Typed λ-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 $[\![M[N/\mathtt{x}]]\!]$ in terms of $[\![M]\!]$ and $[\![N]\!]$. But that will not be possible in CBV, as the following example demonstrates. We define terms $\mathtt{x} : \mathtt{bool} \vdash M, M' : \mathtt{bool}$ and $\vdash N : \mathtt{bool}$ by

$$\begin{aligned} M &\stackrel{\text{def}}{=} \mathtt{true} \\ M' &\stackrel{\text{def}}{=} \mathtt{case\ x\ of\ } \{\mathtt{true} \to \mathtt{true} \mid \mathtt{false} \to \mathtt{true}\} \\ N &\stackrel{\text{def}}{=} \mathtt{error\ CRASH} \end{aligned}$$

But in any CBV semantics we will have

$$\begin{aligned} [\![M]\!] &= [\![M']\!] \qquad \text{because } M =_{\eta\,\mathtt{bool}} M' \\ [\![M[N/\mathtt{x}]]\!] &\neq [\![M'[N/\mathtt{x}]]\!] \end{aligned}$$

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

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

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 $\mathtt{x} : A$ in Γ 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 λ-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 `bool` would denote $\mathbb{B} + E$, because a closed term of type `bool` either returns `true` or `false`, or raises an error. Likewise `int` should denote $\mathbb{Z} + E$.

Next, we have to define $[\![\Gamma]\!]$, for a context Γ, and this should be the set of semantic environments. In particular, the context $\mathtt{x} : \mathtt{bool}, \mathtt{y} : \mathtt{int}$ should denote $\mathbb{B} \times \mathbb{Z}$. But there does not seem to be any way of obtaining that set from the sets $[\![\mathtt{bool}]\!]$ and $[\![\mathtt{int}]\!]$ 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 `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 $\mathtt{x} : A$ in Γ 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.

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

$$[\![A + B]\!] = [\![A]\!] + [\![B]\!]$$

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

$$[\![A \to B]\!] = [\![A]\!] \to [\![B]\!] + E$$

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 $[\![\Gamma]\!]$ to $[\![A]\!]_\perp$. But, for the substitution lemma, we *also* want V to denote a function

$$[\![\Gamma]\!] \xrightarrow{[\![V]\!]^{\mathrm{val}}} [\![A]\!]$$

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 ρ is a semantic environment for Γ. Then*

$$[\![V]\!]\rho = (\#\mathrm{up}, [\![V]\!]^{\mathrm{val}}\rho)$$

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

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 ρ be a semantic environment for Δ.*

1. *For any term $\Gamma \vdash M : B$, we have $[\![k^* M]\!]\rho = [\![M]\!]([\![k]\!]\rho)$.*
2. *For any value $\Gamma \vdash V : B$, we have $[\![k^* V]\!]^{\mathrm{val}}\rho = [\![V]\!]^{\mathrm{val}}([\![k]\!]\rho)$.*

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 $[\![M]\!] = \mathrm{inl}\ [\![V]\!]^{\mathrm{val}}$.*
2. *If $M \lightning e$, then $[\![M]\!] = \mathrm{inr}\ e$.*

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