Programmierung

Lösungen zum 13. Übungsblatt

Zeitraum: 11. – 15. Juli 2016

Übung 1

Induktionsanfang (IA) mit `r = []`: Sei `x :: Int`. Dann gilt:

$$\begin{aligned}
\texttt{sumTree (toTree [] x)} &\overset{12}{=} \texttt{sumTree (Leaf x)}\\
&\overset{4}{=} \texttt{x}\\
&= \texttt{x * 1}\\
&\overset{8}{=} \texttt{x * prod []}
\end{aligned}$$

Induktionsvoraussetzung (IV): Sei `r :: [Int]` beliebig aber fest so dass

`sumTree (toTree r x) = x * prod r`

für jedes `x :: Int`.

Induktionsschritt (IS): Sei `a :: Int` beliebig.

$$\begin{aligned}
\texttt{sumTree(toTree (a : r) x)} &\overset{13}{=} \texttt{sumTree(Node (toTree r (2 * a * x)) (toTree r (-a * x)))}\\
&\overset{5}{=} \texttt{sumTree (toTree r (2 * a * x)) + sumTree (toTree r (-a * x))}\\
&\overset{IV}{=} \texttt{2 * a * x * prod r + (-a * x) * prod r}\\
&= \texttt{a * x * prod r}\\
&= \texttt{x * (a * prod r)}\\
&\overset{9}{=} \texttt{x * prod (a : r)}
\end{aligned}$$

Übung 2

(a)

$$\begin{aligned}
&(y(\lambda x.y)x)((\lambda x\,\underbrace{y.x(\lambda z.z)y}_{GV=\{y,z\}})\underbrace{(\lambda x.y)}_{FV=\{y\}})\\
&\Rightarrow_\alpha (y(\lambda x.y)x)((\lambda x\,\underbrace{y_1.x(\lambda z.z)y_1}_{GV=\{y_1,z\}})\underbrace{(\lambda x.y)}_{FV=\{y\}})\\
&\Rightarrow_\beta (y(\lambda x.y)x)(\lambda y_1.(\lambda x.\underbrace{\ y\ }_{GV=\emptyset})\underbrace{(\lambda z.z)}_{FV=\emptyset}y_1)\\
&\Rightarrow_\beta (y(\lambda x.y)x)(\lambda y_1.y\,y_1)
\end{aligned}$$

(b)

$$\begin{aligned}
\langle G\rangle = (\lambda f\,x\,y.\langle ite\rangle&(\langle iszero\rangle x)\ y\\
&(\langle ite\rangle(\langle iszero\rangle(\langle mod\rangle x\,\langle 2\rangle))\\
&\quad(\langle add\rangle\langle 3\rangle(f\,(\langle pred\rangle x)(\langle succ\rangle(\langle succ\rangle y))))\\
&\quad(\langle mult\rangle\langle 2\rangle(f\,(\langle pred\rangle x)(\langle succ\rangle y)))))
\end{aligned}$$

(c)

Nebenrechnung: $\langle Y\rangle\langle F\rangle = (\lambda h.\,(\lambda y.h\,(y\,y))\,(\lambda y.h\,(y\,y)))\langle F\rangle$
$\Rightarrow_\beta \underbrace{(\lambda y.\langle F\rangle(y\,y))\,(\lambda y.\langle F\rangle(y\,y))}_{=\langle Y_F\rangle} \Rightarrow_\beta \ \langle F\rangle\langle Y_F\rangle$

$\langle Y\rangle\langle F\rangle\langle 1\rangle\langle 5\rangle$

$\Rightarrow^* \ \langle F\rangle\langle Y_F\rangle\langle 1\rangle\langle 5\rangle$
$\Rightarrow^* \ \langle ite\rangle(\underbrace{\langle iszero\rangle\langle 1\rangle}_{\Rightarrow^* \langle false\rangle})(\langle succ\rangle(\langle succ\rangle\langle 5\rangle))$
$(\langle add\rangle\langle 5\rangle(\ \underbrace{\langle Y_F\rangle}_{\Rightarrow^* \langle F\rangle\langle Y_F\rangle}\ (\underbrace{\langle pred\rangle\langle 1\rangle}_{\Rightarrow^* \langle 0\rangle})(\underbrace{\langle succ\rangle\langle 1\rangle}_{\Rightarrow^* \langle 2\rangle})$

$\Rightarrow^* \ \langle add\rangle\langle 5\rangle(\langle F\rangle\langle Y_F\rangle\langle 0\rangle\langle 2\rangle)$
$\Rightarrow^* \ \langle add\rangle\langle 5\rangle(\langle ite\rangle(\underbrace{\langle iszero\rangle\langle 0\rangle}_{\Rightarrow^* \langle true\rangle})\underbrace{(\langle succ\rangle(\underbrace{\langle succ\rangle\langle 2\rangle}_{\Rightarrow^* \langle 3\rangle}))}_{\Rightarrow^* \langle 4\rangle}$
$(\langle add\rangle\langle 2\rangle(\langle Y_F\rangle(\langle pred\rangle\langle 0\rangle)(\langle succ\rangle\langle 2\rangle))))$

$\Rightarrow^* \ \langle add\rangle\langle 5\rangle\langle 4\rangle$

$\Rightarrow^* \ \langle 9\rangle$

Übung 3

(a)
$$tab_{g+lDecl} = [f/(\text{proc},1), g/(\text{proc},2), x/(\text{var},\text{global},1), y/(\text{var},\text{global},2),$$
$$b/(\text{var-ref},-2), a/(\text{var},\text{lokal},1)]$$

AM_1-Code:

```
2.1.1:  LOAD (global,1); LIT 0; GT;
        JMC 2.1.2;
        LOAD (global,1); LOAD (lokal,1); SUB; STORE (global,1);
        LOAD (global,2); PUSH;
        LOADA (lokal,1); PUSH;
        CALL 1;
        JMP 2.1.1;
2.1.2:  LOAD(lokal,-2); PUSH; CALL 2;
```

(b)

BZ	DK	LK	REF	Inp	Out
13	ε	$0:3:0:9:7$	3	ε	ε
14	7	$0:3:0:9:7$	3	ε	ε
15	ε	$0:3:0:9:7:7$	3	ε	ε
16	4	$0:3:0:9:7:7$	3	ε	ε
17	ε	$0:3:0:9:7:7:4$	3	ε	ε
4	ε	$0:3:0:9:7:7:4:18:3$	9	ε	ε
5	ε	$0:3:0:9:7:7:4:18:3$	9	ε	ε
6	9	$0:3:0:9:7:7:4:18:3$	9	ε	ε
7	$7:9$	$0:3:0:9:7:7:4:18:3$	9	ε	ε
8	16	$0:3:0:9:7:7:4:18:3$	9	ε	ε
9	ε	$16:3:0:9:7:7:4:18:3$	9	ε	ε
18	ε	$16:3:0:9:7$	3	ε	ε
19	ε	$16:3:0:9:7$	3	ε	16
3	ε	16	0	ε	16
0	ε	16	0	ε	16

Übung 4

(a) $SI = (z = (x - x1) \cdot 3y) \wedge (x1 \geq 0)$

(b) $A = C = SI$

$B = SI \wedge \neg\pi = (z = (x - x1) \cdot 3y) \wedge (x1 \geq 0) \wedge (x1 \leq 0)$

$D = SI \wedge \pi = (z = (x - x1) \cdot 3y) \wedge (x1 \geq 0) \wedge (x1 > 0)$

Zusatzaufgabe 1

```haskell
-- (a)
expo :: Int -> Int -> Int
expo 0 0 = error "0^0 ist nicht definiert!"
expo x 1 = x
expo _ 0 = 1
expo x n = x * expo x (n - 1)

-- (b)
data Tree a = Branch (Tree a) (Tree a) | Leaf a deriving Show

check :: Tree a -> Int -> Bool
check (Leaf _)     k = k == 0
check (Branch l r) k = check l (k - 1) || check r (k - 1)

-- (c)
test :: [Int] -> Bool
test []       = True
test (z : zs) = notElem z zs && test zs
  where
    notElem :: Int -> [Int] -> Bool
    notElem _ []       = True
    notElem x (y : ys) = x /= y && notElem x ys
```

Hinweis: Die Funktion `notElem` ist normalerweise im Modul `Prelude` bereits mit folgendem Typ vordefiniert: `notElem :: Eq a => a -> [a] -> Bool`.

Zusatzaufgabe 2

(a)

$$
\begin{aligned}
&\left\{\begin{pmatrix} \underline{\sigma}(\tau(x_4,x_2),\sigma(\gamma(x_1),x_1)) \\ \underline{\sigma}(x_1,\sigma(x_3,\tau(\alpha,x_2))) \end{pmatrix}\right\} \\
\overset{\text{Dek.}}{\Longrightarrow}\ &\left\{\begin{pmatrix} \tau(x_4,x_2) \\ x_1 \end{pmatrix}, \begin{pmatrix} \underline{\sigma}(\gamma(x_1),x_1) \\ \underline{\sigma}(x_3,\tau(\alpha,x_2)) \end{pmatrix}\right\} \\
\overset{\text{Dek.}}{\Longrightarrow}\ &\left\{\begin{pmatrix} \tau(x_4,x_2) \\ \underline{x_1} \end{pmatrix}, \begin{pmatrix} \gamma(\underline{x_1}) \\ x_3 \end{pmatrix}, \underline{\begin{pmatrix} x_1 \\ \tau(\alpha,x_2) \end{pmatrix}}\right\} \\
\overset{\text{Sub.}}{\Longrightarrow}\ &\left\{\begin{pmatrix} \underline{\tau}(x_4,x_2) \\ \underline{\tau}(\alpha,x_2) \end{pmatrix}, \begin{pmatrix} \gamma(\tau(\alpha,x_2)) \\ x_3 \end{pmatrix}, \begin{pmatrix} x_1 \\ \tau(\alpha,x_2) \end{pmatrix}\right\} \\
\overset{\text{Dek.}}{\Longrightarrow}\ &\left\{\begin{pmatrix} x_4 \\ \alpha \end{pmatrix}, \underline{\begin{pmatrix} x_2 \\ x_2 \end{pmatrix}}, \begin{pmatrix} \gamma(\tau(\alpha,x_2)) \\ x_3 \end{pmatrix}, \begin{pmatrix} x_1 \\ \tau(\alpha,x_2) \end{pmatrix}\right\} \\
\overset{\text{El.}}{\Longrightarrow}\ &\left\{\begin{pmatrix} x_4 \\ \alpha \end{pmatrix}, \underline{\begin{pmatrix} \gamma(\tau(\alpha,x_2)) \\ x_3 \end{pmatrix}}, \begin{pmatrix} x_1 \\ \tau(\alpha,x_2) \end{pmatrix}\right\} \\
\overset{\text{Vert.}}{\Longrightarrow}\ &\left\{\begin{pmatrix} x_4 \\ \alpha \end{pmatrix}, \begin{pmatrix} x3 \\ \gamma(\tau(\alpha,x_2)) \end{pmatrix}, \begin{pmatrix} x_1 \\ \tau(\alpha,x_2) \end{pmatrix}\right\}
\end{aligned}
$$

allgemeinster Unifikator: $x_1 \mapsto \tau(\alpha,x_2),\ x_2 \mapsto x_2,\ x_3 \mapsto \gamma(\tau(\alpha,x_2)),\ x_4 \mapsto \alpha$

(b) $t_1 = \gamma(x_1)$, $t_2 = \gamma(\gamma(x_1))$