Tutorial 7: Progress

November 18, 2010

1. Consider a very simple arithmetic language containing only numbers (n) and addition:

$$A ::= n \mid A + A$$

Consider the following small-step semantics for this language:

$$\frac{A_1 \to A_1'}{A_1 + A_2 \to A_1' + A_2} \qquad \frac{A_2 \to A_2'}{n_1 + A_2 \to n_1 + A_2'} \qquad \frac{}{n_1 + n_2 \to n}\, n = n_1 + n_2$$

Prove the following lemma.

Lemma 1 (Progress for simple arithmetic expressions). *For all arithmetic expressions A, either A is a number or there exists some A' such that $A \to A'$.*

2. Suppose we extend the language with variables:

$$A ::= n \mid A + A \mid x$$

To evaluate these extended expressions, we need to know what the values of the variables are. Hence, we will evaluate the arithmetic expression in a particular *state* σ, a function from variables to values.

The rules for addition are still the same as before, except that we must now pass the state around:

$$\frac{\langle \sigma, A_1 \rangle \to \langle \sigma', A_1' \rangle}{\langle \sigma, A_1 + A_2 \rangle \to \langle \sigma', A_1' + A_2 \rangle} \qquad \frac{\langle \sigma, A_2 \rangle \to \langle \sigma', A_2' \rangle}{\langle \sigma, n_1 + A_2 \rangle \to \langle \sigma', n_1 + A_2 \rangle} \qquad \frac{}{\langle \sigma, n_1 + n_2 \rangle \to \langle \sigma, n \rangle}\, n = n_1 + n_2$$

and we need a new rule for variables:

$$\frac{}{\langle \sigma, x \rangle \to \langle \sigma, n \rangle}\, n = \sigma(x)$$

Update your proof of progress for this extended language:

Lemma 2 (Progress for arithmetic expressions with variables). *For all arithmetic expressions A and states σ, either A is a number or there exists some A' and σ' such that $\langle \sigma, A \rangle \to \langle \sigma', A' \rangle$.*

3. Prove that the state in the language above never changes.

Lemma 3 (Purity). *For all arithmetic expressions A, A' and states σ, σ', if $\langle \sigma, A \rangle \to \langle \sigma', A' \rangle$ then $\sigma = \sigma'$.*

4. Arithmetic expressions in many languages, however, are not pure. For instance, we could add a C-style post-increment operator:

$$A ::= n \mid A + A \mid x \mid x{+}{+}$$

The small-step rules for this language are as above; we just need one more rule:

$$\frac{}{\langle \sigma, x{+}{+} \rangle \to \langle \sigma[x \mapsto n+1], n \rangle}\ \sigma(x) = n$$

Update the proof of progress once more for this language.

Lemma 4 (Progress for arithmetic expressions with side-effects)**.** *For all arithmetic expressions A and states σ, either A is a number or there exists some A' and σ' such that $\langle \sigma, A \rangle \to \langle \sigma', A' \rangle$.*