

50003 - Models of Computation - (Dr Raad) Lecture 4

Oliver Killane

24/10/21

Small Step semantics

Expressions

$$\begin{aligned}
& \text{(W-EXP.LEFT)} \frac{\langle E_1, s \rangle \rightarrow_e \langle E'_1, s' \rangle}{\langle E_1 + E_2, s \rangle \rightarrow_e \langle E'_1 + E_2, s' \rangle} \\
& \text{(W-EXP.RIGHT)} \frac{\langle E, s \rangle \rightarrow_e \langle E', s' \rangle}{\langle n + E, s \rangle \rightarrow_e \langle n + E', s' \rangle} \\
& \text{(W-EXP.VAR)} \frac{}{\langle x, s \rangle \rightarrow_e \langle n, s \rangle} \quad s(x) = n \\
& \text{(W-EXP.ADD)} \frac{}{\langle n_1 + n_2, s \rangle \rightarrow_e \langle n_3, s \rangle} \quad n_3 = n_1 + n_2
\end{aligned}$$

Assignment

$$\begin{aligned}
& \text{(W-ASS.EXP)} \frac{\langle E, s \rangle \rightarrow_e \langle E', s' \rangle}{\langle x := E, s \rangle \rightarrow_c \langle x := E', s' \rangle} \\
& \text{(W-ASS.NUM)} \frac{}{\langle x := n, s \rangle \rightarrow_c \langle \text{skip}, s[x \mapsto n] \rangle}
\end{aligned}$$

Sequential Composition

$$\begin{aligned}
& \text{(W-SEQ.LEFT)} \frac{\langle C_1, s \rangle \rightarrow_c \langle C'_1, s' \rangle}{\langle C_1; C_2, s \rangle \rightarrow_c \langle C'_1; C_2, s' \rangle} \\
& \text{(W-SEQ.SKIP)} \frac{}{\langle \text{skip}; C, s \rangle \rightarrow_c \langle C, s \rangle}
\end{aligned}$$

Conditional

$$\begin{aligned}
& \text{(W-COND.TRUE)} \frac{}{\langle \text{if } \text{true} \text{ then } C_1 \text{ else } C_2, s \rangle \rightarrow_c \langle C_1, s \rangle} \\
& \text{(W-COND.FALSE)} \frac{}{\langle \text{if } \text{false} \text{ then } C_1 \text{ else } C_2, s \rangle \rightarrow_c \langle C_2, s \rangle} \\
& \text{(W-COND.BEXP)} \frac{\langle B, s \rangle \rightarrow_b \langle B', s' \rangle}{\langle \text{if } B \text{ then } C_1 \text{ else } C_2, s \rangle \rightarrow_c \langle \text{if } B' \text{ then } C_1 \text{ else } C_2, s' \rangle}
\end{aligned}$$

While

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

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:

$$\begin{aligned} \forall C, C_1, C_2 \in Com \forall s, s_1, s_2. [\langle C, s \rangle \rightarrow_c \langle C_1, s_1 \rangle \wedge \langle C, s \rangle \rightarrow_c \langle C_2, s_2 \rangle \rightarrow \\ \exists C' \in Com, s'. [\langle C_1, s_1 \rangle \rightarrow_c \langle C', s' \rangle \wedge \langle C_2, s_2 \rangle \rightarrow_c \langle C', s' \rangle]] \end{aligned}$$

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 **suck 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.

while *true* 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:

do $x := x + 1$ return x

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

(value depends on evaluation order)

Short Circuit Semantics

$$\frac{B_1 \rightarrow_b B'_1}{B_1 \& B_2 \rightarrow_b B'_1 \& B_2}$$

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

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

Strictness

An operation is **strict** when arguments must be evaluated before the operation is evaluated. Addition is strict 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 $s = (x \mapsto 3, y \mapsto 17, z \mapsto 42)$:

$$\frac{\frac{(W-SEQ.LEFT) \frac{(W-ASS.EXP) \frac{(W-EXP.VAR) \overline{\langle x, s \rangle \rightarrow_e \langle 3, s \rangle}}{\langle y := x, s \rangle \rightarrow_c \langle y := 3, s \rangle}}{\langle y := x; C, s \rangle \rightarrow_c \langle y := 3; C, s \rangle}}{\langle y := x; C, s \rangle \rightarrow_c \langle y := 3; C, s \rangle}}$$

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 $s = (x \mapsto 3, y \mapsto 17, z \mapsto 42)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

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

Result:

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

Evaluate if

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

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 1)$:

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

Result:

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

Eliminate Skip

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

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)$:

$$\text{(W-SEQ.LEFT)} \frac{\text{(W-ASS.EXP)} \frac{\text{(W-EXP.SUB.LEFT)} \frac{\text{(W-EXP.VAR)} \overline{\langle y, s \rangle \rightarrow \langle 3, s \rangle}}{\langle y - 1, s \rangle \rightarrow_e \langle 3 - 1, s \rangle}}{\langle y := y - 1, s \rangle \rightarrow_c \langle y := 3 - 1, s \rangle}}{\langle y := y - 1; C, s \rangle \rightarrow_c \langle y := 3 - 1, s \rangle}$$

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)$:

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

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 $s = (x \mapsto 3, y \mapsto 3, z \mapsto 42, a \mapsto 3)$:

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

Result:

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

Eliminate skip

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

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

Result:

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