

2. Exercise sheet *Semantics and Verification of Software WS0809*

Due to Monday, 3 Nov. 2008, before the exercise course begins.

Exercise 2.1:

(2 points)

Show that the bigstep relation and the singlestep relation on arithmetic expressions, as defined in Exercise 1.2, are equivalent, i.e., that for every $a \in AExp$, $\delta \in \Sigma$, and $z \in \mathbb{Z}$:

$$\langle a, \delta \rangle \rightarrow z \quad \text{iff} \quad \langle a, \delta \rangle \rightarrow_1^* z.$$

Solution

By induction on the structure of a :

$$\langle a, \delta \rangle \rightarrow z \iff \langle a, \delta \rangle \rightarrow_1^* z$$

for all $a ::= z \mid x \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \in AExp$

$$\text{I.B. } a = z : \quad \begin{array}{l} \langle z, \delta \rangle \rightarrow z \\ \langle z, \delta \rangle \rightarrow_1 z \end{array}$$

$$\text{I.B. } a = x : \quad \begin{array}{l} \langle x, \delta \rangle \rightarrow z \\ \iff z = \delta(x) \\ \iff \langle x, \delta \rangle \rightarrow_1 \langle \delta(x), \delta \rangle = \langle z, \delta \rangle \rightarrow_1 z \end{array}$$

$$\text{I.S. } a = a_1 + a_2 : \quad \begin{array}{l} \langle a_1 + a_2, \delta \rangle \rightarrow z \\ \iff \exists z_1, z_2 \in \mathbb{Z} : \\ \quad \langle a_1, \delta \rangle \rightarrow z_1, \\ \quad \langle a_2, \delta \rangle \rightarrow z_2, \\ \quad z = z_1 + z_2 \\ \iff \langle a_1 + a_2, \delta \rangle \\ \quad \rightarrow_1^* \langle z_1 + a_2, \delta \rangle \quad \text{where } \langle a_1, \delta \rangle \xrightarrow_1^* \langle z_1, \delta \rangle \quad \text{I.H.} \\ \quad \rightarrow_1^* \langle z_1 + z_2, \delta \rangle \quad \text{where } \langle a_2, \delta \rangle \xrightarrow_1^* \langle z_2, \delta \rangle \quad \text{I.H.} \\ \quad \rightarrow_1 \langle z, \delta \rangle \rightarrow_1 z \quad \text{where } z = z_1 + z_2 \end{array}$$

Difference and product analogously.

□

Exercise 2.2:

(1+2 points)

- Write the WHILE program for $x \bmod y$;
- Construct the derivation tree for the operational semantics for the above program starting in a state $\sigma \in \Sigma$ with $\sigma(x) = 7$ and $\sigma(y) = 3$.

Solution

(a) The WHILE program for $x \bmod y$ is $\mathbf{while} \overbrace{y \leq x}^b \mathbf{do} \overbrace{x := x - y}^c$

(b) The derivation tree is as follows:
 (Notation: $\sigma_{i,j} : \sigma(x) = i, \sigma(y) = j$)

$$\begin{array}{c}
 \frac{\frac{\frac{\langle y, \sigma_{7,3} \rangle \rightarrow 3}{\langle b, \sigma_{7,3} \rangle \rightarrow \mathbf{true}}}{\langle x, \sigma_{7,3} \rangle \rightarrow 7} \quad \frac{\frac{\frac{\langle x, \sigma_{7,3} \rangle \rightarrow 7 \quad \langle y, \sigma_{7,3} \rangle \rightarrow 3}{\langle x - y, \sigma_{7,3} \rangle \rightarrow 4}}{\langle c, \sigma_{7,3} \rangle \rightarrow \sigma_{4,3}}}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma_{7,3} \rangle \rightarrow \sigma_{1,3}} \\
 \frac{\frac{\frac{\frac{\langle y, \sigma_{4,3} \rangle \rightarrow 3 \quad \langle x, \sigma_{4,3} \rangle \rightarrow 4}{\langle b, \sigma_{4,3} \rangle \rightarrow \mathbf{true}}}{\langle c, \sigma_{4,3} \rangle \rightarrow \sigma_{1,3}}}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma_{4,3} \rangle \rightarrow \sigma_{1,3}} \\
 \frac{\frac{\frac{\frac{\langle x, \sigma_{4,3} \rangle \rightarrow 4 \quad \langle y, \sigma_{4,3} \rangle \rightarrow 3}{\langle x - y, \sigma_{4,3} \rangle \rightarrow 1}}{\langle c, \sigma_{4,3} \rangle \rightarrow \sigma_{1,3}}}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma_{1,3} \rangle \rightarrow \sigma_{1,3}} \\
 \frac{\frac{\frac{\langle y, \sigma_{1,3} \rangle \rightarrow 3 \quad \langle x, \sigma_{1,3} \rangle \rightarrow 1}{\langle b, \sigma_{1,3} \rangle \rightarrow \mathbf{false}}}{\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma_{1,3} \rangle \rightarrow \sigma_{1,3}}
 \end{array}$$

□

Exercise 2.3:**(3 points)**

Show that the statement $c \in \text{Cmd}$ given by

$$y := 1; \text{while } \neg(x = 1) \text{ do } (y := y + x; x := x - 1)$$

computes the sum, i.e., that its operational semantics satisfies the following condition:

$$\mathfrak{D}[[c]](\sigma)(y) = \sum_{m=1}^{\sigma(x)} m$$

for every $\sigma \in \Sigma$ with $\sigma(x) \geq 1$.

Solution

$$c = y := 1; \text{while } \overbrace{\neg(x = 1)}^b \text{ do } \overbrace{(y := y + x; x := x - 1)}^{c_0}$$

To prove: $\mathfrak{D}[[c]](\sigma)(y) = \sum_{m=1}^{\sigma(x)} m$ for each $\sigma \in \Sigma$ with $\sigma(x) \geq 1$ (otherwise non-terminating)

Notation: $\sigma_{i,j} : \sigma(x) = i, \sigma(y) = j$

First, we analyse **while** b **do** c_0 separately for $i, j \geq 1$. By induction over i , we will show:

$$\forall i, j \geq 1 : \langle \text{while } b \text{ do } c_0, \sigma_{i,j} \rangle \rightarrow \sigma_{1, j-1 + \sum_{m=1}^i m} \quad (\star)$$

(I.B.) $i = 1$:

$$\frac{\frac{\frac{\langle x, \sigma_{1,j} \rangle \rightarrow 1 \quad \langle 1, \sigma_{1,j} \rangle \rightarrow 1}{\langle x = 1, \sigma_{1,j} \rangle \rightarrow \text{true}}}{\langle b, \sigma_{1,j} \rangle \rightarrow \text{false}}}{\langle \text{while } b \text{ do } c_0, \sigma_{1,j} \rangle \rightarrow \sigma_{1,j}}$$

Notice that the base case holds because for $i = 1$ we get $\sigma_{i, j-1 + \sum_{m=1}^i m} = \sigma_{1,j}$.

(I.S.) $i \rightarrow i + 1$:

$$\frac{\frac{\frac{\frac{\langle y, \sigma_{i+1,j} \rangle \rightarrow j \quad \langle x, \sigma_{i+1,j} \rangle \rightarrow i+1}{\langle y + x, \sigma_{i+1,j} \rangle \rightarrow j + (i+1)}}{\langle c_1, \sigma_{i+1,j} \rangle \rightarrow \sigma_{i+1, j+(i+1)}}}{\langle c_2, \sigma_{i+1, j+(i+1)} \rangle \rightarrow \sigma_{i, j+(i+1)}}}{\langle c_0, \sigma_{i+1,j} \rangle \rightarrow \sigma_{i, j+(i+1)}} \quad (\star)$$

$$\frac{\frac{\frac{\langle x, \sigma_{i+1,j} \rangle \rightarrow i+1 \quad \langle 1, \sigma_{i+1,j} \rangle \rightarrow 1}{\langle x = 1, \sigma_{i+1,j} \rangle \rightarrow \text{false}}}{\langle \neg(x = 1), \sigma_{i+1,j} \rangle \rightarrow \text{true}} \quad (\star) \quad \langle \text{while } b \text{ do } c_0, \sigma_{i, j+(i+1)} \rangle \rightarrow \sigma_{1, j+(i+1)-1 + \sum_{m=1}^i m} \quad (\text{I.H.})}{\langle \text{while } b \text{ do } c_0, \sigma_{i+1,j} \rangle \rightarrow \sigma_{1, j-1 + \sum_{m=1}^{i+1} m}}$$

For $j = 1$ we get $\sigma(y) = \sum_{m=1}^i m$ is the sum of the first i natural numbers. This is used when analysing c .

Analysis of c :

$$\frac{\frac{\langle 1, \sigma_{i,j} \rangle \rightarrow 1}{\langle y := 1, \sigma_{i,j} \rangle \rightarrow \sigma_{i,1}} \quad \langle \text{while } b \text{ do } c_0, \sigma_{i,1} \rangle \rightarrow \sigma_{1, \sum_{m=1}^i m} \quad (\star)}{\langle y := 1; \text{while } b \text{ do } c_0, \sigma_{i,j} \rangle \rightarrow \sigma_{1, \sum_{m=1}^i m}}$$

Thus for any σ with $\sigma(y) \geq 1$: $\mathcal{D}[[c]](\sigma)(y) = \sum_{m=1}^{\sigma(x)} m$.

□

Exercise 2.4:

(1+2+2 points)

(a) Extend the WHILE language by a loop construct of the form

repeat c until b

and define its execution relation \rightarrow without (explicitly) using the **while** statement.

(b) Establish the following semantic equivalence:

repeat c until b \sim c ; if b then skip else (repeat c until b).

(c) Establish the following semantic equivalence:

repeat c until b \sim c ; while $\neg b$ do c .

Solution

$$(a) \text{ (Rt)} \quad \frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \langle b, \sigma' \rangle \rightarrow \mathbf{true}}{\langle \mathbf{repeat } c \text{ until } b, \sigma \rangle \rightarrow \sigma'}$$

$$(Rf) \quad \frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \langle b, \sigma' \rangle \rightarrow \mathbf{false} \quad \langle \mathbf{repeat } c \text{ until } b, \sigma' \rangle \rightarrow \sigma''}{\langle \mathbf{repeat } c \text{ until } b, \sigma \rangle \rightarrow \sigma''}$$

(b) Two cases: $\langle b, \sigma' \rangle \rightarrow \mathbf{true}$ and $\langle b, \sigma' \rangle \rightarrow \mathbf{false}$

Abbreviation: $r = \mathbf{repeat } c \text{ until } b$

$$\frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \langle b, \sigma' \rangle \rightarrow \mathbf{true}}{\langle r, \sigma \rangle \rightarrow \sigma'} \iff \frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \frac{\langle b, \sigma' \rangle \rightarrow \mathbf{true} \quad \langle \mathbf{skip}, \sigma' \rangle \rightarrow \sigma'}{\langle \mathbf{if } b \text{ then skip else } r, \sigma' \rangle \rightarrow \sigma'}}{\langle c; \mathbf{if } b \text{ then skip else } r, \sigma \rangle \rightarrow \sigma'}$$

$$\frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \langle b, \sigma' \rangle \rightarrow \mathbf{false} \quad \langle r, \sigma' \rangle \rightarrow \sigma''}{\langle r, \sigma \rangle \rightarrow \sigma''} \iff \frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \frac{\langle b, \sigma' \rangle \rightarrow \mathbf{false} \quad \langle r, \sigma' \rangle \rightarrow \sigma''}{\langle \mathbf{if } b \text{ then skip else } r, \sigma' \rangle \rightarrow \sigma''}}{\langle c; \mathbf{if } b \text{ then skip else } r, \sigma \rangle \rightarrow \sigma''}$$

(c) Induction over proof tree:

Abbreviation: $w = \mathbf{while } \neg b \text{ do } c$

$$\frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \langle b, \sigma' \rangle \rightarrow \mathbf{true}}{\langle r, \sigma \rangle \rightarrow \sigma'} \Rightarrow \frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \frac{\frac{\langle b, \sigma' \rangle \rightarrow \mathbf{true}}{\langle \neg b, \sigma' \rangle \rightarrow \mathbf{false}} \quad \langle w, \sigma' \rangle \rightarrow \sigma'}{\langle w, \sigma' \rangle \rightarrow \sigma'}}{\langle c; w, \sigma \rangle \rightarrow \sigma'}$$

Induction hypothesis: $\langle r, \sigma' \rangle \rightarrow \sigma''' \Rightarrow \langle c; w, \sigma' \rangle \rightarrow \sigma'''$

$$\frac{\langle c, \sigma \rangle \rightarrow \sigma'' \quad \langle b, \sigma'' \rangle \rightarrow \mathbf{false} \quad \langle r, \sigma'' \rangle \rightarrow \sigma''' \quad (\text{I.H.})}{\langle r, \sigma \rangle \rightarrow \sigma'''}$$

$$\Rightarrow \frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \frac{\frac{\langle b, \sigma' \rangle \rightarrow \mathbf{false}}{\langle \neg b, \sigma' \rangle \rightarrow \mathbf{true}} \quad \langle c, \sigma' \rangle \rightarrow \sigma'' \quad \langle w, \sigma'' \rangle \rightarrow \sigma''' \text{ (I.H.)}}{\langle w, \sigma' \rangle \rightarrow \sigma'''}}{\langle c; w, \sigma \rangle \rightarrow \sigma'''}$$

“ \Leftarrow ” : analogously

_____ \square