# Exercise session 6
# – Fixed points and partial orders –

**Exercise 22.** Finding fixed points

Determine the functional $F$ associated with the statement

$$\texttt{while x}\#0 \texttt{ do x} := \texttt{x} - 1 \texttt{ end}.$$

Consider the following partial functions of State $\hookrightarrow$ State:

$$g_1(\sigma) = \text{undefined for all } \sigma$$

$$g_2(\sigma) = \begin{cases} \sigma[\texttt{x} \mapsto 0] & \text{if } \sigma(\texttt{x}) \geq 0 \\ \text{undefined} & \text{if } \sigma(\texttt{x}) < 0 \end{cases}$$

$$g_3(\sigma) = \begin{cases} \sigma[\texttt{x} \mapsto 0] & \text{if } \sigma(\texttt{x}) \geq 0 \\ \sigma & \text{if } \sigma(\texttt{x}) < 0 \end{cases}$$

$$g_4(\sigma) = \sigma[\texttt{x} \mapsto 0] \text{ for all } \sigma$$

$$g_5(\sigma) = \sigma \text{ for all } \sigma$$

Determine which of these functions are fixed points of $F$.
Determine which of the fixed points is the desired fixed point.

**Exercise 23.** Subtype relation

Consider a (Java-like) programming language with single inheritance where each class inherits from exactly one class. The only exception is class `Object`, which does not inherit from any class. The subtype relation is acyclic.
Tasks:

- Formalize the subtype relation on types.
- Prove that it is a partial order and show that it is not a total order.
- Determine whether the partial order has a least element or not. If yes, give the least element.

Now, consider a (C++ like) language with multiple inheritance where each class inherits from 0, 1 or more classes. The subtype relation is acyclic.
Work out the same tasks as for the previous language.

**Exercise 24.** Partial ordering

Let $g_1, g_2$, and $g_3$ be defined as follows:

$$g_1(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathtt{x}) \text{ is even} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$g_2(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathtt{x}) \text{ is prime} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$g_3(\sigma) = \sigma \text{ for all } \sigma$$

Determine the ordering on the $\sqsubseteq$ relation among these partial functions.
Determine a partial function $g_4$ such that $g_4 \sqsubseteq g_1$, $g_4 \sqsubseteq g_2$, and $g_4 \sqsubseteq g_3$.
Determine a partial function $g_5$ such that $g_1 \sqsubseteq g_5$, $g_2 \sqsubseteq g_5$, and $g_5 \sqsubseteq g_3$, but $g_5$ is neither equal to $g_1, g_2$ nor $g_3$.

## Solutions

**Exercise 22.** Finding fixed points

From the direct style semantics of loops we know that functional $F$ for a loop `while` b `do` s `end` is defined as

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

In our case this gives

$$F(g) = cond(\mathcal{B}[\![\mathtt{x}\#0)]\!], g \circ \mathcal{S}_{DS}[\![\mathtt{x} := \mathtt{x} - 1]\!], id).$$

We know that $\mathcal{S}_{DS}[\![\mathtt{x} := \mathtt{x} - 1]\!]\sigma = \sigma[\mathtt{x} \mapsto \mathcal{A}[\![\mathtt{x} - 1]\!]\sigma] = \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) - 1]$.
Now we can apply function *cond* and get the functional:

$$F(g)\sigma = \begin{cases} g(\sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) - 1]) & \text{if } \sigma(\mathtt{x}) \neq 0 \\ \sigma & \text{if } \sigma(\mathtt{x}) = 0 \end{cases}$$

In order to determine which of the given functions, $g_i$, are fixed points we have to find out whether $F(g_i)\sigma = g_i(\sigma)$ holds or not.

1. $g_1(\sigma) =$ undefined for all $\sigma$.

$$F(g_1)\sigma = \begin{cases} \text{undefined} & \text{if } \sigma(\mathtt{x}) \neq 0 \\ \sigma & \text{if } \sigma(\mathtt{x}) = 0 \end{cases}$$

Since $F(g_1)\sigma \neq g_1(\sigma)$, $g_1$ is not a fixed point of $F$.

2. $g_2\ \sigma = \begin{cases} \sigma[\mathtt{x} \mapsto 0] & \text{if } \sigma(\mathtt{x}) \geq 0 \\ \text{undefined} & \text{if } \sigma(\mathtt{x}) < 0 \end{cases}$

$$\begin{aligned} F(g_2)\sigma &= \begin{cases} \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) - 1][\mathtt{x} \mapsto 0] & \text{if } \sigma(\mathtt{x}) > 0 \\ \text{undefined} & \text{if } \sigma(\mathtt{x}) < 0 \\ \sigma & \text{if } \sigma(\mathtt{x}) = 0 \end{cases} \\ &= \begin{cases} \sigma[\mathtt{x} \mapsto 0] & \text{if } \sigma(\mathtt{x}) \geq 0 \\ \text{undefined} & \text{if } \sigma(\mathtt{x}) < 0 \end{cases} \\ &= g_2(\sigma) \end{aligned}$$

We can see that $F(g_2)\sigma = g_2(\sigma)$, thus $g_2$ is a fixed point of $F$.

3. $g_3(\sigma) = \begin{cases} \sigma[\mathtt{x} \mapsto 0] & \text{if } \sigma(\mathtt{x}) \geq 0 \\ \sigma & \text{if } \sigma(\mathtt{x}) < 0 \end{cases}$

$$\begin{aligned} F(g_3)\sigma &= \begin{cases} \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) - 1][\mathtt{x} \mapsto 0] & \text{if } \sigma(\mathtt{x}) > 0 \\ \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) - 1] & \text{if } \sigma(\mathtt{x}) < 0 \\ \sigma & \text{if } \sigma(\mathtt{x}) = 0 \end{cases} \\ &= \begin{cases} \sigma[\mathtt{x} \mapsto 0] & \text{if } \sigma(\mathtt{x}) \geq 0 \\ \sigma[\mathtt{x} \mapsto \sigma(x) - 1] & \text{if } \sigma(\mathtt{x}) < 0 \end{cases} \end{aligned}$$

We can see that $F(g_3)\sigma \neq g_3\ \sigma$, thus $g_3$ is not a fixed point of $F$.

4. $g_4(\sigma) = \sigma[\mathtt{x} \mapsto 0]$ for all $\sigma$.

$$\begin{aligned} F(g_4)\sigma \quad &= \begin{cases} \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) - 1][\mathtt{x} \mapsto 0] & \text{if } \sigma(\mathtt{x}) \neq 0 \\ \sigma & \text{if } \sigma(\mathtt{x}) = 0 \end{cases} \\ &= \sigma[\mathtt{x} \mapsto 0] \\ &= g_4(\sigma) \end{aligned}$$

We can see that $F(g_4)\sigma = g_4\ \sigma$, thus $g_4$ is a fixed point of $F$.

5. $g_5(\sigma) = \sigma$ for all $\sigma$

$$F(g_5)\sigma = \begin{cases} \sigma[\mathtt{x} \mapsto \sigma(\mathtt{x}) - 1] & \text{if } \sigma(\mathtt{x}) \neq 0 \\ \sigma & \text{if } \sigma(\mathtt{x}) = 0 \end{cases}$$

We can see that $F(g_5)\sigma \neq g_5(\sigma)$, thus $g_5$ is not a fixed point of $F$.

Now we have to decide whether $g_2$ or $g_4$ is the preferred fixed point. We can see that it is $g_2$ because $g_2(\sigma) = \sigma' \Rightarrow g_4(\sigma) = \sigma'$, but not vice versa.

**Exercise 23.** Subtype relation

We start with the Java-like language.
**Formalization.**
Let's use symbol $\prec$ denoting direct subtype relation between two classes. That is, a class declaration `class A extends B` $\{\ldots\}$ leads to relation $A \prec B$. Let's use symbol $\preceq$ for the reflexive transitive closure of $\prec$ (i.e. $A \preceq B$ expresses that $A$ is a subtype of $B$).
We can formalize the subtype relation as a pair $(Classes, \preceq)$, where $Classes$ is the set of classes occuring in a given program.

**Partially ordered set.**
First we prove that $(Classes, \preceq)$ is a partially ordered set.

1. the relation is reflexive as for any class $A$, $A$ is subtype of $A$.
2. the relation is transitive as if $A$ is subtype of $B$ and $B$ is subtype of $C$ then $A$ is subtype of $C$.
3. the relation is anti-symmetric as if $A$ is subtype of $B$ and $B$ is subtype of $A$ then $A$ and $B$ must be the same class (subtype relation is acyclic).

**Not total.**
The relation is not a total order as we can have the following class declarations:

```
class A extends C {...}
class B extends C {...}
```

In this case class $A$ and $B$ are not related to each other.

**No least element.**
A least element ($C$) of our partially ordered set should satisfy

$$\forall C' \in Classes : C \preceq C',$$

that is, $C$ should be subtype of all classes in set $Classes$. From the previous example we can see that neither $A$, $B$ or $C$ is a subtype of all classes $A$, $B$, and $C$. Thus, our partially ordered set does not have a least element.

Now we look at the C++ like language.
The formalization is the same and partiality can be proved as above.
The relation is not a total order as we can have two classes that are not in subtype relation with any other class. Thus, they are not related to each other. For the same reason we do not have a least element.

**Exercise 24.** Partial ordering

Recall that $g \sqsubseteq g'$ means that $g(\sigma) = \sigma' \Rightarrow g'(\sigma) = \sigma'$ for all $\sigma, \sigma'$.
Since $g(\sigma) = \sigma' \Rightarrow g(\sigma) = \sigma'$ holds for all functions $g$ and states $\sigma, \sigma'$, relation $\sqsubseteq$ is reflexive. Thus, $g_1 \sqsubseteq g_1, g_2 \sqsubseteq g_2$, and $g_3 \sqsubseteq g_3$.
Furthermore,

- $g_1 \sqsubseteq g_3$ because in a state $\sigma$ where the value of `x` is an even number both functions give $\sigma$ and for all other states $g_1$ gives undefined (using the definition we can see that this makes the left-hand side of the implication false, thus the implication yields true).
  But $g_3 \not\sqsubseteq g_1$ as $g_3$ is a total function while $g_1$ is partial, thus the implication cannot yield true for all states (e.g. for state $\sigma$ where the value of `x` is 3).

- $g_2 \sqsubseteq g_3$ because in a state $\sigma$ where the value of `x` is a prime number both functions give $\sigma$ and for all other states $g_2$ gives undefined.
  But $g_3 \not\sqsubseteq g_2$ as $g_3$ is a total function while $g_2$ is partial, thus the implication cannot yield true for all states (e.g. for state $\sigma$ where the value of `x` is 4).

We can show that $g_1$ and $g_2$ are not related by two examples:

- For a state $\sigma$ where the value of `x` is 4 $g_1$ gives $\sigma$ while $g_2$ gives undefined. Thus, $g_1 \not\sqsubseteq g_2$.

- For a state $\sigma$ where the value of `x` is 3 $g_2$ gives $\sigma$ while $g_1$ gives undefined. Thus, $g_2 \not\sqsubseteq g_1$.

The simplest partial function $g_4$ that fulfils the requirements is $g_4(\sigma) =$ undefined (the implication holds vacuously for all states).
Another partial function (that is not undefined on all states) could be

$$g_4(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathtt{x}) = 2 \\ \text{undefined} & \text{otherwise} \end{cases}$$

For a state $\sigma$ where the value of `x` is 2 $g_4, g_1, g_2$, and $g_3$ gives $\sigma$ and in all other states $g_4$ is undefined.

A partial function $g_5$ that fulfils the requirements is

$$g_5(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathtt{x}) \text{ is even or prime} \\ \text{undefined} & \text{otherwise} \end{cases}$$

We can see that

- $g_1 \sqsubseteq g_5$ as both functions give $\sigma$ for a state $\sigma$ where the value of `x` is even.
- $g_2 \sqsubseteq g_5$ as both functions give $\sigma$ for a state $\sigma$ where the value of `x` is prime.
- $g_5 \sqsubseteq g_3$ as both functions give $\sigma$ for a state $\sigma$ where the value of `x` is even or prime.