Universität Siegen
Lehrstuhl Theoretische Informatik
Carl Philipp Reh

Übungsblatt 11

Aufgabe 1. Bestimmen Sie für die folgenden Haskell-Programme jeweils die Semantik. Welche Funktionen müssen bereits in der Umgebung eingetragen sein? Welche Data-Deklarationen muss es im Programm geben?

(a)

```
let a = \x -> x == 42 in a 3
```

Lösung. Zunächst fordern wir, dass wir die übliche Definition von `Bool` haben. Dann fordern wir, dass in jeder Umgebung η gilt, dass

$$\eta((==))\colon \mathrm{Dom} \to \mathrm{Dom} \to \mathrm{Dom},$$
$$\eta((==))(x)(y) = \begin{cases} \underline{\mathrm{True}} & \text{falls } x = y \text{ und } x, y \in \mathbb{Z}, \\ \underline{\mathrm{False}} & \text{falls } x \neq y \text{ und } x, y \in \mathbb{Z}, \\ \bot & \text{sonst.} \end{cases}$$

Als Semantik ergibt sich dann

$$[\![\textbf{let a} = \backslash \mathrm{x} \mathrel{-\!\!>} \mathrm{x} == 42 \textbf{ in a } 3]\!]_\eta = [\![\mathrm{a}\ 3]\!]_{\eta \circledcirc [a/\mu f]},$$

wobei $f\colon \mathrm{Dom} \to \mathrm{Dom}$ definiert ist als

$$f(d) = [\![\backslash \mathrm{x} \mathrel{-\!\!>} \mathrm{x} == 42]\!]_{\eta \circledcirc [a/d]}.$$

Man beachte, dass a hier keine Rolle spielt, weil die Definition nicht rekursiv ist. Deshalb gilt

$$\mu f = [\![\backslash \mathrm{x} \mathrel{-\!\!>} \mathrm{x} == 42]\!]_\eta = f',$$

wobei $f'\colon \mathrm{Dom} \to \mathrm{Dom}$ die Funktion ist mit

$$\begin{aligned} f'(d) &= [\![\mathrm{x} == 42]\!]_{\eta \circledcirc [x/d]} \\ &= [\![(==)]\!]_{\eta \circledcirc [x/d]}([\![\mathrm{x}]\!]_{\eta \circledcirc [x/d]})([\![42]\!]_{\eta \circledcirc [x/d]}) \\ &= \begin{cases} \underline{\mathrm{True}} & \text{falls } d = 42, \\ \underline{\mathrm{False}} & \text{falls } d \neq 42 \text{ und } d \in \mathbb{Z}, \\ \bot & \text{sonst.} \end{cases} \end{aligned}$$

Damit erhalten wir $[\![\mathrm{a}\ 3]\!]_{\eta \circledcirc [a/\mu f]} = f'(3) = \underline{\mathrm{False}}$.

(b)

```
let s = \x -> case x <= 0 of
        { True -> 0
        ; False -> x + s (x - 1) }
in s 10
```

Lösung. Wir benötigen wieder die übliche Definition von `Bool`. Dann fordern wir, dass in jeder Umgebung η gilt, dass

$$\eta((<=)) \colon \mathrm{Dom} \to \mathrm{Dom} \to \mathrm{Dom},$$
$$\eta((<=))(x)(y) = \begin{cases} \underline{\mathrm{True}} & \text{falls } x \leq y \text{ und } x, y \in \mathbb{Z}, \\ \underline{\mathrm{False}} & \text{falls } x > y \text{ und } x, y \in \mathbb{Z}, \\ \bot & \text{sonst.} \end{cases}$$

Außerdem soll für $\eta((-))$ gelten, dass

$$\eta((-)) \colon \mathrm{Dom} \to \mathrm{Dom} \to \mathrm{Dom},$$
$$\eta((-))(x)(y) = \begin{cases} x - y & \text{falls } x, y \in \mathbb{Z}, \\ \bot & \text{sonst.} \end{cases}$$

In der Vorlesung haben wir bereits gefordert, dass $\eta((+))$ die Addition ist. Als Semantik ergibt sich dann

$$\begin{aligned} &[\![\textbf{let } s = \backslash x \texttt{->} \textbf{ case } x \texttt{<=} 0 \textbf{ of} \\ &\qquad \{\underline{\mathrm{True}} \texttt{->} 0 \\ &\qquad \texttt{;}\ \underline{\mathrm{False}} \texttt{->} x + s\ (x-1)\}]\!]_\eta \\ &= [\![\mathrm{s}\ 10]\!]_{\eta \circledcirc [s/\mu f]}, \end{aligned}$$

wobei $f \colon \mathrm{Dom} \to \mathrm{Dom}$ definiert ist als

$$\begin{aligned} f(d) = [\![&\backslash x \texttt{->} \textbf{ case } x \texttt{<=} 0 \textbf{ of} \\ &\{\underline{\mathrm{True}} \texttt{->} 0 \\ &\texttt{;}\ \underline{\mathrm{False}} \texttt{->} x + s\ (x-1)\}]\!]_{\eta \circledcirc [s/d]}. \end{aligned}$$

Es gilt $f(d) = f'_d$, wobei $f'_d \colon \mathrm{Dom} \to \mathrm{Dom}$ definiert ist als

$$\begin{aligned} f'_d(d') = [\![&\textbf{case } x \texttt{<=} 0 \textbf{ of} \\ &\{\underline{\mathrm{True}} \texttt{->} 0 \\ &\texttt{;}\ \underline{\mathrm{False}} \texttt{->} x + s\ (x-1)\}]\!]_{\eta \circledcirc [s/d, x/d']}. \end{aligned}$$

Zunächst stellen wir fest, dass für $d, d' \in \text{Dom}$ gilt, dass

$$[\![\text{x} <= 0]\!]_{\eta @ [s/d, x/d']} = \begin{cases} \underline{\text{True}} & \text{falls } d' \leq 0, \\ \underline{\text{False}} & \text{falls } d' > 0, \\ \bot & \text{sonst.} \end{cases}$$

Dann lässt sich f'_d umformen zu

$$f'_d(d') = \begin{cases} [\![0]\!]_{\eta @ [s/d, x/d']} & \text{falls } d' \leq 0, \\ [\![\text{x} + \text{s } (\text{x} - 1)]\!]_{\eta @ [s/d, x/d']} & \text{falls } d' > 0, \\ \bot & \text{sonst.} \end{cases}$$

Für den zweiten Fall erhalten wir für $d, d' \in \text{Dom}$, dass

$$\begin{aligned} [\![\text{x} + \text{s } (\text{x} - 1)]\!]_{\eta @ [s/d, x/d']} &= [\![(((+)\ \text{x})\ (\text{s}\ (((-)\ \text{x})\ 1)))]\!]_{\eta @ [s/d, x/d']} \\ &= \begin{cases} d' + d(d' - 1) & \text{falls } d, d' \neq \bot, \\ \bot & \text{sonst.} \end{cases} \end{aligned}$$

Damit ergibt sich, dass

$$f'_d(d') = \begin{cases} 0 & \text{falls } d' \leq 0, \\ d' + d(d' - 1) & \text{falls } d' > 0 \text{ und } d \neq \bot, \\ \bot & \text{sonst.} \end{cases}$$

Wir müssen nun bestimmen, was μf ist. Dazu verwenden wir den Fixpunktsatz, also müssen wir $\sqcup \lambda i. f^i(\bot)$ bestimmen. Sei $d_0 \colon \text{Dom} \to \text{Dom}$ mit $d_0(x) = \bot$ und sei für $i \in \mathbb{N}$ die Funktion $d_{i+1} \colon \text{Dom} \to \text{Dom}$ definiert als

$$d_{i+1}(x) = \begin{cases} 0 & \text{falls } x \leq 0, \\ \sum_{j=1}^{x} j & \text{falls } 0 < x \leq i, \\ \bot & \text{sonst.} \end{cases}$$

Dann erhalten wir $f^0(\bot) = d_0$ und $f(d_i) = d_{i+1}$ für alle $i \in \mathbb{N}$, weil

$$\begin{aligned} f'_{d_i}(d') &= \begin{cases} 0 & \text{falls } d' \leq 0, \\ d' + d_i(d' - 1) & \text{falls } d' > 0, \\ \bot & \text{sonst,} \end{cases} \\ &= \begin{cases} 0 & \text{falls } d' \leq 0, \\ \sum_{j=1}^{d'} j & \text{falls } 0 < d' \leq i, \\ \bot & \text{sonst.} \end{cases} \end{aligned}$$

Mit Induktion folgt also, dass $f^i(\bot) = d_i$ für alle $i \in \mathbb{N}$. Wir erhalten dann für den kleinsten Fixpunkt, dass

$$(\mu f)(d') = (\sqcup \lambda i.f^i(\bot))(d') = \begin{cases} 0 & \text{falls } d' \leq 0, \\ \sum_{j=1}^{d'} j & \text{falls } d' > 0, \\ \bot & \text{sonst.} \end{cases}$$

Insgesamt erhalten wir $[\![\text{s } 10]\!]_{\eta \mathbin{@} [s/\mu f]} = \sum_{i=1}^{10} = 55$.

Aufgabe 2. Wir haben bisher **if** e_1 **then** e_2 **else** e_3 verwendet, was in der Basissyntax aber nicht erlaubt ist.

(a) Definieren Sie eine geeignete Semantik für **if** e_1 **then** e_2 **else** e_3.

Lösung.

$$[\![\textbf{if } \text{e}_1 \textbf{ then } \text{e}_2 \textbf{ else } \text{e}_3]\!]_\eta = \begin{cases} [\![\text{e}_2]\!]_\eta & \text{falls } [\![\text{e}_1]\!]_\eta = \underline{\text{True}}, \\ [\![\text{e}_3]\!]_\eta & \text{falls } [\![\text{e}_1]\!]_\eta = \underline{\text{False}}, \\ \bot & \text{sonst.} \end{cases}$$

(b) Geben Sie eine Übersetzung von **if** e_1 **then** e_2 **else** e_3 in die Basissyntax an.

Lösung. **case** e_1 **of** $\{\underline{\text{True}} \text{ -> } e_2 \text{ ; } \underline{\text{False}} \text{ -> } e_3\}$.

(c) Bestimmen Sie die Semantik von Ihrer Übersetzung. Stimmt diese mit Ihrer Semantik für **if** e_1 **then** e_2 **else** e_3 überein?

Lösung.

$$[\![\textbf{case } \text{e}_1 \textbf{ of } \{\underline{\text{True}} \text{ -> } \text{e}_2 \text{ ; } \underline{\text{False}} \text{ -> } \text{e}_3\}]\!]_\eta = \begin{cases} [\![\text{e}_2]\!]_\eta & \text{falls } [\![\text{e}_1]\!]_\eta = \underline{\text{True}}, \\ [\![\text{e}_3]\!]_\eta & \text{falls } [\![\text{e}_1]\!]_\eta = \underline{\text{False}}, \\ \bot & \text{sonst.} \end{cases}$$

Die Semantiken stimmen also überein.