Programmierung

Aufgabe 1 (AGS 12.1.63)

```
import Prelude hiding (equal)

retrieve :: [a] -> Int -> a
retrieve (a:as) n | n == 1 = a
                  | n > 1  = retrieve as (n-1)

data CV = C Char | V Int

substitute :: [CV] -> [[Char]] -> [Char]
substitute _ [] = []
substitute sub ((C a):xs) = a : substitute sub xs
substitute sub ((V n):xs) = retrieve sub n ++ substitute sub xs

data Tree = Node [Tree]

maxrank :: Tree -> Int
maxrank (Node xs) = foldr (max . maxrank) (length xs) xs

-- oder
maxrank :: Tree -> Int
maxrank (Node xs) = go xs
  where go [] = length xs
    go (x:xs) = max (maxrank x) (go xs)

equal :: Tree -> Tree -> Bool
equal (Node xs) (Node ys) = length xs == length ys
                            && all (==True) (zipWith equal xs ys)

-- oder
equal :: Tree -> Tree -> Bool
equal (Node xs) (Node ys) = go xs ys
  where go []     []     = True
    go (x:xs) (y:ys) = equal x y && go xs ys
    go _      _      = False
```

Aufgabe 2 (AGS 12.2.17)

(a)

$$\left\{\begin{pmatrix}\delta(\gamma(\sigma(\alpha,x_1)),\beta,x_3)\\ \delta(\gamma(x_3),\beta,\sigma(x_2,x_1))\end{pmatrix}\right\} \overset{\text{Dek.}}{\Longrightarrow} \left\{\begin{pmatrix}\gamma(\sigma(\alpha,x_1))\\ \gamma(x_3)\end{pmatrix},\begin{pmatrix}\beta\\ \beta\end{pmatrix},\begin{pmatrix}x_3\\ \sigma(x_2,x_1)\end{pmatrix}\right\}$$
$$\overset{2\times\text{Dek.}}{\Longrightarrow} \left\{\begin{pmatrix}\sigma(\alpha,x_1)\\ x_3\end{pmatrix},\begin{pmatrix}x_3\\ \sigma(x_2,x_1)\end{pmatrix}\right\} \overset{\text{Subst.}}{\Longrightarrow} \left\{\begin{pmatrix}\sigma(\alpha,x_1)\\ \sigma(x_2,x_1)\end{pmatrix},\begin{pmatrix}x_3\\ \sigma(x_2,x_1)\end{pmatrix}\right\}$$
$$\overset{\text{Dek.}}{\Longrightarrow} \left\{\begin{pmatrix}\alpha\\ x_2\end{pmatrix},\begin{pmatrix}x_1\\ x_1\end{pmatrix},\begin{pmatrix}x_3\\ \sigma(x_2,x_1)\end{pmatrix}\right\} \overset{\text{Elim.}}{\Longrightarrow} \left\{\begin{pmatrix}\alpha\\ x_2\end{pmatrix},\begin{pmatrix}x_3\\ \sigma(x_2,x_1)\end{pmatrix}\right\}$$
$$\overset{\text{Tausch}}{\Longrightarrow} \left\{\begin{pmatrix}x_2\\ \alpha\end{pmatrix},\begin{pmatrix}x_3\\ \sigma(x_2,x_1)\end{pmatrix}\right\} \overset{\text{Subst.}}{\Longrightarrow} \left\{\begin{pmatrix}x_2\\ \alpha\end{pmatrix},\begin{pmatrix}x_3\\ \sigma(\alpha,x_1)\end{pmatrix}\right\}$$

Unifikatoren:

- $\varphi(x_1)=x_1,\varphi(x_2)=\alpha,\varphi(x_3)=\sigma(\alpha,x_1)$ (allgemeinster Unifikator)
- $\varphi_2(x_1)=\alpha,\varphi_2(x_2)=\alpha,\varphi_2(x_3)=\sigma(\alpha,\alpha)$
- $\varphi_3(x_1)=\gamma(\alpha),\varphi_3(x_2)=\alpha,\varphi_3(x_3)=\sigma(\alpha,\gamma(\alpha))$

(b) $\begin{pmatrix}x_1\\ \gamma(x_1)\end{pmatrix}, \quad \begin{pmatrix}\sigma(\gamma(x_1),x_2)\\ \sigma(x_1,x_2)\end{pmatrix}$

Aufgabe 3 (AGS 12.3.33)

(IA)

$$\begin{aligned}
&\texttt{treesum (foo [])}\\
=&\texttt{treesum (Nil (-1))} && (4)\\
=&\texttt{-1} && (11)\\
=&\,0-3^0\\
=&\,0-3^{\texttt{(length [])}} && (19)\\
=&\texttt{trip []}-3^{\texttt{(length [])}} && (15)
\end{aligned}$$

(IS) Sei `xs :: [Int]`, sodass gilt:

$$\texttt{treesum (foo xs)} = \texttt{trip xs} - 3^{\texttt{(length xs)}}. \qquad \text{(IV)}$$

Für jedes `x :: Int` gilt

$$
\begin{aligned}
&\texttt{treesum (foo (x:xs))}\\
=&\texttt{treesum (Node x (foo xs) (bar (foo xs)))} && (5)\\
=&\texttt{treesum (Node x (foo xs) (Node 0 (foo xs) (foo xs)))} && (8)\\
=&\texttt{treesum (foo xs) + x + treesum (Node 0 (foo xs) (foo xs))} && (12)\\
=&\texttt{treesum (foo xs) + x + (treesum (foo xs) + 0 + treesum (foo xs))} && (12)\\
=&\,\texttt{x} + 3 \cdot (\texttt{treesum (foo xs)})\\
=&\,\texttt{x} + 3 \cdot (\texttt{trip xs} - 3^{\texttt{(length xs)}}) && (\mathrm{IV})\\
=&\texttt{(x + 3*(trip xs))} - 3^{\texttt{(1 + length xs)}}\\
=&\texttt{(x + 3*(trip xs))} - 3^{\texttt{(length (x:xs))}} && (20)\\
=&\texttt{trip (x:xs)} - 3^{\texttt{(length (x:xs))}}. && (16)
\end{aligned}
$$

Aufgabe 4 (AGS 12.4.38)

(a) $\langle G \rangle = \Big(\lambda g x\, y.\langle \mathrm{ite} \rangle \;\; (\langle \mathrm{iszero} \rangle\; x) \;\; y \;\; (g\; (\langle \mathrm{pred} \rangle\; x)\; (\langle \mathrm{add} \rangle\; x\; y)\,) \Big)$

(b)

$$
\begin{aligned}
&(\lambda f\, \underbrace{x.f\,x\,x}_{\mathrm{GV}=\{x\}})\; \underbrace{(\lambda y\, z.x\, y\, z)}_{\mathrm{FV}=\{\mathrm{x}\}}\; w\\
&\Rightarrow_\alpha (\lambda f\, \underbrace{x_1.f\,x_1\,x_1}_{\mathrm{GV}=\{x_1\}})\; \underbrace{(\lambda y\, z.x\, y\, z)}_{\mathrm{FV}=\{\mathrm{x}\}}\; w\\
&\Rightarrow_\beta (\lambda x_1.\underbrace{(\lambda y\, z.x\, y\, z)\, x_1\, x_1}_{\mathrm{GV}=\{\mathrm{y,z}\}})\; \underbrace{w}_{\mathrm{FV}=\{\mathrm{w}\}}\\
&\Rightarrow_\beta (\lambda y\; \underbrace{z.x\, y\, z}_{\mathrm{GV}=\{\mathrm{z}\}})\; \underbrace{w}_{\mathrm{FV}=\{\mathrm{w}\}}\; w\\
&\Rightarrow_\beta (\lambda z.\underbrace{x\, w\, z}_{\mathrm{GV}=\emptyset})\; \underbrace{w}_{\mathrm{FV}=\{\mathrm{w}\}}\\
&\Rightarrow_\beta x\, w\, w
\end{aligned}
$$

oder

$$
\begin{aligned}
&(\lambda f\, \underbrace{x.f\,x\,x}_{\mathrm{GV}=\{x\}})\; \underbrace{(\lambda y\, z.x\, y\, z)}_{\mathrm{FV}=\{\mathrm{x}\}}\; w\\
&\Rightarrow_\alpha (\lambda f\, \underbrace{x_1.f\,x_1\,x_1}_{\mathrm{GV}=\{x_1\}})\; \underbrace{(\lambda y\, z.x\, y\, z)}_{\mathrm{FV}=\{\mathrm{x}\}}\; w\\
&\Rightarrow_\beta (\lambda x_1.(\lambda y\; \underbrace{z.x\, y\, z}_{\mathrm{GV}=\{z\}})\; \underbrace{x_1}_{\mathrm{FV}=\{\mathrm{x_1}\}}\; x_1)\; w\\
&\Rightarrow_\beta (\lambda x_1.(\lambda z.\underbrace{x\, x_1\, z}_{\mathrm{GV}=\emptyset})\; \underbrace{x_1}_{\mathrm{FV}=\{\mathrm{x_1}\}})\; w\\
&\Rightarrow_\beta (\lambda x_1.\underbrace{x\, x_1\, x_1}_{\mathrm{GV}=\emptyset})\; \underbrace{w}_{\mathrm{FV}=\{\mathrm{w}\}}\\
&\Rightarrow_\beta x\, w\, w
\end{aligned}
$$

(c) $\langle Y\rangle\langle F\rangle = \Big(\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))\Big)\langle F\rangle \Rightarrow_\beta \underbrace{(\lambda x.\langle F\rangle(xx))(\lambda x.\langle F\rangle(xx))}_{=:\langle Y_F\rangle} \Rightarrow_\beta \langle F\rangle\langle Y_F\rangle$

$\langle Y\rangle\ \langle F\rangle\ \langle 2\rangle\ \langle 3\rangle\ \langle 7\rangle$

$\Rightarrow^*\langle F\rangle\langle Y_F\rangle\langle 2\rangle\ \langle 3\rangle\ \langle 7\rangle$

$\Rightarrow^*\langle \text{ite}\rangle\ \underbrace{(\langle \text{gt}\rangle(\underbrace{\langle \text{add}\rangle\ \langle 2\rangle\ \langle 3\rangle}_{\Rightarrow^*\langle 5\rangle})\ \langle 7\rangle)}_{\Rightarrow^*\text{false}}\ \langle 7\rangle\ (\langle Y_F\rangle\ (\underbrace{\langle \text{succ}\rangle\ \langle 2\rangle}_{\Rightarrow^*\langle 3\rangle})\ (\underbrace{\langle \text{succ}\rangle\ \langle 3\rangle}_{\Rightarrow^*\langle 4\rangle})\ (\underbrace{\langle \text{pred}\rangle\ \langle 7\rangle}_{\Rightarrow^*\langle 6\rangle}))$

$\Rightarrow^*\langle F\rangle\langle Y_F\rangle\langle 3\rangle\ \langle 4\rangle\ \langle 6\rangle$

$\Rightarrow^*\langle \text{ite}\rangle\ \underbrace{(\langle \text{gt}\rangle(\underbrace{\langle \text{add}\rangle\ \langle 3\rangle\ \langle 4\rangle}_{\Rightarrow^*\langle 7\rangle})\ \langle 6\rangle)}_{\Rightarrow^*\text{true}}\ \langle 6\rangle\ (...))$

$\Rightarrow^*\langle 6\rangle$

Aufgabe 5 (AGS 13.14)

```
% (a)
leq(   0,    X) :- nat(X).
leq(s(X), s(Y)) :- leq(X,Y).

% (b)
minimum([X], X).
minimum([X|Xs], X) :- minimum(Xs, Y), leq(X, Y).
minimum([X|Xs], Y) :- minimum(Xs, Y), leq(Y, X).

% (c)
                              ?-  derv(sum(pow(x,<2>),<4>), D).
{D = sum(D1,D2)}              ?-  derv(pow(x,<2>), D1), derv(<4>, D2). % 6
{D1 = prod(<2>,pow(x,<1>))} ?-  derv(<4>, D2).                        % 5
{D2 = 0}                      ?-  nat(<4>).                           % 4
                              ?-* nat(0).                             % 2
                              ?-.                                     % 1

% D = sum(prod(<2>,pow(x,<1>)), 0)
```

Aufgabe 6 (AGS 15.27)

(a)

$$tab_{\mathtt{g}} = [\mathtt{x}/(\text{var}, \text{global}, 1), \mathtt{g}/(\text{proc}, 1), \mathtt{f}/(\text{proc}, 2), \mathtt{p}/(\text{var-ref}, -2), \mathtt{l}/(\text{var}, \text{lokal}, 1)]$$

```
LOAD(global, 1);        LIT 1;                          CALL 1;
LIT 0;                  ADD;                  1.1.1:    LOADI(-2);
EQ;                     STORE(global,1);                PUSH;
JMC 1.1.1;              LOADA(lokal, 1);                CALL 2;
LOAD(global, 1);        PUSH;
```

(b)

BZ	DK	LK	REF	In	Out
11	ε	0 : 3 : 0	3	5	ε
12		5 : 3 : 0		ε	
13	5				
14	0 : 5				
15	1				

BZ	DK	LK	REF	In	Out
18	ε	5 : 3 : 0 : 5	3	ε	ε
19	1				
20	ε	5 : 3 : 0 : 5 : 1			
4		5 : 3 : 0 : 5 : 1 : 21 : 3	7		
5		5 : 3 : 0 : 5 : 1 : 21 : 3 : 0			

BZ	DK	LK	REF	In	Out
7	1 : 5	5 : 3 : 0 : 5 : 1 : 21 : 3 : 0	7	ε	ε
8	4				
9	ε	4 : 3 : 0 : 5 : 1 : 21 : 3 : 0			
21		4 : 3 : 0	3		
12					

Aufgabe 7 (AGS 16.34)

(a) SI $= (y = x^{n-i}) \wedge (i \geq 0)$

(b)

$A =$ SV $\qquad B = (x \geq 0) \wedge (i = n) \wedge (y = 1) \wedge (n \geq 0)$

$C =$ SI $\qquad D =$ SN

$E =$ SI $\wedge \neg(i > 0) \qquad F =$ SI $\wedge (i > 0)$

$G =$ `{y=y*x; i=i-1;}` $\qquad H =$ SI

$I = (y = x^n) \qquad J = I$

(c) Compregel (SN oder SV sind auch möglich), Elternknoten IR

Aufgabe 8 (AGS 17.6 a, 17.6)

(a) $f(n,x) = \begin{cases} x & \text{für } n = 0 \\ f(n-1,(n+1)^2+x) & \text{für } n \geq 1 \end{cases}$,

Aufruf: $s(n) = f(n,0)$

```
module Main where

f :: Int -> Int -> Int
f x1 x2 = if x1 == 0
            then x2
            else (f (x1 - 1) (((x1 + 1) * (x1 + 1)) + x2)

main = do x1 <- readLn
          print (f x1 0)
```

(b)

```
 1: READ 1;     6: JMP 7;    11: LOAD 1;   16: MUL;      21: STORE 1;
 2: LOAD 1;     7: LOAD 1;   12: LIT 1;    17: STORE 2;  22: WRITE 1;
 3: LIT 1;      8: LIT 0;    13: SUB;      18: STORE 1;  23: JMP 0;
 4: STORE 2;    9: GT;       14: LOAD 1;   19: JMP 7;
 5: STORE 1;   10: JMC 20;   15: LOAD 2;   20: LOAD 2;
```