191220154 AsOpSem

1. Let c_1 be $x := \mathbf{0}$, c_2 be $\textbf{\textit{while true do}}\ x := x + \mathbf{1}$, for any σ there is

$$(c_1; c_2, \sigma) = (x := \mathbf{0}; \textbf{\textit{while true do}}\ x := x + \mathbf{1}, \sigma)$$
$$\rightarrow\ (\textbf{\textit{skip}}; \textbf{\textit{while true do}}\ x := x + \mathbf{1}, \sigma\{x \rightsquigarrow 0\})$$
$$\rightarrow\ (\textbf{\textit{while true do}}\ x := x + \mathbf{1}, \sigma\{x \rightsquigarrow 0\})$$
$$\rightarrow\ (\textbf{\textit{if true then}}\ (x := x + \mathbf{1}; \textbf{\textit{while true do}}\ x := x + \mathbf{1})\ \textbf{\textit{else skip}}, \sigma\{x \rightsquigarrow 0\})$$
$$\rightarrow\ (x := x + \mathbf{1}; \textbf{\textit{while true do}}\ x := x + \mathbf{1}, \sigma\{x \rightsquigarrow 0\})$$
$$\rightarrow^*\ (\textbf{\textit{skip}}; \textbf{\textit{while true do}}\ x := x + \mathbf{1}, \sigma\{x \rightsquigarrow 1\})$$
$$\rightarrow\ (\textbf{\textit{while true do}}\ x := x + \mathbf{1}, \sigma\{x \rightsquigarrow 1\})$$

There exists $\sigma' = \sigma\{x \rightsquigarrow 1\}$ so that $(c_1; c_2, \sigma)\ \rightarrow^*\ (c_2, \sigma')$. However,

$$(c_1, \sigma) = (x := \mathbf{0}, \sigma)\ \rightarrow\ (\textbf{\textit{skip}}, \sigma\{x \rightsquigarrow 0\})$$

$(\textbf{\textit{skip}}, \sigma\{x \rightsquigarrow 0\})\ \nrightarrow^*\ (\textbf{\textit{skip}}, \sigma\{x \rightsquigarrow 1\})$, thus $(c_1, \sigma)\ \nrightarrow^*\ (\textbf{\textit{skip}}, \sigma')$

2. (a) the small-step operational semantics rule for ++x

$$\frac{\sigma(x) = \lfloor \boldsymbol{n} \rfloor \qquad \lfloor \boldsymbol{n}' \rfloor = \lfloor \boldsymbol{n} \rfloor + 1}{(\texttt{++}x, \sigma) \rightarrow (\boldsymbol{n}', \sigma\{x \rightsquigarrow \lfloor \boldsymbol{n} \rfloor + 1\})}$$

(b) the full execution path for the program

$$(\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x), \{x \rightsquigarrow 6\})$$
$$\rightarrow\ (\textbf{\textit{if}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ (x := (x++) + (++x);$$
$$\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x))\ \textbf{\textit{else skip}}, \{x \rightsquigarrow 6\})$$
$$\rightarrow\ (\textbf{\textit{if}}\ \mathbf{6} < \mathbf{7}\ \textbf{\textit{do}}\ (x := (x++) + (++x);$$
$$\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x))\ \textbf{\textit{else skip}}, \{x \rightsquigarrow 6\})$$
$$\rightarrow\ (\textbf{\textit{if true do}}\ (x := (x++) + (++x);$$
$$\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x))\ \textbf{\textit{else skip}}, \{x \rightsquigarrow 6\})$$
$$\rightarrow\ (x := (x++) + (++x); \textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x), \{x \rightsquigarrow 6\})$$
$$\rightarrow\ (x := \mathbf{6} + (++x); \textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x), \{x \rightsquigarrow 7\})$$
$$\rightarrow\ (x := \mathbf{6} + \mathbf{8}; \textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x), \{x \rightsquigarrow 8\})$$
$$\rightarrow\ (x := \mathbf{14}; \textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x), \{x \rightsquigarrow 8\})$$
$$\rightarrow\ (\textbf{\textit{skip}}; \textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x), \{x \rightsquigarrow 14\})$$
$$\rightarrow\ (\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x), \{x \rightsquigarrow 14\})$$
$$\rightarrow\ (\textbf{\textit{if}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ (x := (x++) + (++x);$$
$$\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x))\ \textbf{\textit{else skip}}, \{x \rightsquigarrow 14\})$$
$$\rightarrow\ (\textbf{\textit{if}}\ \mathbf{14} < \mathbf{7}\ \textbf{\textit{do}}\ (x := (x++) + (++x);$$

$$\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x))\ \textbf{\textit{else skip}}, \{x \rightsquigarrow 14\})$$

$$\rightarrow (\textbf{\textit{if false do}}\ (x := (x++) + (++x);$$

$$\textbf{\textit{while}}\ x < \mathbf{7}\ \textbf{\textit{do}}\ x := (x++) + (++x))\ \textbf{\textit{else skip}}, \{x \rightsquigarrow 14\})$$

$$\rightarrow (\textbf{\textit{skip}}, \{x \rightsquigarrow 14\})$$

3. (a) small-step operational semantics rules for the new expression

$$\frac{(c,\sigma) \rightarrow (c',\sigma')}{(\textbf{\textit{do}}\ c\ \textbf{\textit{return}}\ e,\sigma) \rightarrow (\textbf{\textit{do}}\ c'\ \textbf{\textit{return}}\ e,\sigma')}$$

$$\frac{}{(\textbf{\textit{do skip return}}\ e,\sigma)\rightarrow(e,\sigma)}$$

(b) For each of the following properties, does it hold?

i. $\forall e_1, e_2.\ (e_1 + e_2) \prec (e_1 + e_2 + \mathbf{1})$

Yes.

ii. $\forall e_1, e_2.\ (e_1 + e_2) \prec (e_2 + e_1 + \mathbf{1})$

No. let e_1 be x, e_2 be $\textbf{\textit{do}}\ x := x - \mathbf{2}\ \textbf{\textit{return}}\ y$

for any given σ, let $\boldsymbol{n_1'} = \sigma(x)$, $\boldsymbol{n_2'} = \sigma(y)$, $\lfloor \boldsymbol{n_3'} \rfloor\ =\ \lfloor \boldsymbol{n_1'} \rfloor\ -\ 2$

$(e_1 + e_2, \sigma)$

$\rightarrow\ (\boldsymbol{n_1'} + e_2, \sigma)$

$\rightarrow\ (\boldsymbol{n_1'} + \boldsymbol{n_2'}, \sigma\{x \rightsquigarrow n_3'\})$

$(e_2 + e_1 + \mathbf{1}, \sigma)$

$\rightarrow\ (\boldsymbol{n_2'} + e_1 + \mathbf{1}, \sigma\{x \rightsquigarrow n_3'\})$

$\rightarrow\ (\boldsymbol{n_2'} + \boldsymbol{n_3'} + \mathbf{1}, \sigma\{x \rightsquigarrow n_3'\})$

so we have $(e_1 + e_2, \sigma)\ \rightarrow^*\ (\boldsymbol{n1}, \sigma_1)$, $(e_2 + e_1 + 1, \sigma)\ \rightarrow^*\ (\boldsymbol{n2}, \sigma_2)$

where $\lfloor \boldsymbol{n_1} \rfloor\ =\ \lfloor \boldsymbol{n_1'} \rfloor + \lfloor \boldsymbol{n_2'} \rfloor$, $\lfloor \boldsymbol{n_2} \rfloor\ =\ \lfloor \boldsymbol{n_2'} \rfloor + \lfloor \boldsymbol{n_3'} \rfloor + 1 =\ \lfloor \boldsymbol{n_1'} \rfloor + \lfloor \boldsymbol{n_2'} \rfloor - 1$

$\lfloor \boldsymbol{n_1} \rfloor > \lfloor \boldsymbol{n_2} \rfloor$, thus $(e_1 + e_2) \prec (e_2 + e_1 + \mathbf{1})$ does not stand

iii. $\forall e.\ \neg(e \prec e)$

No. let e be $\textbf{\textit{do}}\ (\textbf{\textit{while true do skip}})\ \textbf{\textit{return}}\ \mathbf{0}$

there exists no $\boldsymbol{n_1}$, σ_1 or $\boldsymbol{n_2}$, σ_2 so that $(e, \sigma)\ \rightarrow^*\ (\boldsymbol{n_1}, \sigma_1)$, $(e, \sigma)\ \rightarrow^*\ (\boldsymbol{n_2}, \sigma_2)$

as the program loops forever, thus we can say $(e \prec e)$

iv. $\forall e_1, e_2, e_3.\ (e_1 \prec e_2) \wedge (e_2 \prec e_3) \Rightarrow (e_1 \prec e_3)$

No. let e_1 be $\mathbf{1}$, $e2$ be $\textbf{\textit{do}}\ (\textbf{\textit{while true do skip}})\ \textbf{\textit{return}}\ \mathbf{0}$, $e3$ be $\mathbf{0}$

there exists no $\boldsymbol{n_2}$, σ_2 so that $(e_2, \sigma)\ \rightarrow^*\ (\boldsymbol{n_2}, \sigma_2)$

thus we can say $(e_1 \prec e_2)$ and $(e_2 \prec e_3)$

however, because $(e_1, \sigma)\ \rightarrow^*\ (\mathbf{1}, \sigma)$ and $(e_3, \sigma)\ \rightarrow^*\ (\mathbf{0}, \sigma)$

we have $\lfloor \mathbf{1} \rfloor > \lfloor \mathbf{0} \rfloor$, thus $(e_1 \prec e_3)$ does not stand