Lecture 1

Arithmetic expression

This lecture closely followed *Software foundations*, Vol. 2, on Small Step Operational Semantics.

The expressions are

$$e ::= \mathsf{const}\ n \mid \mathsf{add}\ e\ e$$

where

$$n ::= 0 \mid \mathsf{succ}\ n$$

We can define the value as a function from expressions to natural numbers

$$[\![\mathsf{const}\ n]\!] = n \qquad [\![\mathsf{add}\ e_0\ e_1]\!] = [\![e_0]\!] + [\![e_1]\!]$$

But we also can define function that refers to the *syntactic* form of an expression, for instance

$$depth\ (\mathsf{const} n) = 0 \qquad depth\ (\mathsf{add}\ e_0\ e_1) = 1 + max\ (depth\ e_0)\ (depth\ e_1)$$

We describe leftmost evaluation by the rules

$$\frac{}{\mathsf{add}\ (\mathsf{const}\ n_0)\ (\mathsf{const}\ n1) \to \mathsf{const}\ (n_0 + n_1)}(C)$$

$$\frac{e_0 \to e_0'}{\mathsf{add}\ e_0\ e_1 \to \mathsf{add}\ e_0'\ e_1}(A_0) \qquad \frac{e_1 \to e_1'}{\mathsf{add}\ (\mathsf{const}\ n)\ e_1 \to \mathsf{add}\ (\mathsf{const}\ n)\ e_1'}(A_1)$$

We say that $e \to e'$ if it is the conclusion of a *derivation tree* using these primitive inference rules.

This defines a *one step evaluation* relation.

This defines a *binary relation* on expressions

A binary relation R is said to be *deterministic* iff we have

$$\forall\ e\ e'\ e''\ \ (R\ e\ e'\ \wedge\ R\ e\ e'') \Rightarrow e' = e''$$

Theorem 0.1 *The relation defined by the rule C, A_0, A_1 is deterministic.*

The first exercise is to do the complete proof of this Theorem.

Another way to state this result is that the rule

$$\frac{e \to e' \quad e \to e''}{e' = e''}$$

is *admissible*.

We define a predicate on expressions: e is a *value* iff e is of the form $\mathsf{const}\ n$. We can describe the expressions as follows

$$e ::= v \mid \mathsf{add}\ e\ e \qquad v ::= \mathsf{const}\ n$$

and the rule (A_1) can be rewritten as

$$\frac{e_1 \to e_1'}{\mathsf{add}\ v\ e_1 \to \mathsf{add}\ v\ e_1'}(A_1)$$

Theorem 0.2 *(strong progress) For all e we have that either e is a value or $\exists\ e'\ e \rightarrow e'$.*

We say that e is in *normal form* iff we have $\neg\exists e'\ e \rightarrow e'$. It is direct to see that if e is a value then e is in normal form. The second exercise is to use strong progress to prove the following.

Theorem 0.3 *An expression is a value iff it is in normal form.*