Exercise session 4
– Structural operational semantics –

Exercise 15. Factorial

Assume that σ(x)=3. Using SOS determine the final state of configuration

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

Exercise 16. Non-interference of statements

Prove that

if $\langle S_1, \sigma \rangle \rightarrow_1^k \sigma'$ then $\langle S_1; S_2, \sigma \rangle \rightarrow_1^k \langle S_2, \sigma' \rangle$

that is, the execution of S_1 is not influenced by the statement following it.

Exercise 17. Properties of SOS

Show that the structural operational semantics is deterministic. Deduce that there is exactly one derivation sequence starting in a configuration $\langle s, \sigma \rangle$. Argue that a statement s of **IMP** cannot both terminate and loop on a state σ and hence it cannot both be always terminating and looping.

Exercise 18. Equivalence of statements

Show that the following statements are semantically equivalent:

- `S;skip` and `S`
- `while` b `do S end` and
  `if` b `then S;while` b `do S end else skip end`
- $\texttt{S}_1\texttt{;(S}_2\texttt{;S}_3\texttt{)}$ and $\texttt{(S}_1\texttt{;S}_2\texttt{);S}_3$

Exercise 19. Holds or not?

Suppose that $\langle S_1; S_2, \sigma \rangle \rightarrow_1^* \langle S_2, \sigma' \rangle$. Is it the case that $\langle S_1, \sigma \rangle \rightarrow_1^* \sigma'$? Either prove that it is the case or give a counter-example to show that it is not the case.

Solutions

Exercise 15. Factorial

In order to determine the final state (if there is one), we have to give the derivation sequence starting from the given configuration.
Let's introduce notation σ_{ab} for state $\sigma[\mathtt{y} \mapsto a][\mathtt{x} \mapsto b]$.
The derivation sequence is as follows:

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

$\langle \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{13}\rangle \rightarrow_1$

$\langle \mathtt{if\ not\ (x=1)\ then\ y:=y*x;\ x:=x-1;}$
$\mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}$
$\mathtt{else\ skip\ end}, \sigma_{13}\rangle \rightarrow_1$

$\langle (\mathtt{y:=y*x;\ x:=x-1});\ \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{13}\rangle \rightarrow_{\mathbf{1}}$

$\langle \mathtt{x:=x-1;\ while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{33}\rangle \rightarrow_{\mathbf{1}}$

$\langle \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{32}\rangle \rightarrow_1$

$\langle \mathtt{if\ not\ (x=1)\ then\ y:=y*x;\ x:=x-1;}$
$\mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}$
$\mathtt{else\ skip\ end}, \sigma_{32}\rangle \rightarrow_1$

$\langle (\mathtt{y:=y*x;\ x:=x-1});\ \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{32}\rangle \rightarrow_{\mathbf{1}}$

$\langle \mathtt{x:=x-1;\ while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{62}\rangle \rightarrow_{\mathbf{1}}$

$\langle \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{61}\rangle \rightarrow_1$

$\langle \mathtt{if\ not\ (x=1)\ then\ y:=y*x;\ x:=x-1;}$
$\mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}$
$\mathtt{else\ skip\ end}, \sigma_{61}\rangle \rightarrow_1$

$\langle \mathtt{skip}, \sigma_{61}\rangle \rightarrow_1 \sigma_{61}$

Thus, the final state is $\sigma[\mathtt{y} \mapsto 6][\mathtt{x} \mapsto 1]$.
However, we are not ready yet, since we have to verify that the transitions above are valid. We have to give the derivation trees for each step. Fortunately, most of the steps were taken by using axioms, for which there are no derivation trees. The 5 transitions for which we have to give derivation trees are marked with bold arrow ($\rightarrow_{\mathbf{1}}$) and they are the following, respectively.

1.

$$\frac{\langle \mathtt{y:=1}, \sigma\rangle \rightarrow_1 \sigma_{13}}{\langle \mathtt{y} := 1;\ \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma\rangle \rightarrow_1 \langle \mathtt{while\ not\ (x=1)\ do\ y:=y*x;\ x:=x-1\ end}, \sigma_{13}\rangle}$$

2.

$$\dfrac{\dfrac{\langle \texttt{y:=y*x}, \sigma_{13}\rangle \to_1 \sigma_{33}}{\langle \texttt{y:=y*x; x:=x-1}, \sigma_{13}\rangle \to_1 \langle \texttt{x:=x-1}, \sigma_{33}\rangle}}{\langle \texttt{(y:=y*x; x:=x-1); while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{13}\rangle \to_1 \langle \texttt{x:=x-1; while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{33}\rangle}$$

3.

$$\dfrac{\langle \texttt{x:=x-1}, \sigma_{33}\rangle \to_1 \sigma_{32}}{\langle \texttt{x:=x-1; while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{33}\rangle \to_1 \langle \texttt{while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{32}\rangle}$$

4.

$$\dfrac{\dfrac{\langle \texttt{y:=y*x}, \sigma_{32}\rangle \to_1 \sigma_{62}}{\langle \texttt{y:=y*x; x:=x-1}, \sigma_{32}\rangle \to_1 \langle \texttt{x:=x-1}, \sigma_{62}\rangle}}{\langle \texttt{(y:=y*x; x:=x-1); while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{32}\rangle \to_1 \langle \texttt{x:=x-1; while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{62}\rangle}$$

5.

$$\dfrac{\langle \texttt{x:=x-1}, \sigma_{62}\rangle \to_1 \sigma_{61}}{\langle \texttt{x:=x-1; while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{62}\rangle \to_1 \langle \texttt{while not (x=1) do y:=y*x; x:=x-1 end}, \sigma_{61}\rangle}$$

Note that the 2nd and 4th tree could have been constructed in one step as the brackets around the inner statement of the loop are not necessary when unfolding the loop.

Exercise 16. Non-interference of statements

We do the proof by induction on the length of the derivation sequence.

Base case: $k = 0$, the property holds as $\langle S_1, \sigma\rangle \to_1^0 \sigma'$ is not a valid transition.

Induction step: we assume that the property holds for $k \leq m$ and prove it for $m + 1$. Thus, we assume $\langle S_1, \sigma\rangle \to_1^{m+1} \sigma'$, which can be written as $\langle S_1, \sigma\rangle \to_1 \gamma \to_1^m \sigma'$ for some intermediate configuration γ. Now we have to make a case distinction depending on whether S_1 was executed in one or in multiple steps.

1. γ was obtained by executing S_1 in one step by transition $\langle S_1, \sigma\rangle \to_1 \sigma'$. Using this transition we can construct a derivation tree for transition $\langle S_1; S_2, \sigma\rangle \to_1 \langle S_2, \sigma'\rangle$. (In this case $m = 0$ and $\gamma = \sigma'$.)

2. γ was obtained by completing the first step of the execution of S_1. In this case we get derivation sequence $\langle S_1, \sigma\rangle \to_1 \langle S_1', \sigma''\rangle \to_1^m \sigma'$ for some statement S_1' and state σ''. Using the induction hypothesis on $\langle S_1', \sigma''\rangle \to_1^m \sigma'$ we get $\langle S_1'; S_2, \sigma''\rangle \to_1^m \langle S_2, \sigma'\rangle$. Using these results we can construct derivation sequence

$$\langle S_1; S_2, \sigma\rangle \to_1 \langle S_1'; S_2, \sigma''\rangle \to_1^m \langle S_2, \sigma'\rangle.$$

The validity of the first step is given by the following derivation tree:

$$\frac{\langle S_1, \sigma\rangle \rightarrow_1 \langle S_1', \sigma''\rangle}{\langle S_1;\ S_2, \sigma\rangle \rightarrow_1 \langle S_1';\ S_2, \sigma''\rangle}$$

Exercise 17. Properties of SOS

We have to prove that

$$\forall\ s, \sigma, \gamma, \gamma'.\ \langle s, \sigma\rangle \rightarrow_1 \gamma \wedge \langle s, \sigma\rangle \rightarrow_1 \gamma' \Longrightarrow \gamma = \gamma'$$

We prove determinism by induction on the shape of the derivation tree for $\langle s, \sigma\rangle \rightarrow_1 \gamma$.

We have the following base cases:

Assignment. To get γ we have to use the assignment axiom which gives transition $\langle \texttt{x:=}e, \sigma\rangle \rightarrow_1 \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]$. Thus, $\gamma = \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]$. The only way to get γ' is to use the assignment axiom on $\langle \texttt{x:=}e, \sigma\rangle$, which yields $\gamma' = \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]$. We can see that $\gamma = \gamma'$.

Cases for **Skip** and **While** are analogous.

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

- $\mathcal{B}[\![b]\!]\sigma = tt$. In this case we can only apply the `if` axiom with the corresponding condition and get transition $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \sigma\rangle \rightarrow_1 \langle s_1, \sigma\rangle$. Thus, $\gamma = \langle s_1, \sigma\rangle$. The only way to get γ' is to apply the corresponding `if` axiom which leads to $\gamma' = \langle s_1, \sigma\rangle$. Thus, $\gamma = \gamma'$.
- $\mathcal{B}[\![b]\!]\sigma = ff$. Analogous.

The induction step is the case of **Composition:** $\langle s_1; s_2, \sigma\rangle \rightarrow_1 \gamma$.

We have to make a case split on whether s_1 is executed in one step or not.

- s_1 can be executed in one step.
  Let's assume that transition $\langle s_1, \sigma\rangle \rightarrow_1 \sigma'$ holds. Then transition $\langle s_1; s_2, \sigma\rangle \rightarrow_1 \langle s_2, \sigma'\rangle$ is valid which yields $\gamma = \langle s_2, \sigma'\rangle$.
  Let's assume we can get $\gamma' = \langle s_2, \sigma''\rangle$ by using transition $\langle s_1, \sigma\rangle \rightarrow_1 \sigma''$. Applying the induction hypothesis to $\langle s_1, \sigma\rangle \rightarrow_1 \sigma'$ we learn that it is deterministic, thus $\sigma' = \sigma''$. This also yields $\gamma = \gamma'$.
- s_1 is executed in multiple steps.
  Let's assume that transition $\langle s_1, \sigma\rangle \rightarrow_1 \langle s_1', \sigma'\rangle$ holds. Then transition $\langle s_1; s_2, \sigma\rangle \rightarrow_1 \langle s_1'; s_2, \sigma'\rangle$ is valid which yields $\gamma = \langle s_1'; s_2, \sigma'\rangle$.
  Let's assume we can get $\gamma' = \langle s_1''; s_2, \sigma''\rangle$ by using transition $\langle s_1, \sigma\rangle \rightarrow_1 \langle s_1'', \sigma''\rangle$. Applying the induction hypothesis to $\langle s_1, \sigma\rangle \rightarrow_1 \langle s_1', \sigma'\rangle$ we learn that it is deterministic, thus $\sigma' = \sigma''$ and $s_1' = s_1''$. This also yields $\gamma = \gamma'$.

We have shown that one step in SOS is deterministic. Since a derivation sequence is a chain of such steps and all of these steps are deterministic, there can only be a unique sequence starting from a given configuration $\langle s, \sigma \rangle$.

Since the derivation sequence of statement s on state σ is unique, it cannot both terminate and loop as the two cases would require different derivation sequences. We have seen that there exists a state σ for which s cannot both terminate and loop, thus s cannot both terminate and loop for all states. This means s cannot both always terminate and always loop.

Exercise 18. Equivalence of statements

Statement s_1 and s_2 are semantically equivalent if for all states:

1. $\langle s_1, \sigma \rangle \rightarrow_1^* \gamma \Leftrightarrow \langle s_2, \sigma \rangle \rightarrow_1^* \gamma$
   where γ is either a stuck configuration or a terminal state.
2. both s_1 and s_2 loop.

Since in the (basic) **IMP** language we cannot get a stuck configuration we only have to look at termination and looping.

- `S; skip` $\equiv$ `S`.

  Direction $\Longrightarrow$

  Case a: `S; skip` terminates, that is, there exists a derivation sequence $\langle \texttt{S; skip}, \sigma \rangle \rightarrow_1^* \sigma'$. Using Lemma 2.19 (p.37 in book, p.116 on slides) we get that transition $\langle \texttt{S}, \sigma \rangle \rightarrow_1^* \sigma''$ and $\langle \texttt{skip}, \sigma'' \rangle \rightarrow_1^* \sigma'$ hold for some state σ''. The second transition also gives $\sigma' = \sigma''$. Using transition $\langle \texttt{S}, \sigma \rangle \rightarrow_1^* \sigma''$ and the fact that $\sigma' = \sigma''$ we can get $\langle \texttt{S}, \sigma \rangle \rightarrow_1^* \sigma'$.

  Case b: `S; skip` loops. Since statement `skip` always terminates it can only be `S` that loops.

  Direction $\Longleftarrow$

  Case a: `S` terminates, that is, there exists a derivation sequence $\langle \texttt{S}, \sigma \rangle \rightarrow_1^* \sigma'$. Using Exercise 16, we get $\langle \texttt{S; skip}, \sigma \rangle \rightarrow_1^* \langle \texttt{skip}, \sigma' \rangle$. Now using the `skip` axiom we can get final state σ'. This gives us the required sequence.

  Case b: `S` loops, thus its sequential composition will loop too.

- `while` b `do S end` $\equiv$ `if` b `then S; while` b `do S end else skip end`.

  Direction $\Longrightarrow$

  Case a: `while` b `do S end` terminates, that is, there exists a derivation sequence $\langle \texttt{while}\, b\, \texttt{do S end}, \sigma \rangle \rightarrow_1 \gamma \rightarrow_1^* \sigma'$. The first step of the sequence

could only be $\langle \texttt{while}\ b\ \texttt{do S end}, \sigma\rangle \rightarrow_1 \langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle$. Thus, there is also a derivation sequence

$$\langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle \rightarrow_1^* \sigma'.$$

Case b: the statement loops, that is, we have an infinite derivation sequence $\langle \texttt{while}\ b\ \texttt{do S end}, \sigma\rangle \rightarrow_1 \gamma \rightarrow_1^* \ldots$
The first step could again only be

$\langle \texttt{while}\ b\ \texttt{do S end}, \sigma\rangle \rightarrow_1 \langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle$

thus we can get the infinite sequence

$$\langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle \rightarrow_1^* \ldots$$

Direction $\Longleftarrow$

Case a: the statement terminates, that is, there exists a derivation sequence $\langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle \rightarrow_1^* \sigma'$. Using the `while` axiom, transition

$\langle \texttt{while}\ b\ \texttt{do S end}\rangle \rightarrow_1 \langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle$

holds. Using this transition and the assumption, we can compose a derivation sequence $\langle \texttt{while}\ b\ \texttt{do S end}\rangle \rightarrow_1^* \sigma'$.

Case b: the statement loops, that is, we have an infinite derivation sequence $\langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle \rightarrow_1 \gamma \rightarrow_1^* \ldots$
This can only occur if $\mathcal{B}[\![b]\!]\sigma = tt$, then using the `while` axiom we get transition $\langle \texttt{if}\ b\ \texttt{then S;}\ \texttt{while}\ b\ \texttt{do S end else skip end}, \sigma\rangle \rightarrow_1 \langle \texttt{while}\ b\ \texttt{do S end}, \sigma\rangle$. Thus, we can get the infinite sequence $\langle \texttt{while}\ b\ \texttt{do S end}, \sigma\rangle \rightarrow_1^* \ldots$

- $\texttt{S}_1\texttt{;(S}_2\texttt{;S}_3\texttt{)} \equiv \texttt{(S}_1\texttt{;S}_2\texttt{);S}_3$

  Direction $\Longrightarrow$

  Case a: all three sub-statements terminate.
  Using Lemma 2.19 we can get derivation sequences $\langle \texttt{S}_1, \sigma\rangle \rightarrow_1^* \sigma''$ and $\langle \texttt{S}_2\texttt{;S}_3, \sigma''\rangle \rightarrow_1^* \sigma'$ for some state σ''. We can apply the lemma again on the second sequence and get $\langle \texttt{S}_2, \sigma''\rangle \rightarrow_1^* \sigma'''$ and $\langle \texttt{S}_3, \sigma'''\rangle \rightarrow_1^* \sigma'$ for some state σ'''. Using the result of Exercise 16 on sequence $\langle \texttt{S}_1, \sigma\rangle \rightarrow_1^* \sigma''$ we get $\langle \texttt{S}_1\texttt{;S}_2, \sigma\rangle \rightarrow_1^* \langle \texttt{S}_2, \sigma''\rangle$. This, combined with sequence $\langle \texttt{S}_2, \sigma''\rangle \rightarrow_1^* \sigma'''$ yields $\langle \texttt{S}_1\texttt{;S}_2, \sigma\rangle \rightarrow_1^* \sigma'''$. Using the result of Exercise 16 on this sequence gives $\langle \texttt{(S}_1\texttt{;S}_2\texttt{);S}_3, \sigma\rangle \rightarrow_1^* \langle \texttt{S}_3, \sigma'''\rangle$. This, combined with sequence $\langle \texttt{S}_3, \sigma'''\rangle \rightarrow_1^* \sigma'$ yields the required sequence $\langle \texttt{(S}_1\texttt{;S}_2\texttt{);S}_3, \sigma\rangle \rightarrow_1^* \sigma'$.

  Case b: any of the three sub-statements loops.
  Since we have a looping statement in sequential composition with other statements, the whole composition will loop. Thus, both $\texttt{S}_1\texttt{;(S}_2\texttt{;S}_3\texttt{)}$ and $\texttt{(S}_1\texttt{;S}_2\texttt{);S}_3$ will loop.

  Direction $\Longleftarrow$ Analogous.

Exercise 19. Holds or not?

A counter-example is the following statement `skip; while true do x:=x+1 end` because we can construct the derivation sequence

$\langle$`skip; while true do x:=x+1 end`$, \sigma\rangle \rightarrow_1$
$\langle$`while true do x:=x+1 end`$, \sigma\rangle \rightarrow_1^3$
$\langle$`while true do x:=x+1 end`$, \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) + 1]\rangle$

Thus,

$\langle$`skip; while true do x:=x+1 end`$, \sigma\rangle \rightarrow_1^* \langle$`while true do x:=x+1 end`$, \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x})+1]\rangle$

holds, but $\langle$`skip`$, \sigma\rangle \rightarrow_1^* \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) + 1]$ is not a valid sequence.