Small-Step Evaluation

Gert Smolka, Saarland University

October 25, 2017

We consider two small-step semantics for expressions. The first one models single evaluation steps and leaves open which step is taken next if several steps are possible. The second small-step semantics models a tail-recursive interpreter for expressions.

1 Values, Operations, and Expressions

We assume a type of values, a type of operations, and an interpretation function for operations

$$v : \mathsf{Val}$$
$$f : \mathsf{Op} \qquad\qquad \varphi : \mathsf{Op} \to \mathsf{Val} \to \mathsf{Val} \to \mathsf{Val}$$

an define expressions

$$e : \mathsf{Exp} ::= v \mid f e_1 e_2$$

and evaluation

$$E : \mathsf{Exp} \to \mathsf{Val}$$
$$Ev := v$$
$$E(f e_1 e_2) := \varphi f (Ee_1)(Ee_2)$$

2 Rewrite semantics

We will define an inductive predicate

$$\succ : \mathsf{Exp} \to \mathsf{Exp} \to \mathbf{P}$$

called rewrite relation that models single evaluation steps and satisfies the equivalence

$$e \succ^* v \;\leftrightarrow\; Ee = v \tag{1}$$

We define $e \succ e'$ with the following rules:

$$\frac{}{f v_1 v_2 \succ \varphi f v_1 v_2} \qquad \frac{e_1 \succ e_1'}{f e_1 e_2 \succ f e_1' e_2} \qquad \frac{e_2 \succ e_2'}{f e_1 e_2 \succ f e_1 e_2'}$$

Note that $r \succ e'$ is not functional because in a compound expression either the left or the right argument expression may be rewritten.

Fact 1 If $e \succ e'$, then $Ee = Ee'$.

Proof Induction on $e \succ e'$. ■

Fact 2 If $e \succ^* e'$, then $Ee = Ee'$.

Proof Induction on $e \succ^* e'$ using Fact 1. ■

Fact 3 If $e_1 \succ^* e_1'$ and $e_2 \succ^* e_2'$, then $f e_1 e_2 \succ^* f e_1' e_2'$.

Proof It suffices to show $f e_1 e_2 \succ^* f e_1' e_2$ and $f e_1' e_2 \succ^* f e_1' e_2'$. The first claim follows by induction on $e_1 \succ^* e_1'$, and the second claim follows by induction on $e_2 \succ^* e_2'$. ■

Fact 4 $e \succ^* Ee$

Proof Induction on e using Fact 3. ■

Theorem 5 $e \succ^* v \leftrightarrow Ee = v$.

Proof Follows with Facts 2 and 4. ■

3 Iterative Semantics

A tail-recursive evaluator for expressions will maintain a control stack and a value stack, where the control stack holds expressions and operations. We will model a tail-recursive evaluator with an inductive predicate

$$\succ : \; \mathsf{L}\,(\mathsf{Exp} + \mathsf{Op}) \to \mathsf{L}\,(\mathsf{Op}) \to \mathbf{P}$$

satisfying the equivalence

$$[e], \mathsf{nil} \succ^* \mathsf{nil}, B \;\leftrightarrow\; B = [Ee]$$

We define $A, B \succ A', B'$ with the following rules:

$$v :: A,\ B \ \succ\ A,\ v :: B$$
$$f e_1 e_2 :: A,\ B \ \succ\ e_2 :: e_1 :: f :: A,\ B$$
$$f :: A,\ v_1 :: v_2 :: B \ \succ\ A,\ \varphi f v_1 v_2 :: B$$

Note the similarity with the execution function R for compiled expressions. We may say that the tail-recursive rewrite relation inlines the compilation of compound expressions.

Fact 6 $e :: A, B \succ^* A, Ee :: B$.

Proof Induction on e. ∎

Fact 7 (Functionality)

1. If $A, B \succ A_1, B_1$ and $A, B \succ A_2, B_2$, then $A_1 = A_2$ and $B_1 = B_2$.
2. If $A, B \succ^* \mathsf{nil}, B_1$ and $A, B \succ^* \mathsf{nil}, B_2$, then $B_1 = B_2$.

Proof Claim 1 follows by case analysis on the first assumption and inversion of the second assumption.

Claim 2 follows by induction on the first assumption, inversion on the second assumption, and Claim 1. ∎

Theorem 8 $[e], \mathsf{nil} \succ^* \mathsf{nil}, B \leftrightarrow B = [Ee]$.

Proof Follows with Facts 6 and 7. ∎

4 Termination

We define a size function $\sigma : \mathsf{Exp} \to \mathsf{N}$ such that $e \succ e'$ implies $\sigma e > \sigma e'$:

$$\sigma v := 1$$
$$\sigma(f e_1 e_2) := 2 + \sigma e_1 + \sigma e_2$$

Fact 9 If $e \succ e'$, then $\sigma e > \sigma e'$.

Proof Induction on $e \succ e'$. ∎

Fact 9 tells us that the rewriting relation $e \succ e'$ terminates.

We want to show that $A, B \succ A', B'$ terminates. For this it suffices to come up with a size function $\tau : \mathsf{L}(\mathsf{Exp} + \mathsf{Op}) \to \mathsf{N}$ such that $A, B \succ A', B'$ implies $\tau A > \tau A'$. We define τ as follows:

$$\tau(\mathsf{nil}) \ :=\ 0$$
$$\tau(e :: A) \ :=\ \sigma e + \tau A$$
$$\tau(f :: A) \ :=\ 1 + \tau A$$

Fact 10 If $A, B \succ A', B'$, then $\tau A > \tau A'$.

Proof Case analysis on $A, B \succ A', B'$. ∎