

# PROGRAMMIERUNG

## Übung 4: Haskell – Typpolymorphie & Unifikation

**Eric Kunze**

`eric.kunze@mailbox.tu-dresden.de`

TU Dresden, 3. Mai 2019

# Unifikation

## Motivation: Typüberprüfung

```
f :: (t, Char) -> (t, [Char])  
f (...) = ...
```

```
g :: (Int, [u]) -> Int  
g (...) = ...
```

```
h = g . f
```

Beide Typausdrücke können in Übereinstimmung gebracht werden, wenn die Typterme  $trans((t, [Char]))$  und  $trans((Int, [u]))$  unifizierbar sind

$\rightarrow t = Int$  und  $u = Char$

# Unifikationsalgorithmus

- **Ziel.** Am Ende sollen nur paarweise verschiedene Variablen in der oberen Zeile stehen und keine Regel mehr anwendbar sein.
- **beliebter Fehler.** Verwechslung von Elimination von Variablen  $(x_i, x_i)$  und Dekomposition von nullären Symbolen  $(\alpha, \alpha)$ .

# Unifikationsalgorithmus – Regeln

- **Dekomposition.** Sei  $\delta \in \Sigma$  ein  $k$ -stelliger Konstruktor,  $s_1, \dots, s_k, t_1, \dots, t_k$  Terme über Konstruktoren und Variablen.

$$\begin{pmatrix} \delta(s_1, \dots, s_k) \\ \delta(t_1, \dots, t_k) \end{pmatrix} \rightsquigarrow \begin{pmatrix} s_1 \\ t_1 \end{pmatrix}, \dots, \begin{pmatrix} s_k \\ t_k \end{pmatrix}$$

- **Elimination.** Sei  $x$  eine Variable !

$$\begin{pmatrix} x \\ x \end{pmatrix} \rightsquigarrow \emptyset$$

- **Vertauschung.** Sei  $t$  keine Variable.

$$\begin{pmatrix} t \\ x \end{pmatrix} \rightsquigarrow \begin{pmatrix} x \\ t \end{pmatrix}$$

- **Substitution.** Sei  $x$  eine Variable,  $t$  keine Variable und  $x$  kommt nicht in  $t$  vor (occur check). Dann ersetze in jedem anderen Term die Variable  $x$  durch  $t$ .

# Aufgabe 3

$$\begin{aligned}
 & \left\{ \left( \begin{array}{ccc} \sigma(\sigma(x_1, \alpha), \sigma(\gamma(x_3), x_3)) \\ \sigma(\sigma(\gamma(x_2), \alpha), \sigma(x_2, x_3)) \end{array} \right) \right\} \\
 \xRightarrow{\text{Dek.}} & \left\{ \left( \begin{array}{ccc} \sigma(x_1, \alpha) \\ \sigma(\gamma(x_2), \alpha) \end{array} \right), \left( \begin{array}{ccc} \sigma(\gamma(x_3), x_3) \\ \sigma(x_2, x_3) \end{array} \right) \right\} \\
 \xRightarrow{\text{Dek.}} & {}^2 \left\{ \left( \begin{array}{c} x_1 \\ \gamma(x_2) \end{array} \right), \left( \begin{array}{c} \alpha \\ \alpha \end{array} \right), \left( \begin{array}{c} \gamma(x_3) \\ x_2 \end{array} \right), \left( \begin{array}{c} x_3 \\ x_3 \end{array} \right) \right\} \\
 \xRightarrow{\text{El.}} & \left\{ \left( \begin{array}{c} x_1 \\ \gamma(x_2) \end{array} \right), \left( \begin{array}{c} \alpha \\ \alpha \end{array} \right), \left( \begin{array}{c} \gamma(x_3) \\ x_2 \end{array} \right) \right\} \\
 \xRightarrow{\text{Dek.}} & \left\{ \left( \begin{array}{c} x_1 \\ \gamma(x_2) \end{array} \right), \left( \begin{array}{c} \gamma(x_3) \\ x_2 \end{array} \right) \right\} \\
 \xRightarrow{\text{Vert.}} & \left\{ \left( \begin{array}{c} x_1 \\ \gamma(x_2) \end{array} \right), \left( \begin{array}{c} x_2 \\ \gamma(x_3) \end{array} \right) \right\} \quad x_2 \text{ kommt nicht in } \gamma(x_3) \text{ vor} \\
 \xRightarrow{\text{Subst.}} & \left\{ \left( \begin{array}{c} x_1 \\ \gamma(\gamma(x_3)) \end{array} \right), \left( \begin{array}{c} x_2 \\ \gamma(x_3) \end{array} \right) \right\}
 \end{aligned}$$

**allgemeinster Unifikator:**

$$x_1 \mapsto \gamma(\gamma(x_3)) \quad x_2 \mapsto \gamma(x_3) \quad x_3 \mapsto x_3$$

**weitere Unifikatoren:**

$$\begin{array}{lll}
 x_1 \mapsto \gamma(\gamma(\alpha)) & x_2 \mapsto \gamma(\alpha) & x_3 \mapsto \alpha \\
 x_1 \mapsto \gamma(\gamma(\gamma(\alpha))) & x_2 \mapsto \gamma(\gamma(\alpha)) & x_3 \mapsto \gamma(\alpha)
 \end{array}$$

**Fehlslag beim occur-check:**

$$\text{Alphabet: } \Sigma = \{\gamma^{(1)}\}$$

$$t_1 = x_1$$

$$t_2 = \gamma(x_1)$$