ-lattices*. November 2000. 51 pp. Short abstract appeared in *Proceedings of Category Theory 99*, Coimbra, Portugal, July 19–24, 1999. Full version to appear in a special conference issue of the *Journal of Pure and Applied Algebra*.

RS-00-27 Zoltán Ésik and Werner Kuich. *Inductive -Semirings*. October 2000. 34 pp.

RS-00-26 František Čapkovič. *Modelling and Control of Discrete Event Dynamic Systems*. October 2000. 58 pp.

RS-00-25 Zoltán Ésik. *Continuous Additive Algebras and Injective Simulations of Synchronization Trees*. September 2000. 41 pp.

RS-00-24 Claus Brabrand and Michael I. Schwartzbach. *Growing Languages with Metamorphic Syntax Macros*. September 2000.

RS-00-23 Luca Aceto, Anna Ingólfsdóttir, Mikkel Lykke Pedersen, and Jan Poulsen. *Characteristic Formulae for Timed Automata*. September 2000. 23 pp.

RS-00-22 Thomas S. Hune and Anders B. Sandholm. *Using Automata in Control Synthesis — A Case Study*. September 2000. 20 pp. Appears in Maibaum, editor, *Fundamental Approaches to Software Engineering: First International Conference*, FASE '00 Proceedings, LNCS 1783, 2000, pages 349–362.