Environment Semantics

Gert Smolka, Saarland University

November 29, 2019

Efficient implementations of call-by-value evaluation are realized with environments rather than substitutions. Also, when one talks about programs informally, one does this in terms of an environment semantics (e.g., the value of a variable at runtime). We study an environment semantics for call-by-value lambda calculus clarifying this situation. We show that for closed terms the environment semantics agrees with a big step semantics, and that the big-step semantics agrees with the standard reduction semantics.

1 Preliminaries

We define the **subscript operation** $A[n]$ for lists as follows:

$$\begin{aligned} [][n] &:= \emptyset \\ (s :: A)[0] &:= {}^\circ s \\ (s :: A)[\mathsf{S}n] &:= A[n] \end{aligned}$$

2 Simple Substitution

For computational purposes we may assume that call-by-value reduction operates on closed terms. This means that only closed β-redexes $(\lambda s)(\lambda t)$ are reduced. Under this assumption a much simplified substitution operation suffices. Given a closed term $(\lambda s)t$, there are no variables $x \geq 1$ in s that need to be lowered, and there are no variables in t that need to be raised.

We define **simple substitution** s^n_t as follows:

$$\begin{aligned} x^n_t &:= \text{if } x = n \text{ then } t \text{ else } x \\ (\lambda u)^n_t &:= \lambda(u^{\mathsf{S}n}_t) \\ (uv)^n_t &:= (u^n_t)\,(v^n_t) \end{aligned}$$

Fact 1

1. $\mathsf{bound}\ s\,k \ \rightarrow\ k \le n \ \rightarrow\ s^n_t = s$.
2. $\mathsf{bound}\ s\,(\mathsf{S}n) \ \rightarrow\ \mathsf{closed}\ t \ \rightarrow\ \mathsf{bound}\ (s^n_t)\ n$.
3. $\mathsf{closed}\ (\lambda s)t \ \rightarrow\ \mathsf{closed}\ s^0_t$.

Proof Claim 1 and 2 follow by induction on $\mathsf{bound}\, s\, k$. Claim 3 follows from claim 2. ∎

Fact 2 (Abstract Agreement) Let the call-by-value reduction relations $\succ_{v1}$ and $\succ_{v2}$ be defined with β_1 and β_2, respectively. Then $\succ_{v1}$ and $\succ_{v2}$ agree on all closed terms (i.e., $\forall s.\ \mathsf{closed}\ s \rightarrow (\forall t.\ s \succ_{v1} t \leftrightarrow s \succ_{v2} t)$) if $\beta_1 st = \beta_2 st$ for all closed terms $(\lambda s)t$. Moreover, $\succ^*_{v1}$ and $\succ^*_{v2}$ agree on closed terms if in addition $\beta_1 st$ is closed whenever $(\lambda s)t$ is closed.

Fact 3 (Substitution Agreement) Let $(\lambda s)t$ be closed. Then $s^0_t = s[t.I]$.

Proof By the assumption we have $\mathsf{bound}\, s\, 1$ and that t is closed. We define substitutions σ_n recursively

$$\begin{aligned} \sigma_0 &:= t.I \\ \sigma_{\mathsf{S}n} &:= 0.(\sigma_n \circ S) \end{aligned}$$

and prove

$$\mathsf{bound}\ s\,(\mathsf{S}n) \ \rightarrow\ s^n_t = s[\sigma_n]$$

by induction on $\mathsf{bound}\ s\, n$. The proof uses the equation

$$\sigma_n(k) \ = \ \begin{cases} k & \text{if } k < n \\ t & \text{if } k = n \\ k-1 & \text{if } k > n \end{cases}$$

which follows by induction on n using $t[S] = t$ (holds since t is closed). Informally, the equation may be written as $\sigma_n \mathrel{\hat{=}} (0, \dots, n-1, t, n, n+1, \dots)$. ∎

Theorem 4 The simple and the full version of the call-by-value reduction predicate $\succ_v$ agree on closed terms. The same holds for the reflexive transitive closure $\succ^*_v$.

Proof Facts 2, 3, and Fact 1 (3). ∎

3 Big-step Semantics

We consider the call-by-value λ-calculus with an abstract β function and define a big-step predicate $s \Downarrow t$ as follows:

$$\frac{}{\lambda s \Downarrow \lambda s} \qquad \frac{s \Downarrow \lambda s' \qquad t \Downarrow t' \qquad \beta s' t' \Downarrow u}{st \Downarrow u}$$

Fact 5 If $s \Downarrow t$, then t is an abstraction.

Proof Induction on $s \Downarrow t$. ■

Recall the definition of call-by-value reduction:

$$\frac{\text{abstraction } t}{(\lambda s)t \succ_v \beta s t} \qquad \frac{s \succ_v s'}{st \succ_v s't} \qquad \frac{t \succ_v t'}{st \succ_v st'}$$

Lemma 6 (Absorption) $s \succ_v t \to t \Downarrow u \to s \Downarrow u$.

Proof Induction on $s \succ_v t$ with inversion of $t \Downarrow u$. ■

Theorem 7 (Agreement) $s \Downarrow t \leftrightarrow s \succ_v^* t \wedge \text{abstraction } t$.

Proof Direction $\to$ follows by induction on $s \Downarrow t$ using compatibility and transitivity of $\succ_v^*$. Direction $\leftarrow$ follows by star induction using the absorption lemma. ■

4 Closures and Environments

We define closures and environments as follows:

- A **closure** $s;E$ is a pair of a term s and an environment E.
  The letters p, q will range over closures.
- An **environment** is a list of closures.
  The letters E, F will range over environments.

We shall use closures to represent terms. By convention, the closures in an environment will always represent procedures. Given a closure $s;E$ in an environment, the environment E will provide values for all variables in λs. We formalize this design with an inductive **representation predicate** $p \overset{n}{\vdash} s$ with a **depth argument** n:

$$\frac{x < n}{x;E \overset{n}{\vdash} x} \qquad \frac{x \geq n \qquad E[x-n] = {}^\circ p \qquad p \overset{1}{\vdash} s}{x;E \overset{n}{\vdash} \lambda s}$$

$$\frac{s;E \overset{Sn}{\vdash} u}{\lambda s;E \overset{n}{\vdash} \lambda u} \qquad \frac{s;E \overset{n}{\vdash} u \qquad t;E \overset{n}{\vdash} v}{st;E \overset{n}{\vdash} uv}$$

We read $p \overset{n}{\vdash} s$ as p represents s at depth n. The idea is that s is a subterm of a closed term appearing below n abstractions. Recall that an index x below n abstractions is bound iff $x < n$. We call a closure p **admissible at depth n** if there is a term s such that $p \overset{n}{\vdash} s$.

Fact 8 $p \overset{n}{\vdash} s \ \rightarrow\ \mathsf{bound}\ s\ n$.

Proof By induction on $p \overset{n}{\vdash} s$. ∎

Thus $p \overset{0}{\vdash} s$ ensures that s is closed.

Fact 9 $\mathsf{bound}\ s\ n \ \rightarrow\ s; E \overset{n}{\vdash} s$.

Proof By induction on $\mathsf{bound}\ s\ n$. ∎

We say that a closure **p represents a procedure s** if

$$p \Vdash s \ := \ \exists t.\ s = \lambda t \wedge p \overset{1}{\vdash} t$$

There is an important connection between closure representation and simple substitution that we will make use of in the following.

Lemma 10 $s; E \overset{\mathsf{S}n}{\vdash} u \ \rightarrow\ p \Vdash t \ \rightarrow\ s; p :: E \overset{n}{\vdash} u^n_t$.

Proof Induction on $s; E \overset{\mathsf{S}n}{\vdash} u$ using Facts 8 and 1. ∎

Fact 11 $s; E \overset{1}{\vdash} u \ \rightarrow\ p \Vdash t \ \rightarrow\ s; p :: E \overset{0}{\vdash} u^0_t$.

Proof Immediate with the preceding lemma. ∎

Exercise 12 Prove $p \Vdash s \ \rightarrow\ n; p :: E \overset{n}{\vdash} s$.

5 Environment Semantics

We will define an inductive predicate $E \vdash s \Downarrow p$ formalizing call-by-value evaluation with environments. We will show that for closed terms s environment evaluation $[] \vdash s \Downarrow p$ agrees with big-step evaluation $s \Downarrow t$ provided big-step evaluation is defined with simple substitution (i.e., $\beta s t = s^0_t$). The two directions of the agreement are formulated by Theorem 15.

We define the inductive predicate $E \vdash s \Downarrow p$ as follows:

$$\frac{E[x] = {}^\circ p}{E \vdash x \Downarrow p} \qquad \frac{}{E \vdash \lambda s \Downarrow s; E} \qquad \frac{E \vdash s \Downarrow u; F \qquad E \vdash t \Downarrow p \qquad p :: F \vdash u \Downarrow q}{E \vdash s t \Downarrow q}$$

Note the following:

- The environment semantics mimics the big-step semantics.
- Neither an auxiliary function nor a substitution is used in the rules defining the environment semantics.
- In a derivation of $s;E \overset{n}{\vdash} p$ only subterms of s and E are used.

To prove the agreement statements of Theorem 15, we need to generalize the claims so that the canonical inductions go through. The proofs are then routine given Fact 11 about closures and simple substitution. Both proofs assume that $s \Downarrow t$ is defined with $\beta st = s_t^0$, which provides for the use of Fact 11.

Lemma 13 $E \vdash s \Downarrow p \;\to\; s;E \overset{0}{\vdash} t \;\to\; \exists u.\, p \Vdash u \;\wedge\; t \Downarrow u.$

Proof By induction on $E \vdash s \Downarrow p$ using Fact 11. ∎

Lemma 14 $s \Downarrow t \;\to\; u;E \overset{0}{\vdash} s \;\to\; \exists p.\, p \Vdash t \;\wedge\; E \vdash u \Downarrow p.$

Proof By induction on $s \Downarrow t$ using Fact 11. ∎

Theorem 15 (Agreement) Environment semantics and big-step semantics agree for closed terms s as follows:

1. $[] \vdash s \Downarrow p \;\to\; \exists t.\, p \Vdash t \;\wedge\; s \Downarrow t.$
2. $s \Downarrow t \;\to\; \exists p.\, p \Vdash t \;\wedge\; [] \vdash s \Downarrow p.$

Proof Claim 1 follows with Lemma 13 and Fact 9. Claim 2 follows with Lemma 14 and Fact 9. ∎

6 Notes

Closures were first identified by Peter Landin in 1964 [3]. Gilles Kahn promoted environment semantics as natural semantics [1]. The semantics of the functional programming language Standard ML is defined with an environment semantics [4]. Kunze et al. [2] verify abstract machines for call-by-value λ-calculus using closures and environments.

References

[1] Gilles Kahn. Natural semantics. In *STACS 87*, pages 22–39. Springer, 1987.

[2] Fabian Kunze, Gert Smolka, and Yannick Forster. Formal small-step verification of a call-by-value lambda calculus machine. In S. Ryu, editor, *16th Asian Symposium on Programming Languages and Systems, APLAS 2018, Wellington, New Zealand, December 2-6*, LNCS 11275, pages 264–283. Springer, Dec 2018.

[3] Peter J. Landin. The mechanical evaluation of expressions. *The Computer Journal*, 6(4):308–320, 1964.

[4] Robin Milner, Mads Tofte, and David MacQueen. *The Definition of Standard ML*. MIT Press, Cambridge, MA, USA, 1997.