50003 - Models of Computation - Lecture 4

Oliver Killane

24/10/21

Lecture Recording

Lecture recording is available here

Small Step semantics

Expressions

$$\text{(W-EXP.LEFT)}\frac{\langle E_1, s\rangle \to_e \langle E_1', s'\rangle}{\langle E_1 + E_2, s\rangle \to_e \langle E_1' + E_2, s'\rangle}$$

$$\text{(W-EXP.RIGHT)}\frac{\langle E, s\rangle \to_e \langle E', s'\rangle}{\langle n + E, s\rangle \to_e \langle n + E', s'\rangle}$$

$$\text{(W-EXP.VAR)}\frac{}{\langle x, s\rangle \to_e \langle n, s\rangle}\ s(x) = n$$

$$\text{(W-EXP.ADD)}\frac{}{\langle n_1 + n_2, s\rangle \to_e \langle n_3, s\rangle}\ n_3 = n_1 + n_2$$

Assignment

$$\text{(W-ASS.EXP)}\frac{\langle E, s\rangle \to_e \langle E', s'\rangle}{\langle x := E, s\rangle \to_c \langle x := E', s'\rangle}$$

$$\text{(W-ASS.NUM)}\frac{}{\langle x := n, s\rangle \to_c \langle skip, s[x \mapsto n]\rangle}$$

Sequential Composition

$$\text{(W-SEQ.LEFT)}\frac{\langle C_1, s\rangle \to_c \langle C_1', s'\rangle}{\langle C_1; C_2, s\rangle \to_c \langle C_1'; C_2, s'\rangle}$$

$$\text{(W-SEQ.SKIP)}\frac{}{\langle skip; C, s\rangle \to_c \langle C, s\rangle}$$

Conditional

$$\text{(W-COND.TRUE)}\frac{}{\langle \text{if } true \text{ then } C_1 \text{ else } C_2, s\rangle \to_c \langle C_1, s\rangle}$$

$$\text{(W-COND.FALSE)}\frac{}{\langle \text{if } false \text{ then } C_1 \text{ else } C_2, s\rangle \to_c \langle C_2, s\rangle}$$

$$\text{(W-COND.BEXP)}\frac{\langle B, s\rangle \to_b \langle B', s'\rangle}{\langle \text{if } B \text{ then } C_1 \text{ else } C_2, s\rangle \to_c \langle \text{if } B' \text{ then } C_1 \text{ else } C_2, s'\rangle}$$

While

$$\text{(W-WHILE)}\frac{}{\langle \text{while } B \text{ do } C, s\rangle \to_c \langle \text{if } B \text{ then } (C; \text{while } B \text{ do } C) \text{ else } skip, s\rangle}$$

Determinacy and Confluence

The execution relation ($\rightarrow_c$) is deterministic.

$$\forall C, C_1, C_2 \in Com \forall s, s_1, s_2.[\langle C, s\rangle \rightarrow_c \langle C_1, s_1\rangle \wedge \langle C, s\rangle \rightarrow_c \langle C_2, s_2\rangle \rightarrow \langle C_1, s_1\rangle = \langle C_2, s_2\rangle]$$

Hence the relation is also confluent:

$$\forall C, C_1, C_2 \in Com \forall s, s_1, s_2.[\langle C, s\rangle \rightarrow_c \langle C_1, s_1\rangle \wedge \langle C, s\rangle \rightarrow_c \langle C_2, s_2\rangle \rightarrow$$

$$\exists C' \in Com, s'.[\langle C_1, s_1\rangle \rightarrow_c \langle C', s'\rangle \wedge \langle C_2, s_2\rangle \rightarrow_c \langle C', s'\rangle]]$$

Both also hold for $\rightarrow_e$ and $\rightarrow_b$.

Answer Configuration

A configuration $\langle skip, s\rangle$ is an **answer configuration**. As there is no rule to execute skip, it is a normal form.

$$\neg\exists C \in Com, s, s'.[\langle skip, s\rangle \rightarrow_c \langle C, s'\rangle]$$

For booleans $\langle true, s\rangle$ and $\langle false, s\rangle$ are answer configurations, and for expressions $\langle n, s\rangle$.

Stuck Configurations

A configuration that cannot be evaluated to a normal form is called a **suck configuration**.

$$\langle y, (x \mapsto 3)\rangle$$

Note that a configuration that leads to a **stuck configuration** is not itself stuck.

$$\langle 5 < y, (x \mapsto 2)\rangle$$

(Not stuck, but reduces to a stuck state)

Normalising

The relations $\rightarrow_b$ and $\rightarrow_e$ are normalising, but $\rightarrow_c$ is not as it may not have a normal form.

$$\text{while } true \text{ do } skip$$

$$\langle \text{while } true \text{ do } skip, s\rangle \rightarrow_c^3 \langle \text{while } true \text{ do } skip, s\rangle$$

($\rightarrow_c^3$ means 3 steps, as we have gone through more than one to get the same configuration, it is an infinite loop)

Side Effecting Expressions

If we allow programs such as:

$$\text{do } x := x + 1 \text{ } return \text{ } x$$

$$(\text{do } x := x + 1 \text{ } return \text{ } x) + (\text{do } x := x \times 1 \text{ } return \text{ } x)$$

(value depends on evaluation order)

Short Circuit Semantics

$$\frac{B_1 \to_b B_1'}{B_1 \& B_2 \to_b B_1' \& B_2}$$

$$\frac{}{false \& B \to_b false}$$

$$\frac{}{true \& B \to_b B}$$

Strictness

An operation is **strict** when arguments must be evaluated before the operation is evaluated. Addition is struct as both expressions must be evaluated (left, then right).

Due to short circuiting, & is left strict as it is possible for the operation to be evaluated without evaluating the right (**non-strict** in right argument).

Factorial Program

$$C = y := x; a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$$

We can attempt to evaluate this for a given input, for example:

$$s = [x \mapsto 3, y \mapsto 17, z \mapsto 42]$$

The evaluation path is as follows:

Start

$$\langle y := x; a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), [x \mapsto 3, y \mapsto 17, z \mapsto 42] \rangle$$

Get x variable

where $C = a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 17, z \mapsto 42)$:

$$\text{(W-SEQ.LEFT)} \frac{\text{(W-ASS.EXP)} \dfrac{\text{(W-EXP.VAR)} \dfrac{}{\langle x, s \rangle \to_e \langle 3, s \rangle}}{\langle y := x, s \rangle \to_c \langle y := 3, s \rangle}}{\langle y := x; C, s \rangle \to_c \langle y := 3; C, s \rangle}$$

Result:

$$\langle y := 3; a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 17, z \mapsto 42) \rangle$$

Assign to y variable

where $C = a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 17, z \mapsto 42)$:

$$\text{(W-SEQ.LEFT)} \frac{\text{(W-ASS.NUM)} \dfrac{}{\langle y := 3, s \rangle \to_c \langle skip, s[y \mapsto 3] \rangle}}{\langle y := 3; C, s \rangle \to_c \langle skip; C, s[y \mapsto 3] \rangle}$$

Result:

$$\langle skip; a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42) \rangle$$

Eliminate skip

where $C = a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42)$:

$$\text{(W-SEQ.SKIP)}\frac{}{\langle skip; C, s\rangle \rightarrow_c \langle C, s\rangle}$$

Result:

$$\langle a := 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42)\rangle$$

Assign a

where $C = \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.NUM)}\dfrac{}{\langle a := 1, s\rangle \rightarrow_c \langle skip, s[a \mapsto 1]\rangle}}{\langle a := 1; C, s\rangle \rightarrow_c \langle skip; C, s[a \mapsto 1]\rangle}$$

Result:

$$\langle skip; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Eliminate skip

where $C = \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$

$$\text{(W-SEQ.SKIP)}\frac{}{\langle skip; C, s\rangle \rightarrow_c \langle C, s\rangle}$$

Result:

$$\langle \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Expand while

where $C = (a := a \times y; y := y - 1)$, $B = 0 < y$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-WHILE)}\frac{}{\langle \text{while } B \text{ do } C, s\rangle \rightarrow_c \langle \text{if } B \text{ then } (C; \text{while } B \text{ do } C) \text{ else } skip, s\rangle}$$

Result:

$$\langle \text{if } 0 < y \text{ then } (a := a{\times}y; y := y{-}1; \text{while } 0 < y \text{ do } a := a{\times}y; y := y{-}1) \text{ else } skip, (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Get y variable

where $C = (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-COND.BEXP)}\frac{\text{(W-BOOL.LESS.RIGHT)}\dfrac{\text{(W-EXP.VAR)}\dfrac{}{\langle y, s\rangle \rightarrow \langle 3, s\rangle}}{\langle 0 < y, s\rangle \rightarrow_b \langle 0 < 3, s\rangle}}{\langle \text{if } 0 < y \text{ then } (C; \text{while } 0 < y \text{ do } C) \text{ else } skip, s\rangle \rightarrow_c \langle \text{if } 0 < 3 \text{ then } (C; \text{while } 0 < y \text{ do } C) \text{ else } skip,}$$

Result:

$$\langle \text{if } 0 < 3 \text{ then } (a := a{\times}y; y := y{-}1; \text{while } 0 < y \text{ do } a := a{\times}y; y := y{-}1); \text{ else } skip, (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Complete if boolean

where $C = (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-COND.EXP)}\frac{\text{(W-BOOl.LESS.TRUE)}\dfrac{}{\langle 0 < 3, s\rangle \to_b \langle \mathit{true}, s\rangle}}{\langle \text{if } 0 < 3 \text{ then } (C; \text{while } 0 < y \text{ do } C) \text{ else } \mathit{skip}, s\rangle \to_c \langle \text{if } \mathit{true} \text{ then } (C; \text{while } 0 < y \text{ do } C) \text{ else } \mathit{skip}, s\rangle}$$

Result:

$$\langle \text{if } \mathit{true} \text{ then } (a := a{\times}y; y := y{-}1; \text{while } 0 < y \text{ do } a := a{\times}y; y := y{-}1);\ \text{else } \mathit{skip}, (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Evaluate if

where $C = (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-COND.TRUE)}\frac{}{\langle \text{if } \mathit{true} \text{ then } (C; \text{while } 0 < y \text{ do } C) \text{ else } \mathit{skip}, s\rangle \to_c \langle C; \text{while } 0 < y \text{ do } C, s\rangle}$$

Result:

$$\langle a := a \times y; y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Evaluate Expression a

where $C = y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.EXP)}\dfrac{\text{(W-EXP.MUL.LEFT)}\dfrac{\text{(W-EXP.VAR)}\dfrac{}{\langle a, s\rangle \to \langle 1, s\rangle}}{\langle a \times y, s\rangle \to_e \langle 1 \times y, s\rangle}}{\langle a := a \times y, s\rangle \to_c \langle a := 1 \times y, s\rangle}}{\langle a := a \times y; C, s\rangle \to_c \langle a := 1 \times y; C, s\rangle}$$

Result:

$$\langle a := 1 \times y; y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Evaluate Expression y

where $C = y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.EXP)}\dfrac{\text{(W-EXP.MUL.RIGHT)}\dfrac{\text{(W-EXP.VAR)}\dfrac{}{\langle y, s\rangle \to_e \langle 3, s\rangle}}{\langle 1 \times y, s\rangle \to_e \langle 1 \times 3, s\rangle}}{\langle a := 1 \times y, s\rangle \to_c \langle a := 1 \times 3, s\rangle}}{\langle a := 1 \times y; C, s\rangle \to \langle a := 1 \times 3; C, s\rangle}$$

Result:

$$\langle a := 1 \times 3; y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Evaluate Multiply

where $C = y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.EXP)}\dfrac{\text{(W-EXP.MUL)}\dfrac{}{\langle 1 \times 3, s\rangle \to_e \langle 3, s\rangle}}{\langle a := 1 \times 3, s\rangle \to_c \langle a := 3, s\rangle}}{\langle a := 1 \times 3; C, s\rangle \to_c \langle a := 3; C, s\rangle}$$

Result:

$$\langle a := 3; y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)\rangle$$

Assign 3 to a

where $C = y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.NUM)}\dfrac{}{\langle a := 3, s\rangle \to_c \langle \mathit{skip}, s[a \mapsto 3]\rangle}}{\langle a := 3; C, s\rangle \to_c \langle \mathit{skip}; C, s[a \mapsto 3]\rangle}$$

Result:

$$\langle \mathit{skip}; y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)\rangle$$

Eliminate Skip

where $C = y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)$:

$$\text{(W-SEQ.SKIP)}\frac{}{\langle \mathit{skip}; C, s\rangle \to_c \langle C, s\rangle}$$

Result:

$$\langle y := y - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)\rangle$$

Assign 3 to y

where $C = \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.EXP)}\dfrac{\text{(W-EXP.SUB.LEFT)}\dfrac{\text{(W-EXP.VAR)}\dfrac{}{\langle y, s\rangle \to \langle 3, s\rangle}}{\langle y - 1, s\rangle \to_e \langle 3 - 1, s\rangle}}{\langle y := y - 1, s\rangle \to_c \langle y := 3 - 1, s\rangle}}{\langle y := y - 1; C, s\rangle \to_c \langle y := 3 - 1, s\rangle}$$

Result:

$$\langle y := 3 - 1; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)\rangle$$

Evaluate Subtraction

where $C = \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.EXP)}\dfrac{\text{(W-EXP.SUB)}\dfrac{}{\langle 3-1, s\rangle \to_e \langle 2, s\rangle}}{\langle y := 3-1, s\rangle \to_c \langle y := 2, s\rangle}}{\langle y := 3-1; C, s\rangle \to_c \langle y := 2; C, s\rangle}$$

Result:

$$\langle y := 2; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)\rangle$$

Assign 2 to y

where $C = \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)$:

$$\text{(W-SEQ.LEFT)}\frac{\text{(W-ASS.NUM)}\dfrac{}{\langle y := 2, s\rangle \to_c \langle \mathit{skip}, s[y \mapsto 2]\rangle}}{\langle y := 2; C, s\rangle \to_c \langle \mathit{skip}; C, s[y \mapsto 2]\rangle}$$

Result:

$$\langle \mathit{skip}; \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 2, z \mapsto 42, a \mapsto 3)\rangle$$

Eliminate skip

where $C = \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1)$ and $s = (x \mapsto 3, y \mapsto 2, z \mapsto 42, a \mapsto 3)$:

$$\text{(W-SEQ.SKIP)}\frac{}{\langle \mathit{skip}; C, s\rangle \to_c \langle C, s\rangle}$$

Result:

$$\langle \text{while } 0 < y \text{ do } (a := a \times y; y := y - 1), (x \mapsto 3, y \mapsto 2, z \mapsto 42, a \mapsto 3)\rangle$$