

Exercise Sheet 5: Denotational Semantics

Due date: May 20th. You can hand in your solutions at the start of the exercise class.

Exercise 1 (Denotational Semantics of a Recursive Function)

75%

Consider the following recursive program for $n \in \mathbb{Z}$:

$$\text{fac}(n) := \text{if } (n = 0) \text{ then } \{1\} \text{ else } \{\text{fac}(n - 1) * n\};$$

- (a) [5%] Determine the functional $\Phi : (\mathbb{Z} \dashrightarrow \mathbb{Z}) \rightarrow (\mathbb{Z} \dashrightarrow \mathbb{Z})$ for $\text{fac}(n)$, as in the lecture.

Solution:

$$\Phi(f)(x) = \begin{cases} 1 & \text{if } x = 0 \\ x * f(x - 1) & \text{otherwise} \end{cases}$$

- (b) [20%] Show that Φ is monotonic and continuous.

Solution:

Monotonicity: To show: for any $f, g : \mathbb{Z} \dashrightarrow \mathbb{Z}$ such that $f \sqsubseteq g$, it also holds that $\Phi(f) \sqsubseteq \Phi(g)$.

Proof: Let $f \sqsubseteq g$. To show: $\Phi(f) \sqsubseteq \Phi(g)$, i.e. $\Phi(f)(x) = x'$ implies $\Phi(g)(x) = x'$. We distinguish the cases $x = 0$ and $x \neq 0$.

$x = 0$:

$$\begin{aligned} \Phi(f)(x) &= x' \\ &= \{\text{def. } \Phi, \text{ using } x = 0\} \\ &= 1 = x' \\ &= \{\text{def. } \Phi, \text{ using } x = 0\} \\ \Phi(g)(x) &= x' \end{aligned}$$

$x \neq 0$:

$$\begin{aligned} \Phi(f)(x) &= x' \\ &= \{\text{def. } \Phi, \text{ using } x \neq 0\} \\ &= x * f(x - 1) = x' \\ &\Rightarrow \{f \sqsubseteq g\} \\ &= x * g(x - 1) = x' \\ &= \{\text{def. } \Phi, \text{ using } x \neq 0\} \\ \Phi(g)(x) &= x' \end{aligned}$$

Continuity: Let F be a chain. To show: $\sqcup\Phi(F) = \Phi(\sqcup F)$. By monotonicity, $\sqcup\Phi(F) \sqsubseteq \Phi(\sqcup F)$ holds. It remains to show that $\Phi(\sqcup F) \sqsubseteq \sqcup\Phi(F)$, i.e. $\Phi(\sqcup F)(x) = x'$ implies $\sqcup\Phi(F) = x'$. We distinguish the cases $x = 0$ and $x \neq 0$.

$x = 0$:

$$\begin{aligned}\Phi(\sqcup F)(x) &= x' \\ &= 1 = x' \\ &= \sqcup\Phi(F)(x) = x'\end{aligned}$$

$x \neq 0$:

$$\begin{aligned}\Phi(\sqcup F)(x) &= x' \\ &= \{\text{def. } \Phi, \text{ using } x \neq 0\} \\ &\quad x * (\sqcup F)(x - 1) = x' \\ &= \{\text{def. } \sqcup F\} \\ &\quad x * (z \mapsto f(z) \text{ for some } f \in F \text{ for which } f(z) \text{ defined})(x - 1) = x' \\ &= (z \mapsto x * f(z) \text{ for some } f \in F \text{ for which } f(z) \text{ is defined})(x - 1) = x' \\ &= (z \mapsto z * f(z - 1) \text{ for some } f \in F \text{ for which } f(z) \text{ is defined})(x) = x' \\ &= (z \mapsto \Phi(f)(z) \text{ for some } f \in F \text{ for which } f(z) \text{ is defined})(x) = x' \\ &= \sqcup\{\Phi(f) \mid f \in F\}(x) = x' \\ &= \sqcup\Phi(F)(x) = x'\end{aligned}$$

(c) [10%] Show that the partial order $(\mathbb{Z} \dashrightarrow \mathbb{Z}, \sqsubseteq)$ is chain complete.

Solution: To show: $(\mathbb{Z} \dashrightarrow \mathbb{Z}, \sqsubseteq)$ is chain complete. Proof: we show that every chain has a LUB by posing a definition for this LUB and show it is correct.

$$\text{Let } (\sqcup S)(x) := \begin{cases} f(x) & \text{if } f(x) \text{ defined, for } f \in S \\ \text{undefined} & \text{otherwise} \end{cases}.$$

This is a well-defined function for all chains, as all functions in a chain are comparable and thus, if they are defined for a particular x , the resulting values are equal.

$(\sqcup S)$ is an upper bound: for all $f \in S$, $f \sqsubseteq (\sqcup S)$, i.e. for all $x \in \mathbb{Z}$, if $f(x) = x'$, then $(\sqcup S)(x) = x'$. Trivial, by definition of $(\sqcup S)$.

$(\sqcup S)$ is the *least* upper bound: let F be an upper bound of S , i.e. $f \sqsubseteq F$ for all $f \in S$. We show that $(\sqcup S) \sqsubseteq F$, i.e. $(\sqcup S)(x) = m$ implies $F(x) = m$.

$$\begin{aligned}(\sqcup S)(x) &= m \\ \Leftrightarrow & \\ f(x) &= m \text{ for some } f \in S \\ \Rightarrow \{F \text{ is upper bound for chain } S\} & \\ F(x) &= m\end{aligned}$$

- (d) [20%] Let $\mathfrak{C}[\text{fac}(n)]$ be defined by $\text{fix}(\Phi)$. Compute the denotational semantics of $\text{fac}(3)$.

Solution: We approximate the fixpoint of Φ by iterating Φ starting from f_0 , the bottom element $f_0 := z \mapsto \text{undefined}$. This forms a chain (by monotonicity of Φ), the supremum of which is the least fixpoint $\text{fix}(\Phi)$ (by Knaster-Tarski). By chain-completeness of $(\mathbb{Z} \rightarrow \mathbb{Z}, \sqsubseteq)$, this fixpoint is well defined.

$$\begin{aligned}
 f_0(z) &= \text{undefined} \\
 f_0(z) &= \Phi(f_0)(z) \\
 &= \begin{cases} 1 & \text{if } z = 0 \\ z * f_0(z - 1) & \text{otherwise} \end{cases} \\
 &= \begin{cases} 1 & \text{if } z = 0 \\ z * \text{undefined} & \text{otherwise} \end{cases} \\
 &= \begin{cases} 1 & \text{if } z = 0 \\ \text{undefined} & \text{otherwise} \end{cases} \\
 f_1(z) &= \Phi(f_0)(z)
 \end{aligned}$$

$$\begin{aligned}
 &= \begin{cases} 1 & \text{if } z = 0 \\ z * f_0(z - 1) & \text{otherwise} \end{cases} \\
 &= \begin{cases} 1 & \text{if } z = 0 \\ z * 1(z - 1) & \text{if } z - 1 = 0 \\ \text{undefined} & \text{otherwise} \end{cases} \\
 &= \begin{cases} 1 & \text{if } z = 0 \\ z * 1 & \text{if } z = 1 \\ \text{undefined} & \text{otherwise} \end{cases} \\
 &= \begin{cases} 1 & \text{if } z = 0 \\ 1 & \text{if } z = 1 \\ \text{undefined} & \text{otherwise} \end{cases} \\
 f_2(z) &= \Phi(f_1)(z)
 \end{aligned}$$

$$\begin{aligned}
&= \begin{cases} 1 & \text{if } z = 0 \\ z * f_1(z - 1) & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } z = 0 \\ z * 1 & \text{if } z - 1 = 0 \\ z * 1 & \text{if } z - 1 = 1 \\ \text{undefined} & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } z = 0 \\ z * 1 & \text{if } z = 1 \\ z * 1 & \text{if } z = 2 \\ \text{undefined} & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } z = 0 \\ 1 & \text{if } z = 1 \\ 2 & \text{if } z = 2 \\ \text{undefined} & \text{otherwise} \end{cases}
\end{aligned}$$

$$f_3(z) = \Phi(f_2)(z)$$

$$\begin{aligned}
&= \begin{cases} 1 & \text{if } z = 0 \\ z * f_2(z - 1) & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } z = 0 \\ z * 1 & \text{if } z - 1 = 0 \\ z * 1 & \text{if } z - 1 = 1 \\ z * 2 & \text{if } z - 1 = 2 \\ \text{undefined} & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } z = 0 \\ z * 1 & \text{if } z = 1 \\ z * 1 & \text{if } z = 2 \\ z * 2 & \text{if } z = 3 \\ \text{undefined} & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } z = 0 \\ 1 & \text{if } z = 1 \\ 2 & \text{if } z = 2 \\ 6 & \text{if } z = 3 \\ \text{undefined} & \text{otherwise} \end{cases}
\end{aligned}$$

So $f_3(3) = 6$. Because $f_3 \sqsubseteq \text{fix}(\Phi)$ (by definition of $\text{fix}(\Phi)$ as supremum), we conclude that $\text{fix}(\Phi)(3) = 6$.

- (e) [20%] Prove that the program `fac` calculates the factorial, i.e. $\text{fix}(\Phi)(n) = n!$ for any $n \in \mathbb{Z}$.

Solution: To show: $\text{fix}(\Phi)(n) = n!$ for $n \in \mathbb{Z}$.

Proof: we show $\text{fix}(\Phi)(n) \stackrel{(1)}{=} f_n(n) \stackrel{(2)}{=} n!$ for all $n \in \mathbb{Z}$, where $f_n = f_\emptyset$ for $n < 0$ and $f_n = \Phi(f_{n-1})$ for $n \geq 0$. By Knaster-Tarski, the least fixpoint of Φ is the supremum of the chain $f_\emptyset, f_0, f_1, \dots$.

- (1) Case $n < 0$

$n!$ is not defined for negative n .

It remains to show that $\text{fix}(\Phi)(n)$ is not defined for negative n either. We show that $f_k(n)$ is not defined for any $k \in \mathbb{N}$. Because $\text{fix}(\Phi)$ is the supremum (the *least* upper bound) of all these functions f_k , $\text{fix}(\Phi)(n)$ is not defined either.

Claim: $f_k(n)$ is not defined for any $k \in \mathbb{N}$. The case $f_\emptyset$ is trivial. Proof by mathematical induction for $k \in \mathbb{N}$.

- Base: $0 \in \mathbb{N}$.

To show: $f_0(n)$ is not defined for any $n < 0$

Proof: let $n < 0$.

$$\begin{aligned} f_0(n) &= \Phi(f_\emptyset)(n) \\ &= \begin{cases} 1 & \text{if } n = 0 \\ n * f_\emptyset(n-1) & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } n \neq 0 \\ \text{undefined} & \text{otherwise} \end{cases} \\ &= \text{undefined} \end{aligned}$$

- Induction step: show claim for k , assuming claim holds for $k' < k$.

Induction Hypothesis: $f_{k'}(n')$ is not defined for any $k' < k$, $n' < 0$.

To show: $f_k(n)$ is not defined for any $n < 0$. Proof: let $n < 0$.

$$\begin{aligned} f_k(n) &= \Phi(f_{k-1})(n) \\ &= \begin{cases} 1 & \text{if } n = 0 \\ n * f_{k-1}(n-1) & \text{otherwise} \end{cases} \\ &= \{n < 0\} \\ &\quad n * f_{k-1}(n-1) \\ &= \{k-1 < k, n < 0 \Rightarrow n-1 < 0: \text{IH applies, with } k' = k-1, n' = n-1\} \\ &\quad n * \text{undefined} \\ &= \text{undefined} \end{aligned}$$

- (2) Case $n < 0$

$$\begin{aligned} f_n(n) &= f_{\emptyset}(n) \\ &= \text{undefined}, \end{aligned}$$

and

$$n! = \text{undefined}.$$

- (1) Case $n \geq 0$

By definition of $\text{fix}(\Phi)$ as the supremum of the chain $f_{\emptyset}, f_0, f_1, \dots$, it holds that $f_n \sqsubseteq \text{fix}(\Phi)$, i.e. $f_n(x) = x' \Rightarrow \text{fix}(\Phi)(x) = x'$. By (2), Case $n \geq 0$, it holds that $f_n(n) = n!$, so we conclude $\text{fix}(\Phi)(n) = n!$.

- (2) Case $n \geq 0$

By mathematical induction on n . Claim: $f_n(n) = n!$, for all $n \in \mathbb{N}$.

- Base: $0 \in \mathbb{N}$.
To show: $f_0(0) = 0!$.
Proof: $f_0(0) = 1 = 0!$
- Induction Step: $n \in \mathbb{N} \Rightarrow n + 1 \in \mathbb{N}$.
Induction Hypothesis: $f_n(n) = n!$.
To show: $f_{n+1}(n + 1) = (n + 1)!$.
Proof:

$$\begin{aligned} f_{n+1}(n + 1) &= \Phi(f_n)(n + 1) \\ &= (n + 1) * f_n((n + 1) - 1) \\ &= (n + 1) * f_n(n) \\ &= \{\text{IH}\} \\ &\quad (n + 1) * n! \\ &= (n + 1)! \end{aligned}$$

Exercise 2 (Denotational Semantics of Guarded Commands)

25%

Dijkstra's *guarded commands* are essentially of the form

$$\text{do}\{b_1 \rightarrow c_1 \quad b_2 \rightarrow c_2\}$$

(where $b_1, b_2 \in \text{BExp}$ and $c_1, c_2 \in \text{Cmd}$). They form a natural generalisation of the **WHILE** loop:

While at least one of the tests is true, the corresponding statement is executed.

Here the satisfaction of both tests results in a non-deterministic choice of the command. The computation terminates as soon as neither of the tests is true.

- (a) [10%] Which function on the natural numbers is computed by the following state-

ment? Transform it to a *WHILE* statement.

```
do{
  x > y → x := x - y
  y > x → y := y - x
}
```

Solution: greatest common divisor: i.e. $\sigma_{12,9} \rightarrow \sigma_{3,9} \rightarrow \sigma_{3,6} \rightarrow \sigma_{3,3}$

Termination when $x = y$.

while $(\neg(x = y))$ **do** {if $(x > y)$ then $\{x := x - y\}$ else $\{y := y - x\}$ }

- (b) [15%] Let $b_1, b_2 \in \mathbf{BExp}$ be two mutually excluding tests (i.e., in no state both b_1 and b_2 are true) and $c_1, c_2 \in \mathbf{Cmd}$. How can the semantics of

```
do{
  b1 → c1
  b2 → c2
}
```

be defined as the least fixpoint of a mapping

$$\Phi : (\Sigma \dashrightarrow \Sigma) \rightarrow (\Sigma \dashrightarrow \Sigma)?$$

Solution: In general: **while** $(b_1 \text{ xor } b_2)$ **do** {if (b_1) then $\{c_1\}$ else $\{c_2\}$ }

Thus $\mathfrak{C}[[c]] = \text{fix}(\Phi)$ where

$$\begin{aligned} \Phi(f) &= \text{cond}(\mathfrak{B}[[b_1 \text{ xor } b_2]], f \circ \mathfrak{C}[[\text{if } (b_1) \text{ then } \{c_1\} \text{ else } \{c_2\}]], \text{id}_\Sigma) \\ &= \text{cond}(\mathfrak{B}[[b_1 \text{ xor } b_2]], f \circ \text{cond}(\mathfrak{B}[[b_1]], \mathfrak{C}[[c_1]], \mathfrak{C}[[c_2]]), \text{id}_\Sigma) \end{aligned}$$

$$\text{For state } \sigma \in \Sigma: \Phi(f)(\sigma) = \begin{cases} f(\mathfrak{C}[[c_1]]\sigma) & \text{if } \mathfrak{B}[[b_1]](\sigma) = \text{true and } \mathfrak{B}[[b_2]](\sigma) = \text{false} \\ f(\mathfrak{C}[[c_2]]\sigma) & \text{if } \mathfrak{B}[[b_2]](\sigma) = \text{true and } \mathfrak{B}[[b_1]](\sigma) = \text{false} \\ \sigma & \text{otherwise} \end{cases}$$