

EXERCISE 1

(a) The program computes the quotient (and remainder) of the integer division x/y . At the end of the execution z holds the quotient and x the remainder.

(b) See page 2. For the definition of the derivation tree we use the following notation:

body $\triangleq z := z + 1; x := x - y$
 while ... $\triangleq$ while $(y \leq x)$ do {body}

EXERCISE 2

(rep-t) $\frac{\langle c, \sigma \rangle \rightarrow \sigma' \quad \langle b, \sigma' \rangle \rightarrow \text{true}}{\langle \text{repeat } c \text{ until } b, \sigma \rangle \rightarrow \sigma'}$

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

EXERCISE 4

Assume $\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma'$. This derivation could have been obtained by the application either of rule wh-f or rule wh-t. In the first case we have $\sigma' = \sigma$ and $\langle b, \sigma \rangle \rightarrow \text{false}$, and we are done. In the second case we have $\langle c, \sigma \rangle \rightarrow \sigma^*$ and $\langle \text{while } b \text{ do } c, \sigma^* \rangle \rightarrow \sigma'$ for some (intermediate) state σ^* . Then by applying our inductive hypothesis in $\langle \text{while } b \text{ do } c, \sigma^* \rangle \rightarrow \sigma'$ we obtain $\langle b, \sigma' \rangle \rightarrow \text{false}$ and we are done.

(seq)

(ssg)

(ssg)

(ssg)

(ssg)

$$\langle z := z+1, \sigma[z \mapsto 0] \rangle \rightarrow \langle [z \mapsto 1] \rangle \langle x := x-y, \sigma[z \mapsto 1] \rangle \rightarrow \sigma[z, x \mapsto 1, \delta]$$

(seq)

$$\langle \text{body}, \sigma[z \mapsto 0] \rangle \rightarrow \sigma[z, x \mapsto 1, \delta]$$

(while)

$$\langle \text{while } \dots, \sigma[z \mapsto 0] \rangle \rightarrow \sigma[z, x \mapsto 1, \delta]$$

(while)

$$\langle \text{body}, \sigma[z, x \mapsto 1, \delta] \rangle \rightarrow \sigma[z, x \mapsto 2, \delta]$$

$$\langle \text{while } \dots, \sigma[z, x \mapsto 1, \delta] \rangle \rightarrow \sigma[z, x \mapsto 2, \delta]$$

(while)

$$\langle \text{while } \dots, \sigma[z, x \mapsto 2, \delta] \rangle \rightarrow \sigma[z, x \mapsto 3, \delta]$$

(while)

$$\langle \text{while } \dots, \sigma[z, x \mapsto 3, \delta] \rangle \rightarrow \sigma[z, x \mapsto 4, \delta]$$

(seq)

$$\langle z := 0, \sigma \rangle \rightarrow \sigma[z \mapsto 0]$$

(while)

$$\langle \text{while } \dots, \sigma[z \mapsto 0] \rangle \rightarrow \sigma[z, x \mapsto 2, \delta]$$

(while)

$$\langle \text{while } \dots, \sigma[z, x \mapsto 2, \delta] \rangle \rightarrow \sigma[z, x \mapsto 3, \delta]$$

$$\langle z := 0; \text{while } (y \leq \delta) \text{ do } \{ \dots \}, \sigma \rangle \rightarrow \sigma[z, x \mapsto 2, \delta]$$

EXERCISE 5

3

$$(a) \text{ mod } (\text{skip}) = \emptyset$$

$$\text{mod } (x := a) = \{x\}$$

$$\text{mod } (C_1; C_2) = \text{mod } (C_1) \cup \text{mod } (C_2)$$

$$\text{mod } (\text{repeat } C \text{ until } b) = \text{mod } (C)$$

$$(b) \text{ dep } (\text{skip}) = \emptyset$$

$$\text{dep } (x := a) = \text{fv}(a)$$

$$\text{dep } (C_1; C_2) = \text{dep } (C_1) \cup \text{dep } (C_2)$$

$$\text{dep } (\text{repeat } C \text{ until } b) = \text{dep } (C) \cup \text{fv}(b)$$

(c) the proof proceeds by structural induction on the derivation tree of $\langle C, \Gamma_1 \rangle \rightarrow \Gamma_1'$ (*)

(*) was derived with rule (skip). In this case we have $C = \text{skip}$. Then $\text{mod } (C) = \emptyset$ and $\Gamma_1' = \Gamma_2'$ holds immediately.

(*) was derived with rule (assign). In this case we have $C = x := a$ for some program variable x and arithmetic expression a , $\Gamma_1' = \Gamma_1[x \mapsto z_1]$, $\Gamma_2' = \Gamma_2[x \mapsto z_2]$ where $\langle a, \Gamma_1 \rangle \rightarrow z_1$ and $\langle a, \Gamma_2 \rangle \rightarrow z_2$, $\text{dep } (C) = \text{fv}(a)$ and $\text{mod } (C) = \{x\}$. We have to prove that $\Gamma_1[x \mapsto z_1] =_{\{x\}} \Gamma_2[x \mapsto z_2]$, which amounts to showing $z_1 = z_2$. By hypothesis we know that $\Gamma_1 =_{\text{fv}(a)} \Gamma_2$, and then by Lemma 2.6 we conclude $z_1 = z_2$.

(*) was derived with rule (seq). In this case we have $C = C_1; C_2$ for some formulas C_1 and C_2 , $\text{dep}(C) = \text{dep}(C_1) \cup \text{dep}(C_2)$, $\text{mod}(C) = \text{mod}(C_1) \cup \text{mod}(C_2)$ and there exist σ_1^*, σ_2^* such that

$$\frac{\langle C_1, \sigma_1 \rangle \rightarrow \sigma_1^* \quad \langle C_2, \sigma_1^* \rangle \rightarrow \sigma_1'}{\langle C_1; C_2, \sigma_1 \rangle \rightarrow \sigma_1'} \quad \frac{\langle C_1, \sigma_2 \rangle \rightarrow \sigma_2^* \quad \langle C_2, \sigma_2^* \rangle \rightarrow \sigma_2'}{\langle C_1; C_2, \sigma_2 \rangle \rightarrow \sigma_2'}$$

By applying the IH on $\langle C_1, \sigma_1 \rangle \rightarrow \sigma_1^*$ we obtain $\sigma_1^* =_{\text{mod}(C_1)} \sigma_2^*$. Now we will show that we can strengthen $\sigma_1^* =_{\text{mod}(C_1)} \sigma_2^*$ to $\sigma_1^* =_R \sigma_2^*$ where $R = \text{mod}(C_1) \cup \text{dep}(C_2)$. Let $R' \triangleq \text{dep}(C_2) \setminus \text{mod}(C_1)$.

By hypothesis we know

that $(*)_1 \sigma_1 =_{R'} \sigma_2$ and by the sequencing lemma we have $(*)_2 \sigma_1 =_{R'} \sigma_1^*$ and $(*)_3 \sigma_2 =_{R'} \sigma_2^*$. Combining $(*)_1$, $(*)_2$ and $(*)_3$ we obtain

$$\sigma_1^* =_{R'} \sigma_2^*, \text{ which combined in turn with } \sigma_1^* =_{\text{mod}(C_1)} \sigma_2^* \text{ gives } \sigma_1^* =_R \sigma_2^*.$$

Now we can apply the IH on $\langle C_2, \sigma_1^* \rangle \rightarrow \sigma_1'$ since we know $\sigma_1^* =_{\text{dep}(C_2)} \sigma_2^*$. This gives $\sigma_1' =_{\text{mod}(C_2)} \sigma_2'$, which following a similar argument as before, can be strengthened to $\sigma_1' =_{\text{mod}(C_1) \cup \text{mod}(C_2)} \sigma_2'$.

(*) was derived using rule (rep-t). We know that $c = \text{repeat } c' \text{ until } b$.
 $\langle c, \tau_1 \rangle \rightarrow \tau_1'$, $\langle b, \tau_1' \rangle \rightarrow \text{true}$, $\text{dep}(c) = \text{dep}(c') \cup \text{fv}(b)$ and $\text{mod}(c) = \text{mod}(c')$.
 Let τ_2^* be such that $\langle c', \tau_2 \rangle \rightarrow \tau_2^*$ (τ_2^* exists on view of one of the aux-
 iliary lemmas). We first observe that $\langle b, \tau_2^* \rangle \rightarrow \text{true}$. This follows from
 Lemma 2.6 applied to Boolean expressions and the fact that $\tau_1 = \tau_2^*_{\text{fv}(b)}$. As a
 consequence we have $\tau_2' = \tau_2^*$. We can now apply the IH on $\langle c', \tau_1 \rangle \rightarrow \tau_1'$
 to conclude $\tau_1' = \tau_2'_{\text{mod}(c')}$.

(*) was derived using rule (rep-f). We know that $c = \text{repeat } c' \text{ until } b$.
 $\text{dep}(c) = \text{dep}(c') \cup \text{fv}(b)$, $\text{mod}(c) = \text{mod}(c')$ and there exists state τ_1^* such that
 $\langle c', \tau_1 \rangle \rightarrow \tau_1^*$ $\langle b, \tau_1^* \rangle \rightarrow \text{false}$ $\langle \text{repeat } c' \text{ until } b, \tau_1^* \rangle \rightarrow \tau_1'$
 $\langle \text{repeat } c' \text{ until } b, \tau_1 \rangle \rightarrow \tau_1'$

Let τ_2^* be such that $\langle c', \tau_2 \rangle \rightarrow \tau_2^*$ (τ_2^* exists on account of one of the
 auxiliary lemmas). Now we observe that $\tau_1^* = \tau_2^*_{\text{dep}(c)}$, which follows from
 showing $(\#_1)$ $\tau_1^* =_{R'} \tau_2^*$ and $(\#_2)$ $\tau_1^* =_{\text{mod}(c)} \tau_2^*$, where $R' = \text{dep}(c) \setminus \text{mod}(c)$. The
 proof of $(\#_1)$ follows from the fact that $\tau_1 =_{R'} \tau_2$ (by hypothesis),
 $\tau_1 =_{R'} \tau_1^*$ and $\tau_2 =_{R'} \tau_2^*$ (by aux. lemma). The proof of $(\#_2)$ follow
 from applying the IH on $\langle c', \tau_1 \rangle \rightarrow \tau_1^*$.

Now since $\tau_1^* = \tau_2^*_{\text{dep}(c)}$ we can use Lemma 2.6 to conclude that $\langle b, \tau_2^* \rangle \rightarrow \text{false}$
 and therefore τ_2^* must satisfy $\langle \text{repeat } c' \text{ until } b, \tau_2^* \rangle \rightarrow \tau_2'$. We can now
 apply the IH on $\langle \text{repeat } c' \text{ until } b, \tau_1^* \rangle \rightarrow \tau_2'$ to conclude $\tau_1' = \tau_2'_{\text{mod}(c)}$.