Unifikation

- Motivation: Typüberprüfung

```
f :: (t, Char) -> (t, [Char])
g :: (Int, [u]) -> Int

h = g . f
```

- Beide Typausdrücke können in Übereinstimmung gebracht werden gdw. die Typterme trans((t, [Char])) und trans((Int, [u])) unifizierbar sind.
- Ziel: in der oberen Zeile nur paarweise verschiedene Variablen
- beliebter Fehler: Verwechslung der Elimination von Variablentupeln (x_i, x_i) mit der Dekomposition von nullären Symbolen (α, α)
- Occur-Check: (x_i, t) kann nur substituiert werden, wenn x_i nicht in t vorkommt.

Übung 1

$$\left\{ \begin{pmatrix} \sigma(\gamma(x_2), \sigma(\gamma(\alpha), x_3)) \\ \sigma(x_1, \sigma(\gamma(\alpha), \sigma(\alpha, x_1))) \end{pmatrix} \right\}$$

$$\xRightarrow{\text{Dek.}} \left\{ \begin{pmatrix} \gamma(x_2) \\ x_1 \end{pmatrix}, \begin{pmatrix} \sigma(\gamma(\alpha), x_3) \\ \sigma(\gamma(\alpha), \sigma(\alpha, x_1)) \end{pmatrix} \right\}$$

$$\xRightarrow{\text{Dek.}} \left\{ \begin{pmatrix} \gamma(x_2) \\ x_1 \end{pmatrix}, \begin{pmatrix} \gamma(\alpha) \\ \gamma(\alpha) \end{pmatrix}, \begin{pmatrix} x_3 \\ \sigma(\alpha, x_1) \end{pmatrix} \right\}$$

$$\xRightarrow{2 \cdot \text{Dek.}} \left\{ \begin{pmatrix} \gamma(x_2) \\ x_1 \end{pmatrix}, \begin{pmatrix} x_3 \\ \sigma(\alpha, x_1) \end{pmatrix} \right\}$$

$$\xRightarrow{\text{Vert.}} \left\{ \begin{pmatrix} x_1 \\ \gamma(x_2) \end{pmatrix}, \begin{pmatrix} x_3 \\ \sigma(\alpha, x_1) \end{pmatrix} \right\}$$

$$\xRightarrow{\text{Sub.}} \left\{ \begin{pmatrix} x_1 \\ \gamma(x_2) \end{pmatrix}, \begin{pmatrix} x_3 \\ \sigma(\alpha, \gamma(x_2)) \end{pmatrix} \right\}$$

Übung 1

allgemeinster Unifikator:

$$x_1 \mapsto \gamma(x_2)$$
$$x_2 \mapsto x_2$$
$$x_3 \mapsto \sigma(\alpha, \gamma(x_2))$$

weitere Unifikatoren:

$$x_1 \mapsto \gamma(\alpha)$$
$$x_2 \mapsto \alpha$$
$$x_3 \mapsto \sigma(\alpha, \gamma(\alpha))$$

$$x_1 \mapsto \gamma(\gamma(\alpha))$$
$$x_2 \mapsto \gamma(\alpha)$$
$$x_3 \mapsto \sigma(\alpha, \gamma(\gamma(\alpha)))$$

Übung 2

Induktionsanfang (IA)

Sei `xs = []`.

$$
\begin{aligned}
& \texttt{sum (foo [])} \\
\overset{(2)}{=}\ & \texttt{sum []} \\
\overset{(6)}{=}\ & 0 \\
=\ & 2 * 0 - 0 \\
\overset{(6)}{=}\ & 2 * \texttt{sum []} - 0 \\
\overset{(10)}{=}\ & 2 * \texttt{sum []} - \texttt{length []}
\end{aligned}
$$

Induktionsvoraussetzung (IV)

Sei `xs :: [Int]`, sodass gilt:

$$\texttt{sum (foo xs)} = 2 * \texttt{sum xs} - \texttt{length xs}$$

Übung 2

Induktionsschritt (IS)

Für alle `x :: Int` zeigen wir, dass

`sum (foo (x:xs)) = 2 * sum (x:xs) - length (x:xs)`

gilt:

$$
\begin{aligned}
& \texttt{sum (foo (x:xs))} \\
\overset{(3)}{=}\ & \texttt{sum (x : x : (-1) : foo xs)} \\
\overset{(7)}{=}\ & \texttt{x + sum (x : (-1) : foo xs)} \\
\overset{(7)}{=}\ & \texttt{x + x + sum ((-1) : foo xs)} \\
\overset{(7)}{=}\ & \texttt{x + x + (-1) + sum (foo xs)} \\
\overset{(IV)}{=}\ & \texttt{x + x + (-1) + 2 * sum xs - length xs} \\
=\ & \texttt{2 * (x + sum xs) - (1 + length xs)} \\
\overset{(7)}{=}\ & \texttt{2 * sum (x:xs) - (1 + length xs)} \\
\overset{(11)}{=}\ & \texttt{2 * sum (x:xs) - length (x:xs)} \qquad \square
\end{aligned}
$$

Übung 3

Induktionsanfang (IA)

Sei `t = Leaf x` für jedes `a, x :: Float`.

```
    sum (add (Leaf x) a)
(4)
 =  sum (Leaf (x+a))
(12)
 =  x+a
(12)
 =  sum (Leaf x) + a
(8)
 =  sum (rev (Leaf x)) + a
```

Induktionsvoraussetzung (IV)

Seien `t1, t2 :: Tree`, sodass für beliebige `a1, a2 :: Float` gilt:

```
sum (add t1 a1) = sum (rev t1) + a1        (IV1)
sum (add t2 a2) = sum (rev t2) + a2        (IV2)
```

Übung 3

Induktionsschritt (IS)

Für alle `t = Branch x t1 t2` und `a, x :: Float` zeigen wir, dass

```
sum (add t a) = sum (rev t) + a
```

gilt:

$$
\begin{aligned}
& \texttt{sum (add (Branch x t1 t2) a)} \\
\overset{(5)}{=}\ & \texttt{sum (Branch (x+a/3) (add t1 (a/3)) (add t2 (a/3)))} \\
\overset{(13)}{=}\ & \texttt{x + a/3 + sum (add t1 (a/3)) + sum (add t2 (a/3))} \\
\overset{(IV1)}{=}\ & \texttt{x + a/3 + sum (rev t1) + a/3 + sum (add t2 (a/3))} \\
\overset{(IV2)}{=}\ & \texttt{x + a/3 + sum (rev t1) + a/3 + sum (rev t2) + a/3} \\
=\ & \texttt{x + sum (rev t2) + sum (rev t1) + a} \\
\overset{(13)}{=}\ & \texttt{sum (Branch x (rev t2) (rev t1)) + a} \\
\overset{(9)}{=}\ & \texttt{sum (rev (Branch x t1 t2)) + a} \qquad \square
\end{aligned}
$$

Übung 4

t	$FV(t)$	$GV(t)$
$(\lambda x.xy)(\lambda y.y)$	$\{y\}$	$\{x, y\}$
$(\lambda x.(\lambda y.z(\lambda z.z(\lambda x.y))))$	$\{z\}$	$\{x, y, z\}$
$(\lambda x.(\lambda y.xz(yz)))(\lambda x.y(\lambda y.y))$	$\{y, z\}$	$\{x, y\}$

Übung 4

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

Übung 4

$$
\begin{aligned}
& (\lambda x.\, \underbrace{(\lambda y.(\lambda z.z))}_{GV=\{y,z\}}\,) \quad \underbrace{x}_{FV=\{x\}} \quad (+y1) \\
\Rightarrow_\beta \quad & (\lambda y.\, \underbrace{(\lambda z.z)}_{GV=\{z\}}\,)\ \underbrace{(+y1)}_{FV=\{y\}} \\
\Rightarrow_\beta \quad & (\lambda z.z)
\end{aligned}
$$

Übung 4

$$
\begin{aligned}
& (\lambda x.(\lambda y.x(\lambda z.yz)))(((\lambda x.\underbrace{(\lambda y.y)}_{GV=\{y\}})\underbrace{8}_{FV=\emptyset})(\lambda x.(\lambda y.y)x)) \\
\Rightarrow_\beta\ & (\lambda x.(\lambda y.x(\lambda z.yz)))((\lambda y.y)(\lambda x.(\lambda y.\underbrace{y}_{GV=\emptyset})\underbrace{x}_{FV=\{x\}})) \\
\Rightarrow_\beta\ & (\lambda x.(\lambda y.x(\lambda z.yz)))((\lambda y.\underbrace{y}_{GV=\emptyset})\underbrace{(\lambda x.x)}_{FV=\emptyset}) \\
\Rightarrow_\beta\ & (\lambda x.\underbrace{(\lambda y.x(\lambda z.yz))}_{GV=\{y,z\}})\underbrace{(\lambda x.x)}_{FV=\emptyset} \\
\Rightarrow_\beta\ & (\lambda y.(\lambda x.\underbrace{x}_{GV=\emptyset})\underbrace{(\lambda z.yz)}_{FV=\{y\}}) \\
\Rightarrow_\beta\ & (\lambda y.(\lambda z.yz)) = (\lambda yz.yz)
\end{aligned}
$$

Übung 4

$$
\begin{aligned}
& (\lambda h.(\lambda x.h(xx))(\lambda x.h(xx)))((\lambda x.\underbrace{x}_{GV=\emptyset})\underbrace{(+15)}_{FV=\emptyset}) \\
\Rightarrow_\beta\ & (\lambda h.\underbrace{(\lambda x.h(xx))(\lambda x.h(xx))}_{GV=\{x\}})\underbrace{(+15)}_{FV=\emptyset} \\
\Rightarrow_\beta\ & (\lambda x.(+15)(xx))(\lambda x.(+15)(xx)) \\
\Rightarrow_\beta\ & (\lambda x.\underbrace{(+15)(xx)}_{GV=\{x\}})\underbrace{(\lambda x.(+15)(xx))}_{FV=\emptyset} \\
\Rightarrow_\beta\ & (+15)((\lambda x.(+15)(xx))(\lambda x.(+15)(xx))) \\
\Rightarrow_\beta\ & \ldots\ \text{(endlose Rekursion)}
\end{aligned}
$$

Übung 4

$$
\begin{aligned}
& (\lambda f.\underbrace{(\lambda a.(\lambda b.fab))}_{GV=\{a,b\}})\underbrace{(\lambda x.(\lambda y.x))}_{FV=\emptyset} \\
\Rightarrow_\beta\ & (\lambda a.(\lambda b.(\lambda x.\underbrace{(\lambda y.x)}_{GV=\{y\}})\underbrace{a}_{FV=\{a\}}b)) \\
\Rightarrow_\beta\ & (\lambda a.(\lambda b.(\lambda y.\underbrace{a}_{GV=\emptyset})\underbrace{b}_{FV=\{b\}})) \\
\Rightarrow_\beta\ & (\lambda a.(\lambda b.a)) = (\lambda ab.a)
\end{aligned}
$$