Exercise session 8
– Fixed point theory (cont'd) –

Exercise 30. Fixed point iteration

Consider the statement

$\texttt{z:=0;while y} \leq \texttt{x do z:=z+1;x:=x} - \texttt{y end}$

First determine the fixed point of the functional of the loop, then determine the final state if the initial state is $\{\texttt{x} \mapsto 3, \texttt{y} \mapsto 1\}$.

Exercise 31.

Let $f : D \to D$ be a continuous function on ccpo $(D, \sqsubseteq)$ and let $d \in D$ satisfy $f(d) \sqsubseteq d$. Prove that $FIX\ f \sqsubseteq d$.

Exercise 32.

Show that $\mathcal{S}_{DS}[\![\texttt{while true do skip end}]\!]$ is the totally undefined function $\perp$.

Exercise 33. Semantical equivalence

Show that the following statements are semantically equivalent:

- `S; skip` and `S`
- $S_1;\ (S_2;\ S_3)$ and $(S_1;\ S_2);\ S_3$
- $\texttt{while } b \texttt{ do } S \texttt{ end}$ and
  $\texttt{if } b \texttt{ then } S\texttt{;while } b \texttt{ do } S \texttt{ end else skip end}$

Solutions

Exercise 30. Fixed point iteration

We start by determining the fixed point of the functional of the loop.
We know that $\mathcal{S}_{DS}[\![\texttt{while y} \leq \texttt{x do z:=z+1;x:=x} - \texttt{y end}]\!] = FIX\ F$, where

$$\begin{aligned}
F(g)\sigma &= cond(\mathcal{B}[\![\texttt{y} \leq \texttt{x}]\!], g \circ \mathcal{S}_{DS}[\![\texttt{z:=z+1;x:=x} - \texttt{y}]\!], id)\sigma \\
&= cond(\mathcal{B}[\![\texttt{y} \leq \texttt{x}]\!], g \circ \mathcal{S}_{DS}[\![\texttt{x:=x} - \texttt{y}]\!] \circ \mathcal{S}_{DS}[\![\texttt{z:=z+1}]\!], id)\sigma \\
&= \begin{cases} (g \circ \mathcal{S}_{DS}[\![\texttt{x:=x} - \texttt{y}]\!] \circ \mathcal{S}_{DS}[\![\texttt{z:=z+1}]\!])\sigma & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases} \\
&= \begin{cases} g(\sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})]) & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases}
\end{aligned}$$

Now we can start the fixed point iteration:

$$\begin{aligned}
F(\perp)\sigma &= \begin{cases} \perp(\sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})]) & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases} \\
&= \begin{cases} \text{undefined} & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases}
\end{aligned}$$

$$\begin{aligned}
F^2(\perp)\sigma &= F(F(\perp))\sigma \\
&= \begin{cases} F(\perp)(\sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})]) & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases} \\
&= \begin{cases} \text{undefined} & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ \sigma(\texttt{y}) \leq \sigma(\texttt{x}) - \sigma(\texttt{y}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})] & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ \sigma(\texttt{y}) > \sigma(\texttt{x}) - \sigma(\texttt{y}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases} \\
&= \begin{cases} \text{undefined} & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ 2 * \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})] & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) < 2 * \sigma(\texttt{y}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases}
\end{aligned}$$

$$\begin{aligned}
F^3(\perp)\sigma &= F(F^2(\perp))\sigma \\
&= \begin{cases} F^2(\perp)(\sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})]) & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases} \\
&= \begin{cases} \text{undefined} & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ \sigma(\texttt{y}) \leq \sigma(\texttt{x}) - \sigma(\texttt{y}) \ \wedge \ 2 * \sigma(\texttt{y}) \leq \sigma(\texttt{x}) - \sigma(\texttt{y}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1 + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y}) - \sigma(\texttt{y})] & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ \sigma(\texttt{y}) \leq \sigma(\texttt{x}) - \sigma(\texttt{y}) < 2 * \sigma(\texttt{y}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})] & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ \sigma(\texttt{y}) > \sigma(\texttt{x}) - \sigma(\texttt{y}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases} \\
&= \begin{cases} \text{undefined} & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ 2 * \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ 3 * \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 2][\texttt{x} \mapsto \sigma(\texttt{x}) - 2 * \sigma(\texttt{y})] & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ 2 * \sigma(\texttt{y}) \leq \sigma(\texttt{x}) < 3 * \sigma(\texttt{y}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})] & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ 2 * \sigma(\texttt{y}) > \sigma(\texttt{x}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases} \\
&= \begin{cases} \text{undefined} & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \ \wedge \ 3 * \sigma(\texttt{y}) \leq \sigma(\texttt{x}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 2][\texttt{x} \mapsto \sigma(\texttt{x}) - 2 * \sigma(\texttt{y})] & \text{if } 2 * \sigma(\texttt{y}) \leq \sigma(\texttt{x}) < 3 * \sigma(\texttt{y}) \\ \sigma[\texttt{z} \mapsto \sigma(\texttt{z}) + 1][\texttt{x} \mapsto \sigma(\texttt{x}) - \sigma(\texttt{y})] & \text{if } \sigma(\texttt{y}) \leq \sigma(\texttt{x}) < 2 * \sigma(\texttt{y}) \\ \sigma & \text{if } \sigma(\texttt{y}) > \sigma(\texttt{x}) \end{cases}
\end{aligned}$$

Now we can see the pattern how the iteration works, thus we can give $F^n(\bot)\sigma$:

$$F^n(\bot)\sigma \quad = \begin{cases} \text{undefined} & \text{if } \sigma(\mathtt{y}) \leq \sigma(\mathtt{x}) \ \wedge \ n * \sigma(\mathtt{y}) \leq \sigma(\mathtt{x}) \\ \sigma[\mathtt{z} \mapsto \sigma(\mathtt{z}) + (j-1)][\mathtt{x} \mapsto \sigma(\mathtt{x}) - (j-1) * \sigma(\mathtt{y})] & \text{if } (j-1) * \sigma(\mathtt{y}) \leq \sigma(\mathtt{x}) < j * \sigma(\mathtt{y}) \ \wedge \ 1 < j \leq n \\ \sigma & \text{if } \sigma(\mathtt{y}) > \sigma(\mathtt{x}) \end{cases}$$

Now we can also give the fixed point of F:

$$(FIX\ F)\sigma \quad = \begin{cases} \text{undefined} & \text{if } \sigma(\mathtt{y}) \leq \sigma(\mathtt{x}) \ \wedge \ \sigma(\mathtt{y}) \leq 0 \\ \sigma[\mathtt{z} \mapsto \sigma(\mathtt{z}) + (n-1)][\mathtt{x} \mapsto \sigma(\mathtt{x}) - (n-1) * \sigma(\mathtt{y})] & \text{if } (n-1) * \sigma(\mathtt{y}) \leq \sigma(\mathtt{x}) < n * \sigma(\mathtt{y}) \ \wedge \ 1 < n \\ \sigma & \text{if } \sigma(\mathtt{y}) > \sigma(\mathtt{x}) \end{cases}$$

As we see, the second conjunct of the first condition has been simplified. This is due to the fact that in case $\sigma(\mathtt{y})$ is positive, the left-hand side gets arbitrary big, as n can be arbitrary big. Thus, there is no value $\sigma(\mathtt{x})$ that can be greater-equal to the left-hand side.

Now we can go back to our original task: determining $\mathcal{S}_{DS}[\![\mathtt{z:=0;while\ y} \leq \mathtt{x\ do\ z:=z+1;x:=x} - \mathtt{y\ end}]\!]\sigma_0$, where $\sigma_0 = \{\mathtt{x} \mapsto 3, \mathtt{y} \mapsto 1\}$.

We know that

$$\begin{array}{lcr}
\mathcal{S}_{DS}[\![\mathtt{z:=0;while\ y} \leq \mathtt{x\ do\ z:=z+1;x:=x} - \mathtt{y\ end}]\!]\sigma_0 & = & [\text{sequential composition}] \\
(\mathcal{S}_{DS}[\![\mathtt{while\ y} \leq \mathtt{x\ do\ z:=z+1;x:=x} - \mathtt{y\ end}]\!] \circ \mathcal{S}_{DS}[\![\mathtt{z:=0}]\!])\sigma_0 & = & [S_{DS} \text{ of assignment and loop}] \\
(FIX\ F)\sigma_0[\mathtt{z} \mapsto 0] & = & [\text{applying } FIX\ f \text{ with } n = 4] \\
\sigma_0[\mathtt{z} \mapsto 0][\mathtt{z} \mapsto \sigma_0[\mathtt{z} \mapsto 0](\mathtt{z}) + (4-1)][\mathtt{x} \mapsto \sigma_0[\mathtt{z} \mapsto 0](\mathtt{x}) - (4-1) * \sigma_0[\mathtt{z} \mapsto 0](\mathtt{y})] & = & [\text{simplication}] \\
\sigma_0[\mathtt{z} \mapsto 0 + 3][\mathtt{x} \mapsto 3 - 3 * 1] & = & [\text{state update}] \\
\{\mathtt{x} \mapsto 0, \mathtt{y} \mapsto 1, \mathtt{z} \mapsto 3\} & &
\end{array}$$

Exercise 31.

The important thing to observe is that $f(d) \sqsubseteq d$. Due to monotonicity of f, this means that $f(f(d)) \sqsubseteq f(d)$, $f(f(f(d))) \sqsubseteq f(f(d))$ and so on. Due to transitivity of $\sqsubseteq$, we get $\forall\ d' \in \{f^n(d)|n > 0\} : d' \sqsubseteq d$. This means that d is an upper bound of $\{f^n(d)|n > 0\}$. Since by definition $FIX\ f$ is the *least upper bound* of $\{f^n(d)|n > 0\}$, we get $FIX\ f = \sqcup\{f^n(d)|n > 0\} \sqsubseteq d$ as required. Note that $\{f^n(d)|n > 0\}$ is a chain, thus it has a least upper bound.

Exercise 32.

From the direct style semantics of loops we know that $\mathcal{S}_{DS}[\![\texttt{while true do skip end}]\!] = FIX\ F$, where functional F is defined as

$$F(g) = cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{DS}[\![s]\!], id).$$

In our case this gives

$$F(g) = cond(\mathcal{B}[\![\texttt{true}]\!], g \circ \mathcal{S}_{DS}[\![\texttt{skip}]\!], id).$$

Since the condition is `true` and $\mathcal{S}_{DS}[\![\texttt{skip}]\!] = id$, it can be simplified to $F(g) = g \circ id = g$. Now we have to find the least fixed point of this functional. We do that by fixed point iteration:
$F(\perp) = \perp$, which means that $\perp$ is the least fixed point, thus $\mathcal{S}_{DS}[\![\texttt{while true do skip end}]\!] = \perp$.

Exercise 33. Semantical equivalence

To show semantical equivalence of statements S_1 and S_2, we have to prove that $\mathcal{S}_{DS}[\![S_1]\!] = \mathcal{S}_{DS}[\![S_2]\!]$.

1) `S; skip` and `S`.
$\mathcal{S}_{DS}[\![\texttt{S; skip}]\!] = \mathcal{S}_{DS}[\![\texttt{skip}]\!] \circ \mathcal{S}_{DS}[\![\texttt{S}]\!] = id \circ \mathcal{S}_{DS}[\![\texttt{S}]\!] = \mathcal{S}_{DS}[\![\texttt{S}]\!]$.

2) $S_1;\ (S_2;\ S_3)$ and $(S_1;\ S_2);\ S_3$.
$\mathcal{S}_{DS}[\![S_1;\ (S_2;\ S_3)]\!] = \mathcal{S}_{DS}[\![S_2;\ S_3]\!] \circ \mathcal{S}_{DS}[\![S_1]\!] = \mathcal{S}_{DS}[\![S_3]\!] \circ \mathcal{S}_{DS}[\![S_2]\!] \circ \mathcal{S}_{DS}[\![S_1]\!]$.
$\mathcal{S}_{DS}[\![(S_1;\ S_2);\ S_3]\!] = \mathcal{S}_{DS}[\![S_3]\!] \circ \mathcal{S}_{DS}[\![S_1;\ S_2]\!] = \mathcal{S}_{DS}[\![S_3]\!] \circ \mathcal{S}_{DS}[\![S_2]\!] \circ \mathcal{S}_{DS}[\![S_1]\!]$.

3) `while` b `do` S `end` and `if` b `then` S`;while` b `do` S `end else skip end`.
We know that $\mathcal{S}_{DS}[\![\texttt{while}\ b\ \texttt{do}\ S\ \texttt{end}]\!] = FIX\ F$, where functional F is defined as

$$F(g) = cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{DS}[\![S]\!], id)$$

We can apply the `if`-axiom on the conditional statement and simplify the result as follows:

$$\begin{array}{lcr}
\mathcal{S}_{DS}[\![\texttt{if}\ b\ \texttt{then}\ S\texttt{;while}\ b\ \texttt{do}\ S\ \texttt{end else skip end}]\!] & = & [S_{DS}\ \texttt{of conditional}] \\
cond(\mathcal{B}[\![b]\!], \mathcal{S}_{DS}[\![S;\ \texttt{while}\ b\ \texttt{do}\ S\ \texttt{end}]\!], \mathcal{S}_{DS}[\![\texttt{skip}]\!]) & = & [S_{DS}\ \texttt{of composition and skip}] \\
cond(\mathcal{B}[\![b]\!], \mathcal{S}_{DS}[\![\texttt{while}\ b\ \texttt{do}\ S\ \texttt{end}]\!] \circ \mathcal{S}_{DS}[\![S]\!], id) & = & [S_{DS}\ \texttt{of loop}] \\
cond(\mathcal{B}[\![b]\!], FIX\ F \circ \mathcal{S}_{DS}[\![S]\!], id) & = & [FIX\ F = g] \\
cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{DS}[\![S]\!], id) & &
\end{array}$$

Now we can see that the semantic function gives the same function for the two statements.