Typed $\lambda$-calculus: Denotational Semantics of Call-By-Value

P. B. Levy
University of Birmingham

Preliminary note: substitution in CBV

For the pure calculus, we gave a substitution lemma expressing $[[M[N/x]]]$ in terms of $[[M]]$ and $[[N]]$. But that will not be possible in CBV, as the following example demonstrates. We define terms $x : \text{bool} \vdash M, M' : \text{bool}$ and $\vdash N : \text{bool}$ by

\[ M \overset{\text{def}}{=} \text{true} \]
\[ M' \overset{\text{def}}{=} \text{case x of \{true. true, false. true\}} \]
\[ N \overset{\text{def}}{=} \text{error CRASH} \]

But in any CBV semantics we will have $[[M]] = [[M']]$ because $M =_{\eta\text{bool}} M'$

\[ [[M[N/x]]] \neq [[M'[N/x]]] \]

However, what we will be able to describe semantically is the substitution of a restricted class of terms, called values.

\[ V ::= x | n | \text{true} | \text{false} | \text{inl} V | \text{inr} V | \lambda x.M \]

A value, in any syntactic environment, is terminal. And a closed term is a value iff it is terminal. In the study of call-by-value, we define a substitution $\Gamma \xrightarrow{k} \Delta$ to be a function mapping each identifier $x : A$ in $\Gamma$ to a value $\Delta \vdash V : A$. If $W$ is a value, then $k^*W$ is a value, for any substitution $k$.

1 Denotational Semantics for CBV

Let us think about how to give a denotational semantics for call-by-value $\lambda$-calculus with errors. Let $E$ be the set of errors.
1.1 First Attempt

Let’s propose that for a type $A$, its denotation $[A]$ will be a set that’s a *universe for terms*: by this I mean that a closed term of type $A$ will denote an element of $[A]$. Then we should have

$$
\begin{align*}
[\text{bool}] &= \mathbb{B} + E \\
[int] &= \mathbb{Z} + E \\
[\text{bool} \times \text{int}] &= (\mathbb{B} \times \mathbb{Z}) + E \\
[A \times B] &= [A] \ast [B]
\end{align*}
$$

where $\ast$ is an operation on sets that would have to satisfy

$$(\mathbb{B} + E) \ast (\mathbb{Z} + E) = (\mathbb{B} \times \mathbb{Z}) + E$$

I can’t see any such operation, so we give up on this proposal.

1.2 Second Attempt

Let’s make $[A]$ a set that’s a *universe for values*, meaning that a closed value of type $A$ will denote an element of type $[A]$. In particular we want

$$
\begin{align*}
[\text{bool}] &= \mathbb{B} \\
[int] &= \mathbb{Z} \\
[A + B] &= [A] + [B] \\
[A \times B] &= [A] \times [B]
\end{align*}
$$

and we postpone the semantic equation for $\rightarrow$.

A *semantic environment* for $\Gamma$ maps each identifier $x : A$ in $\Gamma$ to an element of $[A]$. We write $[\Gamma]$ for the set of semantic environments.

A closed term of type $B$ either returns a closed value or raises an error. So it should denote an element of $[B] + E$. More generally, a term $\Gamma \vdash M : B$ should denote, for each semantic environment $\rho \in [\Gamma]$, an element of $[B] + E$. Hence

$$[\Gamma] \xrightarrow{[M]} [B] + E$$
Now let’s think about \([A \rightarrow B]\). A closed value of type \(A \rightarrow B\) is a \(\lambda\)-abstraction \(\lambda x_A.M\). This can be applied to a closed value \(V\) of type \(A\), and gives a closed term \(M[V/x]\) of type \(B\). So we define

\[
[A \rightarrow B] = [A] \rightarrow ([B] + E)
\]

We can easily write out the semantics of terms now.

### 1.3 Substitution Lemma

According to what we have said, a value \(\Gamma \vdash V : A\) denotes a function

\[
[\Gamma] \overset{[V]}{\rightarrow} [A] + E
\]

To formulate a substitution lemma, we also want \(V\) to denote a function

\[
[\Gamma] \overset{[V]^{\text{val}}}{\rightarrow} [A]
\]

and \([V]^{\text{val}}\) should be related to \([V]\) by

\[
[V]\rho = \text{inl} [V]^{\text{val}}\rho
\]  

(1)

or as a diagram:

\[
\begin{array}{c}
[\Gamma] \overset{[V]^{\text{val}}}{\rightarrow} [A] \\
\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
[A] + E
\end{array}
\]

We define \([V]^{\text{val}}\) and verify (1) by induction on \(V\).

Given a substitution \(\Gamma \overset{k}{\rightarrow} \Delta\), we obtain a function \([\Delta] \overset[k]{[\Gamma]}{\rightarrow} [\Gamma]\). It maps \(\rho \in [\Delta]\) to the semantic environment for \(\Gamma\) that takes each identifier \(x : A\) in \(\Gamma+\) to \([k(x)]^{\text{val}}\rho\).

Now we can formulate two substitution lemmas: one for substitution into terms, and one for substitution into values.

**Proposition 1.** Let \(\Gamma \overset{k}{\rightarrow} \Delta\) be a substitution, and let \(\rho\) be a semantic environment for \(\Delta\).
1. For any term $\Gamma \vdash M : B$, we have $[k^* M] \rho = [M]([k] \rho)$, or as a diagram:

\[
\begin{array}{c}
\[\Delta\] \\
\downarrow[k] \\
[\Gamma] \\
\downarrow [M] \\
\[B\] + E
\end{array}
\]

2. For any value $\Gamma \vdash V : B$, we have $[k^* V]^\text{val} \rho = [V]^\text{val}([k] \rho)$, or as a diagram:

\[
\begin{array}{c}
\[\Delta\] \\
\downarrow[k] \\
[\Gamma] \\
\downarrow [V]^\text{val} \\
\[B\] + E
\end{array}
\]

As usual we first prove this for renamings (or at least weakening).

### 1.4 Computational Adequacy

It is all very well to define a denotational semantics, but it’s no good if it doesn’t agree with the way the language was defined (the operational semantics).

**Proposition 2.** Let $M$ be a closed term.

1. If $M \Downarrow V$, then $[M] = \text{inl} [V]^\text{val}$.
2. If $M \Downarrow e$, then $[M] = \text{inr} e$.

We prove this by induction on $\Downarrow$ and $\Downarrow$.