50003 - Models of Computation - Lecture 2

Oliver Killane

15/10/21

Lecture Recording
Lecture recording is available here

Syntax of a while Language

We can define a simple while language (if, else, while loops) to build programs from & to analyse.

$$\begin{array}{rcl} B \in Bool & ::= & true|false|E = E|E < E|B\&B|\neg B \dots \\ E \in Exp & ::= & x|n|E + E|E \times E| \dots \\ C \in Com & ::= & x := E|if\ B\ then\ C\ else\ C|C;C|skip|while\ B\ do\ C \end{array}$$

Where $x \in Var$ ranges over variable identifiers, and $n \in \mathbb{N}$ ranges over natural numbers.

We can also define simple expressions (**SimpleExp**) to work on:

$$E \in SimpleExp\ ::=\ n|E + E|E \times E| \dots$$

Operational Semantics for SimpleExp

- **Small-Step** Also called structural, gives a method for evaluating an expression step-by-step.
- **Big-Step** Also called Natural, ignores intermediate steps and gives result immediately.

Big Step Semantics of SimpleExp

The properties OF $\Downarrow$ are:

- **Determinacy** For all E, n_1 and n_2 if $E \Downarrow n_1$ and $E \Downarrow n_2$ then $n_1 = n_2$
- **Totality** For all E there exists an n such that $E \Downarrow n$.

  We can break this with loops in matching, e.g

  $$\text{(B-NON-TOTAL)}\frac{}{true \Downarrow true}$$

  As a result, on hitting true will not stop.

Small Step Semantics of SimpleExp

Given a realtion $\rightarrow$ we can define a new relation $\leftarrow *$ such that:

$E \leftarrow *E'$ holds if and only if $E = E'$ or there is some finite sequence $E \rightarrow E_1 \rightarrow E_3 \rightarrow \dots \rightarrow E_k \rightarrow E'$

- **Normal Form** E is in its normal form (irreducable) if there is no E' such that $E \rightarrow E'$

  In **SimpleExp** the normal form is the natural numbers.

- **Determinacy** For all E, E_1, E_2 if $E \rightarrow E_1$ and $E \rightarrow E_2$ then $E_1 = E_2$.

  There is at most one next step.

- **Confluence** For all E, E_1, E_2 if $E \to *E_1$ and $E \to *E_2$ then there exists some E' such that $E_1 \to *E'$ and $E_2 \to *E'$.

  Determinate $\to$ Confluent.

  There are several evaluations paths, but they all get the same end result.

- **(Strong) Normalisation** There are no infinite sequences of expressions $E_1 \to E_2 \to E_3 \to \dots$ such that for all i, $E_i \to E_{i+1}$.

  Every evaluation path eventually reaches a normal form.

Theorem: for all E, n_1, n_2, if $E \to *n_1$ and $E \to *n_2$ then $n_1 = n_2$.