Exercise session 3
– Natural semantics –

Exercise 11. Termination and looping

For each of the following statements determine whether or not it always terminates and whether or not it always loops. Try to argue for your answers using the axioms and rules of natural semantics of **IMP**.

1. `while not (x=1) do y:=y*x; x:=x-1 end`
2. `while 1≤x do y:=y*x; x:=x-1 end`
3. `while true do skip end`

Exercise 12. Factorial

Consider the factorial statement

```
y:=1; while not (x=1) do y:=y*x; x:=x-1 end
```

and let σ be a state with $\sigma(\mathtt{x})$=3.
Show that

$$\langle \mathtt{y:=1;\ while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma\rangle \mapsto \sigma[\mathtt{y} \mapsto 6][\mathtt{x} \mapsto 1] \quad (*)$$

Exercise 13. Semantical equivalence / inequivalence

Prove that the two statements $\mathtt{S}_1\mathtt{;(S}_2\mathtt{;S}_3\mathtt{)}$ and $\mathtt{(S}_1\mathtt{;S}_2\mathtt{);S}_3$ are semantically equivalent. What does this result imply for compiler construction?
Construct a statement showing that $\mathtt{S}_1\mathtt{;S}_2$ is not, in general, semantically equivalent to $\mathtt{S}_2\mathtt{;S}_1$.

Exercise 14. Extension with `repeat`

Extend the language **IMP** with the statement `repeat` S `until` b and define the relation $\rightarrow$ for it.
The semantics of the `repeat`-construct is not allowed to rely on the existence of a `while`-construct in the language.

Prove that
`repeat` s `until` b

and

s; `if` b `then skip else repeat` s `until` b `end`

are semantically equivalent.

Solutions

Exercise 11. Termination and looping

In **IMP** the execution of statement s on a state σ *loops* iff the `while` axiom is not applicable when trying to construct the derivation tree with root $\langle s, \sigma\rangle \rightarrow \sigma'$.
Statement s *always loops* iff the fact that the `while` axiom is not applicable when constructing the derivation tree is independent of the choice of σ.

1. Does not always terminate and does not always loop since the applicability of the `while` axiom depends on the initial state:
   - if initially $\mathtt{x} \geq 1$ then the axiom will be applicable and the statement terminates,
   - otherwise the `while` axiom will not be applicable (only the rule) and the statement loops.
2. Always terminates as the `while` axiom is eventually applicable independently of the initial state (it only determines *when* the axiom becomes applicable).
3. Always loops as the `while` axiom is never applicable.

Exercise 12. Factorial

In a bottom-up manner we construct a derivation tree T for (*) starting from root

$$\langle \mathtt{y:=1;\ while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma\rangle \mapsto \sigma_{61}$$

The statement in the configuration on the left-hand side is a composition of two statements, thus the last step in constructing T must have used the composition rule:

$$\frac{\langle \mathtt{y:=1}, \sigma\rangle \rightarrow \sigma_{13}, \qquad \langle \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{13}\rangle \mapsto \sigma_{61}\ (**)}{\langle \mathtt{y:=1;\ while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma\rangle \mapsto \sigma_{61}}$$

for some state σ_{13}.
Since $\langle \mathtt{y:=1}, \sigma\rangle \rightarrow \sigma_{13}$ is an instance of the assignment axiom, we get $\sigma_{13} = \sigma[\mathtt{y} \mapsto 1]$.

The statement of (**) must have been constructed by applying either the `while` rule or the `while` axiom. Since $\mathcal{B}[\![\mathtt{not(x = 1)}]\!]\sigma_{13} = tt$ we see that only the `while` rule could have been applied. Thus the derivation tree of (**) has the form:

$$\frac{\langle \mathtt{y:=y*x;x:=x-1}, \sigma_{13}\rangle \rightarrow \sigma_{32}, \qquad \langle \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{32}\rangle \mapsto \sigma_{61}}{\langle \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{13}\rangle \mapsto \sigma_{61}} \quad (***)$$

for some state σ_{32}.

The premise on the left-hand side of (***) contains sequential composition in the statement, thus we know it was constructed as:

$$\frac{\langle \texttt{y:=y*x}, \sigma_{13}\rangle \to \sigma_{33}, \qquad \langle \texttt{x:=x-1}, \sigma_{33}\rangle \mapsto \sigma_{32}}{\langle \texttt{y:=y*x;x:=x-1}, \sigma_{13}\rangle \mapsto \sigma_{32}}$$

where, after applying the assignment axiom twice, we get $\sigma_{33} = \sigma[\texttt{y} \mapsto 3]$ and $\sigma_{32} = \sigma[\texttt{y} \mapsto 3][\texttt{x} \mapsto 2]$.
In a similar way as we proceeded with (**) we can construct the derivation tree for the premise on the right-hand side of (***):

$$\frac{\dfrac{\langle \texttt{y:=y*x}, \sigma_{32}\rangle \to \sigma_{62}, \quad \langle \texttt{x:=x-1}, \sigma_{62}\rangle \mapsto \sigma_{61}}{\langle \texttt{y:=y*x;x:=x-1}, \sigma_{13}\rangle \to \sigma_{32}}, \quad \langle \texttt{while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{61}\rangle \mapsto \sigma_{61}}{\langle \texttt{while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{32}\rangle \mapsto \sigma_{61}}$$

where $\sigma_{62} = \sigma[\texttt{y} \mapsto 6][\texttt{x} \mapsto 2]$ and $\sigma_{61} = \sigma[\texttt{y} \mapsto 6][\texttt{x} \mapsto 1]$.
(Hint: first we applied the `while` rule and then the compositional rule on the premise to the left.)

Finally we see that the premise on the right-hand side is an instance of the `while` axiom since $\mathcal{B}[\![\texttt{not}(\texttt{x} = 1)]\!]\sigma_{61} = \mathit{ff}$. This completes the construction of the derivation tree T.

Exercise 13. Semantical equivalence / inequivalence

We have to show that for all σ, σ'

$$\langle S_1; (S_2; S_3), \sigma\rangle \to \sigma' \;\Leftrightarrow\; \langle (S_1; S_2); S_3), \sigma\rangle \to \sigma'$$

holds.

1. Direction $\Longrightarrow$: we know that there is a derivation tree for $\langle S_1; (S_2; S_3), \sigma\rangle \to \sigma'$ and have to show that there exists one for $\langle (S_1; S_2); S_3), \sigma\rangle \to \sigma'$.

The only derivation tree for $S_1; (S_2; S_3)$ is

$$\frac{\langle S_1, \sigma\rangle \to \sigma'', \quad \dfrac{\langle S_2, \sigma''\rangle \to \sigma''', \quad \langle S_3, \sigma'''\rangle \to \sigma'}{\langle S_2; S_3, \sigma''\rangle \to \sigma'}}{\langle S_1; (S_2; S_3), \sigma\rangle \to \sigma'}$$

Thus, we know that transitions $\langle S_1, \sigma\rangle \to \sigma''$, $\langle S_2, \sigma''\rangle \to \sigma'''$ and $\langle S_3, \sigma'''\rangle \to \sigma'$ hold. Putting them together in a different way, we can get the following derivation tree:

$$\frac{\dfrac{\langle S_1, \sigma\rangle \to \sigma'', \quad \langle S_2, \sigma''\rangle \to \sigma'''}{\langle S_1; S_2, \sigma\rangle \to \sigma'''}, \quad \langle S_3, \sigma'''\rangle \to \sigma'}{\langle (S_1; S_2); S_3, \sigma\rangle \to \sigma'}$$

2. Direction $\Longleftarrow$: Analogous.

To show that $S_1; S_2$ and $S_2; S_1$ are in general not equivalent, consider statements `x:=1` and `x:=2`. Depending on the order we compose them, starting from σ the final states are $\sigma[x \mapsto 1][x \mapsto 2] = \sigma[x \mapsto 2]$ and $\sigma[x \mapsto 2][x \mapsto 1] = \sigma[x \mapsto 1]$ which are not identical states.

Exercise 14. Extension with `repeat`

For the `repeat` construct we need two rules

$$\frac{\langle s, \sigma \rangle \to \sigma'}{\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma \rangle \to \sigma'}\ \mathcal{B}[\![b]\!]\sigma' = tt$$

$$\frac{\langle s, \sigma \rangle \to \sigma', \quad \langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma' \rangle \to \sigma''}{\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma \rangle \to \sigma''}\ \mathcal{B}[\![b]\!]\sigma' = ff$$

The equivalence proof is as follows:

1. Direction $\Longrightarrow$: we assume there is a derivation tree T for

$$\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma \rangle \to \sigma'$$

and have to show that there exists one for

$$\langle s\texttt{;if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma \rangle \to \sigma'.$$

We make a case split on the value of $\mathcal{B}[\![b]\!]$ in the state we get after executing s once in state σ.

- $\mathcal{B}[\![b]\!]\sigma' = tt$
  The last step in the construction of T was to use the first `repeat` rule. Thus, we know that $\langle s, \sigma \rangle \to \sigma'$ holds. Furthermore, we know that for all states σ' transition $\langle \texttt{skip}, \sigma' \rangle \to \sigma'$ holds. Using these two transitions and condition $\mathcal{B}[\![b]\!]\sigma' = tt$ we can construct derivation tree:

$$\frac{\langle s, \sigma \rangle \to \sigma', \quad \dfrac{\langle \texttt{skip}, \sigma' \rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma' \rangle \to \sigma'}\ \mathcal{B}[\![b]\!]\sigma' = tt}{\langle s; \texttt{if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma \rangle \to \sigma'}$$

- $\mathcal{B}[\![b]\!]\sigma'' = ff$
  The last step in the construction of T was to use the second `repeat` rule. Thus, we know that $\langle s, \sigma \rangle \to \sigma''$ and $\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma'' \rangle \to \sigma'$ hold. Using these two transitions and condition $\mathcal{B}[\![b]\!]\sigma'' = ff$ we can construct derivation tree:

$$\frac{\langle s, \sigma \rangle \to \sigma'', \quad \dfrac{\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma'' \rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma'' \rangle \to \sigma'}\ \mathcal{B}[\![b]\!]\sigma'' = ff}{\langle s; \texttt{if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma \rangle \to \sigma'}$$

2. Direction $\Longleftarrow$: we assume there is a derivation tree T for

$$\langle s\texttt{;if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma \rangle \to \sigma''$$

and have to show that there exists one for

$$\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma\rangle \rightarrow \sigma''.$$

The last step in the construction of T was to use the composition rule:

$$\frac{\langle s,\sigma\rangle \rightarrow \sigma', \quad \langle \texttt{if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma'\rangle \rightarrow \sigma''}{\langle s; \texttt{if}\ b\ \texttt{then skip else repeat}\ s\ \texttt{until}\ b\ \texttt{end}, \sigma\rangle \rightarrow \sigma''} \quad (*)$$

We make a case split on the value of $\mathcal{B}[\![b]\!]\sigma'$.

- $\mathcal{B}[\![b]\!]\sigma' = tt$
  Using the left-hand side premise of (*), we can use the first `repeat` rule to construct derivation tree T_1:

$$\frac{\langle s,\sigma\rangle \rightarrow \sigma'}{\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma\rangle \rightarrow \sigma'} \quad \mathcal{B}[\![b]\!]\sigma' = tt$$

  Since $\mathcal{B}[\![b]\!]\sigma' = tt$, from the right-hand side premise of (*) we can deduce $\langle \texttt{skip}, \sigma'\rangle \rightarrow \sigma''$, thus we know that $\sigma' = \sigma''$. Using this result and the root of T_1 we get $\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma\rangle \rightarrow \sigma''$.

- $\mathcal{B}[\![b]\!]\sigma' = f\!f$
  From the right-hand side premise of (*) we can deduce $\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma'\rangle \rightarrow \sigma''$. Using this result and the left-hand side premise of (*) we can use the second `repeat` rule to construct derivation tree:

$$\frac{\langle s,\sigma\rangle \rightarrow \sigma', \quad \langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma'\rangle \rightarrow \sigma''}{\langle \texttt{repeat}\ s\ \texttt{until}\ b, \sigma\rangle \rightarrow \sigma''} \quad \mathcal{B}[\![b]\!]\sigma' = f\!f$$