CS263-Spring 2008
Topic 2: Type Theory
Section 6.1: Type Semantics II

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 23 February 2008

A set-theoretic interpretation of types (Review)

Type constructs

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.

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***.

$U\,(\mathcal{A} \cap \mathcal{B})\,V$ if, and only if, both $U\,\mathcal{A}\,V$ 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?) What 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 $\mathcal{A} \bigcup \mathcal{B}$. 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 $\mathcal{A} \bigcap \mathcal{B} = \emptyset$, then $\mathcal{A} \bigcup \mathcal{B}$ 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***.

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

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:

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

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 $\mathcal{A} \bigcap \mathcal{B}$. (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.

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

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

Again, there is another way of writing this definition:

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

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

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

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***.

$$\mathcal{A}_\perp = (\mathcal{A} + \emptyset) \bigcup \perp$$

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***.

Relating types

Isomorphism

Many types represent the ***same structure***. For example, types such as $\mathcal{Id}(\{n\})$ and $\mathcal{Id}(\{m\})$ are both types of ***a single, isolated element***, and it makes little difference whether we call it $n = 0$ or $m = 1001$or some other integer. The two types give us the same kind of structure and are ***isomorphic***. Similarly, $\mathcal{Id}(\{0, 1\})$ and $\mathcal{Id}(\{13, 666\})$ are isomorphic.

Here is the formal definition:

Two types $\mathcal{A}$ and $\mathcal{B}$ are said to be ***isomorphic*** provided that

there are U and V with $U : \mathcal{A} \to \mathcal{B}$ and $V : \mathcal{B} \to \mathcal{A}$ such that

$V \circ U(\mathcal{A} \to \mathcal{A})\, \lambda X. X$ and $U \circ V(\mathcal{B} \to \mathcal{B})\, \lambda X. X$.

We notate this relationship as $\mathcal{A} \cong \mathcal{B}$.

Note. A stronger condition is being ***recursively isomorphic***. This adds the extra condition that both U and V are in $\mathbb{RE}$. We shall find many examples where this stronger relationship holds because so many basic operators are ***computable***.

The conditions on U and V for giving an isomorphism could also have been stated as:

$$\forall X : \mathcal{A} . V[U[X]]\, \mathcal{A}\, X \text{ and } \forall Y : \mathcal{B} . U\,[V\,[Y]]\, \mathcal{B}\, Y.$$

(Why?)

Theorem. Isomorphism (and, likewise, recursive isomorphism) is an ***equivalence relation*** among types.

Proof. (1) Inasmuch as $\lambda X. X\, (\mathcal{A} \to \mathcal{A})\, \lambda X. X$, then $\mathcal{A} \cong \mathcal{A}$ follows.

(2) If we have $\mathcal{A} \cong \mathcal{B}$ using U and V as in the definition, then we have $\mathcal{B} \cong \mathcal{A}$ using V and U.

(3) Suppose $\mathcal{A} \cong \mathcal{B}$ using U and V, and $\mathcal{B} \cong C$ using S and T. Then we can show that $\mathcal{A} \cong C$ by using $S \circ U$ and $V \circ T$. (Why?) **Q.E.D.**

Note. In formulating these conditions, some authors prefer the more algebraic notation using these abbreviations: $\mathbb{1} = \lambda X.X$ and $\mathbb{1}_{\mathcal{A}} = \lambda X : \mathcal{A}.X$.

Theorem. For $A \in \mathbb{P}$ and $\mathbb{A} \subseteq \mathbb{P}$, we have $\perp \cong \mathcal{Id}(\{A\}) \cong \mathbb{A} \times \mathbb{A} \cong (\mathbb{A} \times \mathbb{A}) \to (\mathbb{A} \times \mathbb{A})$.

We also note that isomorphisms ***compose*** under ***some of*** our type constructs:

Theorem. For ***PERs*** $\mathcal{A}_0 \cong \mathcal{A}_1$ and $\mathcal{B}_0 \cong \mathcal{B}_1$, we have

$$\mathcal{A}_0 \to \mathcal{B}_0 \cong \mathcal{A}_1 \to \mathcal{B}_1,$$

$$\mathcal{A}_0 \times \mathcal{B}_0 \cong \mathcal{A}_1 \times \mathcal{B}_1,$$

$$\mathcal{A}_0 + \mathcal{B}_0 \cong \mathcal{A}_1 + \mathcal{B}_1, \text{ and}$$

$$\mathcal{A}_{0_\perp} \cong \mathcal{A}_{1_\perp}.$$

Question. What about $\bigcap$ and $\uplus$?

■ Order-theoretic properties

Note. On pp. 116-118 of "***An Introduction to Lambda Calculi for Computer Scientists***" by Chris Hankin, a number of properties of ***intersection types*** are mentioned. We shall now verify the corresponding properties using our definitions.

Clearly $\subseteq$ is a partial ordering, and we have no need of discussing the obvious properties of that relation in connection with $\bigcap$. There are also similar properties for $\uplus$.

Keep in mind that we also have two ***extreme types*** $\emptyset$ and $\mathbb{P} = \mathbb{P} \times \mathbb{P}$, so that for all types $\mathcal{A}$ we have $\emptyset \subseteq \mathcal{A} \subseteq \mathbb{P}$. And $\mathbb{P}$ is a very ***silly*** type making all things equivalent. Hence, we can say both that $\mathbb{P} = \mathbb{P} \to \mathbb{P}$ and $\mathbb{P} \cong \mathcal{Id}(\{0\})$. (Why?) Those are the ***uninteresting*** order-theoretic properties. Here are more interesting ones:

Theorem. For types $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ we have

$$(\mathcal{A} \to \mathcal{C}) \bigcap (\mathcal{B} \to \mathcal{C}) \subseteq ((\mathcal{A} \bigcap \mathcal{B}) \to \mathcal{C}), \text{ and}$$

$$(\mathcal{A} \to (\mathcal{B} \bigcap \mathcal{C})) = (\mathcal{A} \to \mathcal{B}) \bigcap (\mathcal{A} \to \mathcal{C}).$$

The proofs are very easy from the definitions.

Warning. The reason that the first relationship is ***not*** an equality is that we might have $(\mathcal{A} \bigcap \mathcal{B}) = \emptyset$ thereby making $((\mathcal{A} \bigcap \mathcal{B}) \to \mathcal{C})$ ***too large***!

In the cases of the other constructs, we find some ***monotonicity***. But one case is ***special***.

Theorem. If we have types $\mathcal{A} \subseteq \mathcal{B}$, and $\mathcal{C} \subseteq \mathcal{D}$, then we have

$$(\mathcal{A} + \mathcal{C}) \subseteq (\mathcal{B} + \mathcal{D}),$$

$$(\mathcal{A} \times \mathcal{C}) \subseteq (\mathcal{B} \times \mathcal{D}),$$

$$\mathcal{A}_\perp \subseteq \mathcal{B}_\perp, \text{ and}$$

$$(\mathcal{B} \to \mathcal{C}) \subseteq (\mathcal{A} \to \mathcal{D}).$$

Warning. In the case of the mapping-space construct there is a ***reversal*** of order. (Why?)

■ Sums, products and exponentials

In the case of intersection types we had an ***identity*** between types. More generally, just ***isomorphisms*** should be expected. (Why?) We start with ***products***.

Theorem. For types $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ we have:

$$\perp \times \mathcal{A} \cong \mathcal{A},$$

$$\mathcal{A} \times \mathcal{B} \cong \mathcal{B} \times \mathcal{A}, \text{ and}$$

$$(\mathcal{A} \times \mathcal{B}) \times \mathcal{C} \cong \mathcal{A} \times (\mathcal{B} \times \mathcal{C}).$$

Note. In fact, here all the isomorphisms are ***recursive isomorphisms***. The proofs are left as an exercise.

Theorem. For types $\mathcal{A}$, $\mathcal{B}$, and C we have:

$$(\mathcal{A} \times \mathcal{B}) \to C \cong \mathcal{A} \to (\mathcal{B} \to C), \text{ and}$$

$$\mathcal{A} \to (\mathcal{B} \times C) \cong (\mathcal{A} \to \mathcal{B}) \times (\mathcal{A} \to C).$$

Remember, $\bot = \mathcal{I}d(\{\emptyset\})$, and keep in mind that one-element types behave as if they were the product of ***no*** types at a time.

Theorem. For all types $\mathcal{A}$, we have

$$\bot \to \mathcal{A} \cong \mathcal{A}, \text{ and}$$

$$\mathcal{A} \to \bot \cong \bot.$$

For other finite types, we have "arithmetic" answers. First some definitions:

$$0\,\mathcal{A} = \emptyset$$

$$(n+1)\,\mathcal{A} = n\,\mathcal{A} + \mathcal{A}$$

$$\mathcal{A}^0 = \bot$$

$$\mathcal{A}^{n+1} = \mathcal{A}^n \times \mathcal{A}$$

It is also convenient to have an abbreviated notation for some ***finite types***.

$$\mathcal{N}_n = \mathcal{I}d\,(\{0,\,1,\,\ldots \mathrm{n}-1\})$$

Theorem. For all types $\mathcal{A}$ and $n \in \mathbb{N}$, we have

$$\mathcal{N}_n \times \mathcal{A} \cong n\,\mathcal{A}, \text{ and}$$

$$\mathcal{N}_n \to \mathcal{A} \cong \mathcal{A}^n.$$

And, of course our definitions agree with ***finite integer arithmetic***.

Theorem. For $n, m \in \mathbb{N}$, we have

$$\mathcal{N}_n + \mathcal{N}_m \cong \mathcal{N}_{n+m},$$

$$\mathcal{N}_n \times \mathcal{N}_m \cong \mathcal{N}_{n\,m}, \text{ and}$$

$$\mathcal{N}_n \to \mathcal{N}_m \cong \mathcal{N}_{m^n}.$$

Turning now to ***sums***, there are several general laws.

Theorem. For types $\mathcal{A}$, $\mathcal{B}$, and C we have

$$\emptyset + \mathcal{A} \cong \mathcal{A},$$

$$\mathcal{A} + \mathcal{B} \cong \mathcal{B} + \mathcal{A},$$

$$(\mathcal{A} + \mathcal{B}) + C \cong \mathcal{A} + (\mathcal{B} + C), \text{ and}$$

$$\mathcal{A} \times (\mathcal{B} + C) \cong (\mathcal{A} \times \mathcal{B}) + (\mathcal{A} \times C).$$

Theorem. For types $\mathcal{A}$, $\mathcal{B}$, and C we have

$$(\mathcal{A} + \mathcal{B}) \to C \cong (\mathcal{A} \to C) \times (\mathcal{B} \to C).$$

Note. It was shown in the book, "***Isomorphisms of types : from λ-calculus to information retrieval and language design***" by Roberto Di Cosmo, that ***no other general laws*** are to be expected. However, as we shall see, there are ***special types*** over $\mathbb{P}$ which ***do*** satisfy interesting ***isomorphism relations***.