CS263-Spring 2008
Topic 1: The Lambda Calculus
Section 3.1: 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 3 February 2008

A Calculus of Operators

Pairs and sequences

- **Those combinators (in case we need them)**
- **Sequences of integers**

For a period in these lectures we need to discuss the (non-negative) integers in a ***mathematical way***.

Let $\mathbb{N}$ stand for the ***set*** of integers, and consider this ***one-one correspondence***:

$$\mathbb{N} \cong 1 + \mathbb{N} \times \mathbb{N}$$

In words, we can say that, save for ***one*** integer, every integer can be regarded as a ***pair*** of integers. We have used this before, and we gave this ***definition*** of a pair:

$$(n, m) = 2^n (2m + 1)$$

The one integer left out is, of course, $\mathbf{0}$.

Let $\mathbb{N}^*$ stand for the set of ***finite sequences*** of integers, and consider this ***one-one correspondence***:

$$\mathbb{N} \cong \mathbb{N}^*$$

This relationship can be justified via the following ***recursive definition***:

$$\langle\rangle = 0$$
$$\langle x_0, x_1, \ldots, x_{n-1}, x_n\rangle = (\langle x_0, x_1, \ldots, x_{n-1}\rangle, x_n)$$

Theorem. Under this definition every integer can be uniquely regarded as a sequence number.

Proof. Inasmuch as ***pair numbers*** uniquely determine their components, so ***sequence numbers*** uniquely determine ***all*** their components.

To argue that ***every*** integer ***is*** a sequence number, argue by ***Strong Induction***.

Assume about an integer k that all ***smaller integers*** are sequence numbers. If $k = 0$, then it is — ***by definition*** — the number of the ***empty sequence***.

If $k > 0$, then ***we know*** (in view of integer factorization) that uniquely $k = 2^p (2q + 1) = (p, q)$.

Because $p < k$, then, by the ***inductive assumption***, $p = \langle x_0, x_1, \ldots, x_{n-1}\rangle$ for suitable uniquely determined integers. Let $x_n = q$, and then $k = \langle x_0, x_1, \ldots, x_{n-1}, x_n\rangle$. **Q.E.D.**

■ Operations on sequences

From time to time, we will need to ***concatenate*** sequences and to check their ***lengths***. Try these definitions:

$$x \frown \langle\rangle = x$$
$$x \frown (y, z) = (x \frown y, z)$$

$$|\langle\rangle| = 0$$
$$|(y, z)| = |y| + 1$$

Question. Why are these definitions ***justified***?

Question. And can you prove these next ***equations***?

$$\langle\rangle \frown x = x$$
$$x \frown (y \frown z) = (x \frown y) \frown z$$

$$\langle x_0, x_1, \ldots, x_{n-1}\rangle \frown \langle y_0, y_1, \ldots, y_{m-1}\rangle = \langle x_0, x_1, \ldots, x_{n-1}, y_0, y_1, \ldots, y_{m-1}\rangle$$
$$\langle x_0, x_1, \ldots, x_{n-1}\rangle == \langle x_0\rangle \frown \langle x_1\rangle \frown \ldots \frown \langle x_{n-1}\rangle$$

$$|x \frown y| = |x| + |y|$$
$$|\langle x_0, x_1, \ldots, x_{n-1}\rangle| == n$$

We also need to form the ***set*** of terms of a sequence:

$$\S\,\langle x_0, x_1, \ldots, x_{n-1}\rangle = \{x_0, x_1, \ldots, x_{n-1}\}$$

Here are some facts:

$$\S\langle\rangle = \emptyset$$
$$\S(y, z) = \S y \cup \{z\}$$
$$\S(x \frown y) = \S x \cup \S y$$

Warning. Remember in standard mathematics sets and sequences are ***different***.

Operations on sets

■ Some key types of sets

We have been letting $\mathbb{N}$ stand for the ***set*** of integers, but what we need in addition are some key ***families of sets of integers.*** First we have the so-called ***powerset*** consisting of ***all*** sets of integers.

$$\mathbb{P} = \mathcal{P}\,\mathbb{N} = \{X \mid X \subseteq \mathbb{N}\}$$

For the theory of computability, a subfamily of $\mathbb{P}$, the ***recursively enumerable*** sets, is important,

as is the family of ***recursive sets***.

$$\mathbb{RE} = \{X \mid X \subseteq \mathbb{N} \;\&\; X \text{ is recursively enumerable}\}$$
$$\mathbb{Rec} = \{X \mid X \subseteq \mathbb{N} \;\&\; X \text{ is recursive}\}$$

As a simplifying ***convention*** we are going to identify ***singleton sets*** of integers with the corresponding integer: $n = \{n\}$ for $n \in \mathbb{N}$. In Set Theory is not generally done, but on ***one*** set which is not otherwise correlated with sets this is ***harmless***.

With this convention about singleton sets, we have so far this range of types.

$$\mathbb{N} \subseteq \mathbb{Rec} \subseteq \mathbb{RE} \subseteq \mathbb{P}$$

One more important family is the family of ***finite sets***:

$$\mathbb{F} = \{X \mid X \subseteq \mathbb{N} \;\&\; X \text{ is finite}\}$$

And we have:

$$\mathbb{N} \subseteq \mathbb{F} \subseteq \mathbb{Rec} \subseteq \mathbb{RE} \subseteq \mathbb{P}$$

Note. There is of course a vast range of families ***between*** $\mathbb{RE}$ and $\mathbb{P}$, but we do not stop now to notate any of them.

■ Continuous operators

There are many useful operations mapping sets of integers ***to*** sets of integers. Here are several we will use:

$$X \cap Y = \{n \mid n \in X \;\&\; n \in Y\}$$
$$X \cup Y = \{n \mid n \in X \vee m \in Y\}$$
$$X + Y = \{m + n \mid n \in X \;\&\; m \in Y\}$$
$$X \times Y = \{(n, m) \mid n \in X \;\&\; m \in Y\}$$
$$X^* = \{x \mid \S x \subseteq X\}$$
$$X \frown Y = \{x \frown y \mid x \in X \;\&\; y \in Y\}$$

These particular operations also have some ***special properties***. First, as operators, they are ***monotone***. For set operators

$\Phi(X_0, X_1, \ldots, X_{n-1})$, ***monotonicity*** is defined as:

$$\Phi(X_0, X_1, \ldots, X_{n-1}) \subseteq \Phi(Y_0, Y_1, \ldots, Y_{n-1}) \text{ whenever } X_i \subseteq Y_i \text{ for all } i < n.$$

In words you can say: ***The more you put in, the more you get out***. It is to be hoped that it is clear that ***all*** the specific operators defined above ***do have*** this ***monotonicity*** property.

Note. The prime example of a ***non-monotone*** set operator is the operation of set-theoretical ***complement***:

$$X \setminus Y = \{n \mid n \in X \ \& \ n \notin Y\}$$

However, our monotone operators above also have an additional property, roughly described as: ***If you only want a little more out, you only have to put a little more in***.

That is not a very precise statement! We have to interpret "***little***" as "***finite***" to make a better characterization. We give the definition for operators on two sets of integers. It is easy to generalize this to ***more*** arguments.

A monotone operator $\Phi(X, Y)$ is said to be ***continuous*** if, and only if,

for all $X, Y \in \mathbb{P}$ whenever $F \subseteq \Phi(X, Y)$ and $F \in \mathbb{F}$, then

there are $D, E \in \mathbb{F}$ with $D \subseteq X$ and $E \subseteq Y$

such that $F \subseteq \Phi(D, E)$, for all $X, Y \in \mathbb{P}$.

Note. There are many reasons to call this property "***continuous***", but it is sufficient to say now that it is analogous to the ***ϵ-δ-definitions*** of continuity from ***The Calculus***.

And, it is to be hoped that it is clear that ***all*** the specific operators defined above ***do have*** this ***continuity*** property.

In the case of operators on one variable, continuity and monotonicity can be captured together in one equation:

Theorem. An operator $\Phi : \mathbb{P} \to \mathbb{P}$ mapping sets of integers to sets of integers is ***both*** monotone and continuous if, and only if, for all $X \in \mathbb{P}$ we have

$$\Phi(X) = \bigcup\{\Phi(D) \mid D \in \mathbb{F} \ \& \ D \subseteq X\}.$$

Note. We can generalize this result to operators with ***many arguments*** easily.

Note. And to save writing, we will now assume "***continuous***" implies "***monotone***".

■ λ Abstraction

Observation. A continuous operator $\Phi : \mathbb{P} \to \mathbb{P}$ is completely determined by the information $n \in \Phi(D)$ for $D \in \mathbb{F}$. (Why?)

Let's consider as well continuous operators of ***many arguments***. To each such operator $\Phi(X, Y_0, Y_1, \ldots, Y_{n-1})$, where ***one*** argument has been singled out, we now associate this ***new*** operator:

$$\lambda X.\Phi(X, Y_0, Y_1, \ldots, Y_{n-1}) = \{0\} \cup \{(x, n) \mid n \in \Phi(\S x, Y_0, Y_1, \ldots, Y_{n-1})\}$$

On the right-hand side, x ranges over ***all sequence numbers*** (*i.e.*, all integers), and then $\S x$ ranges over ***all finite sets of integers***.

Note. The right-hand side of the definition does not depend on the set variable X anymore. In other words the λX on the left-hand side is a new ***variable-binding construct***.

The result is an operator that only depends on the remaining variables $Y_0, Y_1, \ldots, Y_{n-1}$. We call the result the ***λ-abstract***.

Note. The need for the **{0}** will be explained later. For the time being, it does no harm.

Theorem. If $\Phi(X, Y_0, Y_1, \ldots, Y_{n-1})$ is a continuous operator of ***all*** its variables $X, Y_0, Y_1, \ldots, Y_{n-1}$, then $\lambda X.\Phi(X, Y_0, Y_1, \ldots, Y_{n-1})$ is a continuous operator of the remaining variables $Y_0, Y_1, \ldots, Y_{n-1}$.

In order to show that not only does the λ-abstraction construct preserve enough information ***to determine*** the original continuous operator, but it also makes it easy to define what we mean by a ***computable operator***. To this end, consider the following binary set operator called ***application***:

$$U[X] = \{n \mid \exists\, (x, n) \in U. \S\, x \subseteq X\}$$

Theorem. Not only is $U[X]$ continuous, but we have

$$\lambda X.\Phi(X, Y_0, Y_1, \ldots, Y_{n-1})[X] == \Phi(X, Y_0, Y_1, \ldots, Y_{n-1})$$

for every continuous operator **Φ**.

The detailed proof is left for a project (though the truth of the assertion should be apparent).

Consider a continuous operator $\Phi(X, Y, Z)$ of three set variables as an ***example***. The iterated λ-abstract $U = \lambda X.\lambda Y.\lambda Z.\Phi(X, Y, Z)$ is just a (slightly) complicated ***set*** of integers. (Why?) But, it contains all the necessary information to ***redefine*** the given operator Φ, because $U[X][Y][Z] = \Phi(X, Y, Z)$. (Why?) Hence,

Corollary. ℙ is a ***model*** for the `crules` of combinators using this defined set operator $U[X]$ as the ***application operation***.

Corollary. Under the `crules`, no combinator can reduce both to `K` and `K[J]`.

Question. Are there *other* models?