CS263-Spring 2008
Topic 2: Type Theory
Section 5.1: Type Semantics I

Dana S. Scott
Hillman University Professor (Emeritus)
School of Computer Science
Carnegie Mellon University
====
Visiting Professor EECS
Visiting Scientist
Logic & Methodology Program
University of California, Berkeley

Last edited 19 February 2008

A set-theoretic interpretation of types

■ Restricting abstraction

When we introduced λ-abstraction in the models, it was pointed out that if Φ is a continuous operator, then $\lambda X.\Phi(X)$ is the ***largest set*** $U \in \mathbb{P}$ such that $U[X] = \Phi(X)$ for all $X \in \mathbb{P}$. If we had known there was at least one set representing Φ this way, then we could have used this as the ***definition***, bytaking a union. Then we could have proved:
$\lambda X.\Phi(X)[Y] == \Phi(Y)$.

But an ***explicit formula*** was given that not only gave ***one such*** U but also was the ***largest***.

Note. To emphasize the above remark, suppose that $U[X] \subseteq \Phi(X)$ for all $X \in \mathbb{P}$. Then we showed that $U \subseteq \lambda X.U[X] \subseteq \lambda X.\Phi(X)$. So $\lambda X.\Phi(X)$ is indeed the largest representative.

In turning next to a discussion of **types**, it will helpful to represent functions on **subsets** of $\mathbb{P}$ — but with values in $\mathbb{P}$. An important example is $\mathbb{N} \subseteq \mathbb{P}$ (and remember we take $n = \{n\}$ for $n \in \mathbb{N}$). But, we will also consider smaller sets $\mathbb{M} \subseteq \mathbb{N} \subseteq \mathbb{P}$.

By definition we take $\lambda\, n \in \mathbb{M}.\, X_n$ to be the largest set $U \subseteq \mathbb{N}$ such that

$$U[n] \subseteq X_n \text{ holds for all } n \in \mathbb{M}.$$

Phrased this way, there certainly ***are*** such U because $U = \emptyset$ is a trivial choice. But, why is there a ***largest***?

The answer is to take the ***union*** of all the U satisfying the displayed condition above. One of the Projects was to prove that under our definition of $U[X]$ we have

$$\bigcup\{U[Y] \mid U \in \mathbb{U}\} = (\bigcup \mathbb{U})[Y]$$

for **any** family of sets $\mathbb{U} \subseteq \mathbb{P}$ and any $Y \in \mathbb{P}$. That justifies the definition. (Why?)

Theorem. For any given system of sets X_n for $n \in \mathbb{M}$, we have $(\lambda\, n \in \mathbb{M}.\, X_n)[m] = X_m$ for all $m \in \mathbb{M}$.

Proof. Take, with an m given, as one of the sets in the union: $U = \{(\langle m \rangle, k) \mid k \in X_m\}$. Clearly, $U[m] = X_m$,while, for $n \neq m$, we have $U[m] = \emptyset$. **Q.E.D.**

Note. When $n \in \mathbb{N} \setminus \mathbb{M}$, because the λ-abstract is taken as ***maximal***, we will have:

$$(\lambda\, n \in \mathbb{M}.\, X_n)[m] = \mathbb{N}, \text{ the largest set in } \mathbb{P}.$$

A special case we need to use often is $\mathbb{M} = \{0, 1, \ldots, n-1\}$. We introduce an ***abbreviation***:

$$\langle\langle X_0, X_1, \ldots, X_{n-1} \rangle\rangle == \lambda\, i < n\, .\, X_i$$

In other words, though we had n-tuples of elements of $\mathbb{N}$ before, we also have to have n-tuples of elements of $\mathbb{P}$. They work out a little differently, however. In $\mathbb{N}$, we find $\langle i, j \rangle \neq \langle i, j, k \rangle$; but in $\mathbb{P}$, we have

$$\langle\langle X_0, X_1, \ldots, X_{n-1} \rangle\rangle = \langle\langle X_0, X_1, \ldots, X_{n-1}, \mathbb{N} \rangle\rangle = \langle\langle X_0, X_1, \ldots, X_{n-1}, \mathbb{N}, \mathbb{N}, \ldots \rangle\rangle.$$

Moreover, $\langle\langle X_0, X_1, \ldots, X_{n-1} \rangle\rangle[i] = X_i$, if $i < n$; but $\langle\langle X_0, X_1, \ldots, X_{n-1} \rangle\rangle[i] = \mathbb{N}$, if $i \geq n$.

Warning. There are ***many different ways*** of introducing ***ordered pairs*** into $\mathbb{P}$.

In order to make a later task easier, we give ***one more*** definition of ***pairs***.

$$\lfloor X, Y \rfloor = 2\, X \bigcup 2\, Y + 1 \text{ for } X,\ Y \in \mathbb{P}$$

$$1\,\text{st} = \lambda\, Z.\{n \mid 2\, n \in Z\}$$

$$2\,\text{nd} = \lambda\, Z.\{n \mid 2\, n + 1 \in Z\}$$

The ***idea*** here in defining $\lfloor X, Y \rfloor$ is to put a ***copy*** of X on the ***even integers*** and a ***copy*** of Y on the ***odd integers***. Clearly we are working with continuous and computable operators here and the key formulae about pairing are easy to prove:

Theorem. For all $X,\ Y,\ Z \in \mathbb{P}$ we have:

> $1\,\text{st}[\lfloor X, Y \rfloor] = X$;
>
> $2\,\text{nd}[\lfloor X, Y \rfloor] = Y$; and
>
> $\lfloor 1\,\text{st}\,[Z], 2\,\text{nd}[Z] \rfloor = Z$.

We defined at least two notions of ***cartesian product*** for sets ***in*** $\mathbb{P}$. Now we use this pairing just defined for taking the products of ***subclasses*** of $\mathbb{P}$.

$$\mathbb{U} \times \mathbb{V} = \{\lfloor X, Y \rfloor \mid X \in \mathbb{U}\ \&\ Y \in \mathbb{V}\} \text{ for } \mathbb{U}, \mathbb{V} \subseteq \mathbb{P}$$

Note. Under this definition it follows that $\mathbb{P} \times \mathbb{P} = \mathbb{P}$. (Why?)

Types as equivalences

The ***semantics for types*** we will be using is based on a kind of ***equivalence relations***.

A ***partial equivalence relation*** or ***PER*** is a binary relation

$$\mathcal{R} \subseteq \mathbb{P} \times \mathbb{P}$$

which is ***symmetric*** and ***transitive***; that is, the implications

$$X \,\mathcal{R}\, Y \Rightarrow Y \,\mathcal{R}\, X \text{ and } X \,\mathcal{R}\, Y \;\&\; Y \,\mathcal{R}\, Z \Rightarrow X \,\mathcal{R}\, Z$$

hold for all $X, Y, Z \in \mathbb{P}$.

And, in the above, we use this familiar ***shorthand***:

$$X \,\mathcal{R}\, Y \Leftrightarrow \lfloor X, Y \rfloor \in \mathcal{R}$$

It is important to keep in mind that we are ***not*** assuming that ***reflexivity*** $X \,\mathcal{R}\, X$ holds for ***all*** $X \in \mathbb{P}$. Thus, the subset of $\mathbb{P}$ given by $\{X \mid X \,\mathcal{R}\, X\}$ is being used a "***preliminary***" ***subtype*** of $\mathbb{P}$. However, keep in mind that the ***whole type*** is the relation $\mathcal{R}$.

For $X \in \mathbb{P}$ and $\mathcal{R}$ a ***PER***, we write $X : \mathcal{R}$ to mean $X \,\mathcal{R}\, X$.

And we read $X : \mathcal{R}$ as ***X is of type*** $\mathcal{R}$.

When $X \,\mathcal{R}\, Y$ we have, however, to think of X and Y as ***equivalent representatives*** of objects of type $\mathcal{R}$. In other words, the ***same object*** may have ***many representations***.

Warning. Mathematicians often insist on having ***uniquely determined*** objects by introducing ***equivalence classes*** and ***quotient sets***. One notation for this is as follows:

For $X : \mathcal{R}$, define $X/\mathcal{R} = \{Y \in \mathbb{P} \mid X \,\mathcal{R}\, Y\}$. Then

$$\mathbb{P}/\mathcal{R} = \{X/\mathcal{R} \mid X : \mathcal{R}\}.$$

This is sometimes convenient, but we will ***not*** make much use of the idea here.

Of course, every subclass of $\mathbb{Q} \subseteq \mathbb{P}$ can be thought of as a type by defining:

$$X \,\mathcal{I}d(\mathbb{Q})\, Y \Leftrightarrow X = Y \in \mathbb{Q}$$

But, as we will find from examples, these may not be the most interesting types.

We note that $\mathcal{I}d(\emptyset) = \emptyset$. But a slightly more interesting ***PER*** will often use is this one:

$$\bot = \mathcal{I}d(\{\emptyset\})$$

The difference between $\mathcal{I}d(\emptyset)$ and $\mathcal{I}d(\{\emptyset\})$ is that $\mathcal{I}d(\emptyset)$ being empty, there is no way to map ***into*** it. On the other hand $\emptyset : \mathcal{I}d(\{\emptyset\})$, and so many operations map into this type — e.g., K[$\emptyset$] ought to be a ***good candidate***.

■ Type mappings

There is a very great variety of types — even among the ***PERs***. To be able to explore them, we define a number of ways of constructing ***new types from old***.

The prime reason we want to single out a specific kinds of type in this way is to have a ***guaranteed*** structure on our data.

Warning. From now on we will often say ***type*** instead of ***PER***.

When, given two types $\mathcal{A}$ and $\mathcal{B}$, we write $U : \mathcal{A} \to \mathcal{B}$ to mean that U represents an operator mapping objects of type $\mathcal{A}$ ***to*** objects of type $\mathcal{B}$, we want to be sure that whenever $X : \mathcal{A}$ we ***will know*** that $U(X) : \mathcal{B}$ and that $U[X]$ is not just some ***random element*** of $\mathbb{P}$. More, actually, is required of operators, because types allow ***equivalent*** representations of objects.

By definition, $U : \mathcal{A} \to \mathcal{B}$ means that

whenever $X \, \mathcal{A} \, Y$ holds, ***then*** $U[X] \, \mathcal{B} \, U[Y]$ holds.

We can read $U : \mathcal{A} \to \mathcal{B}$ as saying:

U is a ***mapping*** from $\mathcal{A}$ to $\mathcal{B}$, or

U ***maps*** $\mathcal{A}$ to $\mathcal{B}$

Already, there are some ***very easy*** theorems to prove:

Theorem. For any type $\mathcal{A}$, it is true that $\lambda X. X : \mathcal{A} \to \mathcal{A}$.

Theorem. For types $\mathcal{A}$ and $\mathcal{B}$, if $B : \mathcal{B}$, then $K[B] = \lambda X. B : \mathcal{A} \to \mathcal{B}$.

Theorem. For types $\mathcal{A}$, $\mathcal{B}$, and C, if $U : \mathcal{A} \to \mathcal{B}$ and $V : \mathcal{B} \to C$, then

$$\lambda X. V[U[X]] : \mathcal{A} \to C.$$

We need to use the ***composition*** of mappings so often we will write:

$$(V \circ U) = \lambda X. V[U[X]].$$

Note. The notation $(V \circ U)$ is preferred, because it is more readable than using a ***combinator***!

Also clear is that a type with ***just one element*** picks out just one element of a type under a mapping. A typical one-element type is the type $\perp = \mathcal{I}d(\{\emptyset\})$.

Theorem. For any $B \in \mathbb{P}$ and any type $\mathcal{B}$, we have $B : \mathcal{B}$ if, and only if, there is a mapping $U : \perp \to \mathcal{B}$ such that $U[\emptyset] == B$.

We remark that the empty type $\mathcal{I}d(\emptyset) = \emptyset$ has an ***extreme*** property.

Theorem. For any $U \in \mathbb{P}$ and any type $\mathcal{B}$, we have $U : \mathcal{I}d(\emptyset) \to \mathcal{B}$.

Now, we have been writing $U : \mathcal{A} \to \mathcal{B}$ as if this were a ***type statement***. Is it possible that $\mathcal{A} \to \mathcal{B}$ can be considered as being a type in itself ***made up from*** $\mathcal{A}$ and $\mathcal{B}$? The answer is ***YES!***

Given types $\mathcal{A}$ and $\mathcal{B}$, the type $(\mathcal{A} \to \mathcal{B})$ is the ***PER*** defined as follows:

$U \, (\mathcal{A} \to \mathcal{B}) \, V$ if, and only if, ***whenever*** $X \, \mathcal{A} \, Y$ holds, ***then*** $U[X] \, \mathcal{B} \, V[Y]$ holds.

This means, roughly speaking, that U and V — as operators — do ***equivalent things*** in $\mathcal{B}$ to ***equivalent things*** in $\mathcal{A}$.

Note. Under this definition that $U : \mathcal{A} \to \mathcal{B}$ does mean the same as $U \, (\mathcal{A} \to \mathcal{B}) \, U$.

We should, however, verify the following statement.

Theorem. Under the definition, $(\mathcal{A} \to \mathcal{B})$ is a ***PER***, provided $\mathcal{A}$ and $\mathcal{B}$ are.

The proof is easy.

Warning. By the way, $(\mathcal{I}d(\emptyset) \to \mathcal{B}) = \mathbb{P}$. (Why?)

We should also relate ***restricted*** abstraction to the ***unrestricted*** version.

Theorem. For types $\mathcal{A}$ and $\mathcal{B}$, if $\lambda X. U[X] : \mathcal{A} \to \mathcal{B}$, then

$$\lambda X : \mathcal{A}. U[X] \; (\mathcal{A} \to \mathcal{B}) \; \lambda X. U[X].$$

Proof. Suppose $S \, \mathcal{A} \, T$. Then $U[S] \, \mathcal{B} \, U[T]$.

We have $S : \mathcal{A}$. (Why?) Therefore, $(\lambda X : \mathcal{A}. U[X])[S] = U[S]$.

Of course, $(\lambda X. U[X])[T] = U[T]$.

Hence, $(\lambda X : \mathcal{A}. U[X])[S] \, (\mathcal{A} \to \mathcal{B}) \, (\lambda X. U[X])[T\}$. **Q.E.D.**

We should also ***generalize*** an earlier theorem.

Theorem. If types $\mathcal{A}$, $\mathcal{B}$, and C are such there are mappings where

$U_0 \, (\mathcal{A} \to \mathcal{B}) \, U_1$ and $V_0 \, (\mathcal{B} \to C) \, V_1$, then we have $V_0 \circ U_0 (\mathcal{A} \to C) \, V_1 \circ U_1$.

■ Type constructs

The definition of $(\mathcal{A} \to \mathcal{B})$ is one way of making new types out of old, and there are a ***multitude*** of other ways of doing so. In this section we present five important ones.

The first is by ***intersection***.

$$U \, (\mathcal{A} \cap \mathcal{B}) \, V \text{ if, and only if, both } U \, \mathcal{A} \, V \text{ and } U \, \mathcal{B} \, V.$$

Theorem. If $\mathcal{A}$ and $\mathcal{B}$ are ***PERs***, then so is $\mathcal{A} \cap \mathcal{B}$.

In fact, the intersection of ***any*** family of ***PERs*** is again a ***PER***, as is easily proved.

It is not the case that the ***union*** of ***PERs*** is necessarily a ***PER***. The example of the two ***Pers*** $\{\{0\}, \{1\}\} \times \{\{0\}, \{1\}\}$ and $\{\{1\}, \{2\}\} \times \{\{1\}, \{2\}\}$ show this. (Why?) Wht is needed is to ***add*** something to a union.

$$U \, (\mathcal{A} \uplus \mathcal{B}) \, V \text{ if, and only if, } \exists \, Z \in \mathbb{P}, n \in \mathbb{N}. \text{ such that } Z[0] = U \text{ and } Z[n] = V \text{ and}$$

$$\forall \, i < n. \text{ either } \; Z[i] \, \mathcal{A} \, Z[i+1] \text{ or } Z[i] \, \mathcal{B} \, Z[i+1].$$

Theorem. If $\mathcal{A}$ and $\mathcal{B}$ are ***PERs***, then so is $\mathcal{A} \uplus \mathcal{B}$.

We could say that $\mathcal{A} \uplus \mathcal{B}$ is the ***PER*** that is ***generated by*** the union $\mathcal{A} \cup \mathcal{B}$. It is also the ***least PER*** containing the union. (Why?)

There are situations when the union of ***PERs*** is again a ***PER***, however. We formulate two.

Theorem. If two ***PERs*** $\mathcal{A}$ and $\mathcal{B}$ are such that $\mathcal{A} \cap \mathcal{B} = \emptyset$, then $\mathcal{A} \cup \mathcal{B}$ is again a ***PER***.

Theorem. If a sequence of ***PERs*** is such that $\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \ldots \subseteq \mathcal{A}_i \subseteq \ldots$, then the union $\bigcup_{i=0}^{\infty} \mathcal{A}_i$ is again a ***PER***.

The next definition expands the idea of a ***cartesian product*** from sets to ***PERs***.

$$U \, (\mathcal{A} \times \mathcal{B}) \, V \text{ if, and only if, } \textbf{\textit{both}} \; 1\,\mathrm{st}[U] \, \mathcal{A} \, 1\,\mathrm{st}[V] \; \textbf{\textit{and}} \; 2\,\mathrm{nd}[U] \, \mathcal{B} \, 2\,\mathrm{nd}[V].$$

We need to remember here that under our interpretation of ***ordered pairs of sets*** in $\mathbb{P}$, every set is at the same time a pair: $U = \lfloor 1\,\mathrm{st}\,[U], 2\,\mathrm{nd}[\,U] \rfloor$. So we could have written this definition equivalently as follows:

$$\lfloor U_0, U_1 \rfloor \, (\mathcal{A} \times \mathcal{B}) \, \lfloor V_0, V_1 \rfloor \text{, and only if, } \textbf{\textit{both}} \; U_0 \, \mathcal{A} \, V_0 \; \textbf{\textit{and}} \; U_1 \, \mathcal{B} \, V_1.$$

Theorem. If $\mathcal{A}$ and $\mathcal{B}$ are ***PERs***, then so is $\mathcal{A} \times \mathcal{B}$.

Warning. Do not ***confuse*** $\mathcal{A} \times \mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$. (Why?)

The idea of making a ***disjoint sum*** (sometimes called ***coproduct***) is to make ***disjoint copies*** of $\mathcal{A}$ and of $\mathcal{B}$ and ***then*** take a union.

$$U\,(\mathcal{A}+\mathcal{B})\,V \text{ if, and only if, } \textbf{\textit{either}}\ 1\,\text{st}[U] = 1\,\text{st}[V] = 0 \text{ and } 2\,\text{nd}[U]\,\mathcal{A}\,2\,\text{nd}[V]\ \textbf{\textit{or}}$$

$$1\,\text{st}[U] = 1\,\text{st}[V] = 1 \text{ and } 2\,\text{nd}[U]\,\mathcal{B}\,2\,\text{nd}[V]$$

Again, there is another way of writing this definition:

$$\lfloor U_0, U_1 \rfloor\,(\mathcal{A}+\mathcal{B})\,\lfloor V_0, V_1 \rfloor \text{ if, and only if, } \textbf{\textit{either}}\ U_0 = V_0 = 0 \text{ and } U_1\,\mathcal{A}\,V_1\ \textbf{\textit{or}}$$

$$U_0 = V_0 = 1 \text{ and } U_1\,\mathcal{B}\,V_1$$

Theorem. If $\mathcal{A}$ and $\mathcal{B}$ are ***PERs***, then so is $\mathcal{A}+\mathcal{B}$.

We can also relate the sum to the product via a ***union***:

Theorem. $\mathcal{A}+\mathcal{B} = (\mathcal{I}d(\{0\}) \times \mathcal{A}) \cup (\mathcal{I}d(\{1\}) \times \mathcal{B})$

Note also these equations :

Corollary. $\mathcal{I}d(\{0\}) \times \mathcal{A} = \mathcal{A} + \emptyset$ and $\mathcal{I}d(\{1\}) \times \mathcal{B} = \emptyset + \mathcal{B}$.

The last construct to be defined now is called ***lifting***.

$$\mathcal{A}_{\bot} == (\mathcal{A}+\emptyset) \cup \bot$$

The need for this construct will be clear later in discussing ***partial functions***.

There are many, many ***relationships*** between types that can be explained in terms of mappings. That will be the subject of the ***next lecture***.