Prof. Dr. J. Giesl
M. Hark

Notes:

- Please solve these exercises in **groups of four**!
- The solutions must be handed in **directly before (very latest: at the beginning of)** the exercise course on Wednesday, 29.05.2019, 14:30, in lecture hall **AH I**. Alternatively you can drop your solutions into a box which is located right next to Prof. Giesl's office (until 30 minutes before the exercise course starts).
- Please write the **names** and **immatriculation numbers** of all students on your solution. Also please staple the individual sheets!

In all exercises, if not explicitly stated otherwise, you can use results from previous parts of the exercises, even if you could not solve them.

Exercise 1 (Fixpoints): (4 points)

Consider the function $ee : \langle \mathbb{Z}_\perp \to \mathbb{B}_\perp \rangle \to \langle \mathbb{Z}_\perp \to \mathbb{B}_\perp \rangle$, which is defined as follows:

$$(ee(g))(x) = \begin{cases} \texttt{True} & \text{if } x = 0 \\ \neg g(x-1) & \text{if } x > 0 \\ \neg g(x+1) & \text{if } x < 0 \\ \perp_{\mathbb{B}_\perp} & \text{if } x = \perp_{\mathbb{Z}_\perp} \end{cases}$$

The function ee is continuous. Thus, by Kleene's Fixpoint Theorem it has a least fixpoint. This least fixpoint of ee is a well-known function. What is the least fixpoint of ee? Prove your claim!

Hints:

- We have $\neg \perp_{\mathbb{B}_\perp} = \perp_{\mathbb{B}_\perp}$. Thus, the function $\neg$ is monotonic and also continuous.
- If the cases "$x > 0$" and "$x < 0$" are analogous, then it suffices to prove one case and note the analogy.

Exercise 2 (Continuity and Fixpoints): (2 + 2 + 4 + 3 = 11 points)

In this exercise, we will prove that for any real number $0 < r \leq 1$ we have $\lim_{n\to\infty} f^n(r) = \underbrace{f(f(\ldots f}_{n \text{ times}}(r)\ldots)) = 1$, where $f : [0, \infty) \to \mathbb{R}, x \mapsto \sqrt{x}$. In this exercise you can use that f is topological–continuous (cf. Exercise Sheet 4, Ex. 4), i.e., for any converging sequence $(x_n)_{n\in\mathbb{N}}$ in $[0,\infty)$ we have $\lim_{n\to\infty} f(x_n) = f(\lim_{n\to\infty} x_n)$. In each of the following parts you can use the results of Exercise Sheet 4, Ex. a)-c).

a) Prove that f is monotonic w.r.t. $\leq$, i.e., if $0 \leq x < y$, then $f(x) \leq f(y)$.

Hints:

- You can use that if $0 \leq x < y$ then $x^2 \leq y^2$
- You can also use that for every $0 \leq z$ we have $(\sqrt{z})^2 = z$ and $(\sqrt{z^2}) = z$.

b) Let $0 < r \leq 1$. Prove that $f_r : [r, 1] \to [r, 1], x \mapsto \sqrt{x}$ is well defined, i.e., if $r \leq x \leq 1$, then $r \leq \sqrt{x} \leq 1$

c) Let $0 < r \leq 1$. Use a), b) and Ex. 4 a)-c) from Exercise Sheet 4 to conclude that f_r is Scott–continuous for any $0 < r \leq 1$. You can use that if $(x_n)_{n\in\mathbb{N}}$ is a converging sequence in $[r, 1]$, then $\lim_{n\to\infty} x_n \in [r, 1]$, too.

d) Use Kleene's Fixpoint Theorem and Ex. 4 a) from Exercise Sheet 4 to conclude that for each $0 < r \leq 1$ $\lim_{n\to\infty} f^n(r) = 1$.

Exercise 3 (Fixpoints and Higher Order Functions): (1 + 1 + 2 + 3 = 7 points)

Consider the following Haskell functions:

```
fact   :: Int -> Int
fact   = \x -> if x <= 0 then 1 else fact (x-1) * x

true :: Bool -> Bool
true = \x -> True

neg_inf    :: Int -> Int
neg_inf    = \x -> neg_inf (x-3)

fib :: Int -> Int
fib = \n -> if n <= 1 then 1 else fib (n - 1) + fib (n - 2)
```

The higher-order Haskell function `f_fact` corresponding to `fact` is:

```
f_fact = \g -> \x -> if x <= 0 then 1 else g (x-1) * x
```

The semantics $\phi_{\texttt{f\_fact}}$ of `f_fact` is:

$$(\phi_{\texttt{f\_fact}}(g))(x) = \begin{cases} 1, & \text{if } x \leq 0 \\ g(x-1) \cdot x, & \text{otherwise} \end{cases}$$

The semantics $\phi_{\texttt{fact}}$ of `fact` is the least fixpoint of $\phi_{\texttt{f\_fact}}$ (where for all $x \leq 0$ we define $x! = 1$):

$$\phi_{\texttt{fact}}(x) = \begin{cases} x!, & \text{if } x \in \mathbb{Z} \\ \perp, & \text{otherwise} \end{cases}$$

(a) Give the Haskell definitions for the higher-order functions `f_true`, `f_neg_inf`, and `f_fib` corresponding to `true`, `neg_inf`, and `fib`.

(b) Give the semantics $\phi_{\texttt{f\_true}}$, $\phi_{\texttt{f\_neg\_inf}}$, and $\phi_{\texttt{f\_fib}}$ of the functions `f_true`, `f_neg_inf`, and `f_fib`.

(c) What does the function $\phi_f^n(\perp)$ compute for $n \in \mathbb{N}$, $f \in \{$`f_true`, `f_neg_inf`, `f_fib`$\}$? Here, $\phi_f^n(\perp)$ denotes n applications of ϕ_f to the undefined function $\perp$.

(d) Give all fixpoints of the semantic functions $\phi_{\texttt{f\_true}}$, $\phi_{\texttt{f\_neg\_inf}}$, and $\phi_{\texttt{f\_fib}}$ from (b). Which ones are the least fixpoints?

Exercise 4 (Domain Lifts): (2 + 2 + 4 = 8 points)

Let $D_1, \ldots, D_n$ be domains with complete partial orders $\sqsubseteq_{D_1}, \ldots, \sqsubseteq_{D_n}$.

a) Prove that $\sqsubseteq_{D_1 \oplus \ldots \oplus D_n}$ is a complete partial order on $D_1 \oplus \ldots \oplus D_n$.

b) Prove that for any $1 \leq k \leq n$ the embedding $\iota_k : D_k \to D_1 \oplus \ldots \oplus D_n, x \mapsto \begin{cases} \perp_{D_1 \oplus \ldots \oplus D_n}, & x = \perp_{D_k} \\ x^{D_k}, & \textit{otherwise} \end{cases}$ is continuous.

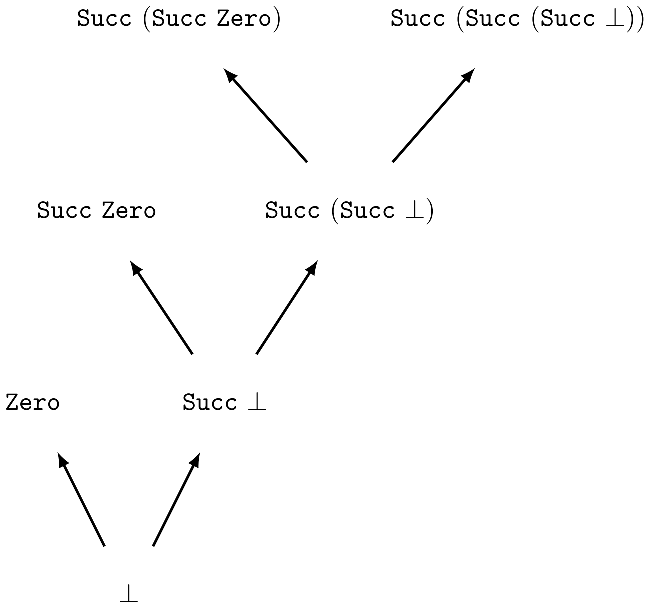
c) Let $f : D_1 \oplus \ldots \oplus D_n \to D$ be a monotonic function, where D is some domain with cpo $\sqsubseteq_D$. Prove that f is continuous if $f \circ \iota_k : D_k \to D$ is continuous for every $1 \leq k \leq n$.

Exercise 5 (Domain Construction): (4 + 1 + 2 = 7 points)

Consider the following data type declaration for natural numbers:

```
data Nats = Zero | Succ Nats
```

A graphical representation of the first four levels of the domain for `Nats` could look like this:



Now consider the following data type declarations:

```
data Unit = U ()
data Foo = A Unit Unit | B Bool
```

a) Give a graphical representation of the whole domain for the type `Foo`. The graphical representation must be a directed graph, as in the example, and must contain all elements of `Foo`.

b) Give Haskell expressions that correspond to the following elements of the domains for the type `Foo` and the type `Unit` respectively, i.e., for each of these elements, give a Haskell expression that has this element as its semantics:

- `A` $\perp$ $\perp$
- `U` $\perp$

c) Consider the following Haskell functions:

```
a :: (Integer, Integer) -> Integer
a (k, n) = if k <= 0 then n+1 else if n <= 0 then a(k-1,1) else a(k-1, a(k, n-1))

u :: Bool -> Unit
u b = if (not b) then U () else u b

f :: Bool -> Bool
f b = f b
```

What is the semantics of the following Haskell expressions?

- `B (f False)`
- `A (u (f False)) (u False)`
- `let g = \n -> a (3, n) < 3 in A (U ()) (u (g 2))`