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

P. B. Levy
adapted for G53POP by T. Altenkirch

Universities of Birmingham and Nottingham

1 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

$$M \equiv \text{true}$$
$$M' \equiv \text{case } x \text{ of } \{ \text{true } \rightarrow \text{true } | \text{false } \rightarrow \text{true } \}$$
$$N \equiv \text{error CRASH}$$

But in any CBV semantics we will have

$$[[M]] = [[M']] \quad \text{because } M =_\eta \text{bool } M'$$
$$[[M[N/x]]] \neq [[M'[N/x]]]$$

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

$$V ::= x | \underline{n} | \text{true } | \text{false } | (\#\text{left}, V) | (\#\text{right}, V) | \lambda x.M$$

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 $k$.

2 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.
2.1 First Attempt

Let’s say that a type denotes a set, and that a closed term of type $A$ denotes an element of $[A]$. Then $\textbf{bool}$ would denote $\mathbb{B} + E$, because a closed term of type $\textbf{bool}$ either returns $\text{true}$ or $\text{false}$, or raises an error. Likewise $\textbf{int}$ should denote $\mathbb{Z} + E$.

Next, we have to define $[\Gamma]$, for a context $\Gamma$, and this should be the set of semantic environments. In particular, the context $x : \textbf{bool}, y : \textbf{int}$ should denote $\mathbb{B} \times \mathbb{Z}$. But there does not seem to be any way of obtaining that set from the sets $[\textbf{bool}]$ and $[\textbf{int}]$ as we have defined them. So we need to do something different.

2.2 Second Attempt

Let’s instead make $[A]$ the set of denotations of closed values, i.e. terminal terms, rather than denotations of closed terms. We then want $\textbf{bool}$ to denote $\mathbb{B}$, and we’ll complete the semantics of types below.

We define $[\Gamma]$ to be the set of functions mapping each identifier $x : A$ in $\Gamma$ to an element of $[A]$.

A closed term of type $A$ either returns a closed value or raises an error. So it should denote an element of $[A] + E$. 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 go through the various types.

- $\textbf{int}$ denotes $\mathbb{Z}$.
- A closed value of type $A + B$ is $(\text{#left}, V)$ or $(\text{#right}, V)$, where $V$ is a closed value, so

$$[A + B] = [A] + [B]$$

- A closed value of type $A \rightarrow B$ is a $\lambda$-abstraction $\lambda x. M$. This can be applied to a closed value $V$ of type $A$, and gives a closed term $M[V/x]$ of type $B$. So

$$[A \rightarrow B] = [A] \rightarrow [B] + E$$

We can easily write out the semantics of terms now.
2.3 Substitution Lemma

As it stands, a value $\Gamma \vdash V : A$ denotes a function from $\llbracket \Gamma \rrbracket$ to $\llbracket A \rrbracket_\bot$. But, for the substitution lemma, we also want $V$ to denote a function $\llbracket \Gamma \rrbracket \overset{[V]}{\rightarrow} \llbracket A \rrbracket$.

This is defined by induction on $V$. The two denotations of $V$ are related as follows.

**Proposition 1.** Suppose $\Gamma \vdash V : A$ is a value, and $\rho$ is a semantic environment for $\Gamma$. Then

$$\llbracket V \rrbracket \rho = (\#up, [V]^{val}_{\rho})$$

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

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

**Proposition 2.** Let $\Gamma \xrightarrow{k} \Delta$ be a substitution, and let $\rho$ be a semantic environment for $\Delta$.

1. For any term $\Gamma \vdash M : B$, we have $\llbracket k^* M \rrbracket_{\rho} = \llbracket M \rrbracket_{\llbracket k \rrbracket_{\rho}}$.
2. For any value $\Gamma \vdash V : B$, we have $\llbracket k^* V \rrbracket^{val}_{\rho} = [V]^{val}_{\llbracket k \rrbracket_{\rho}}$.

2.4 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 3.** Let $M$ be a closed term.

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

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