TECHNISCHE UNIVERSITÄT DRESDEN

PROGRAMMIERUNG

Übung 5: Unifikation & Induktion auf Listen

Eric Kunze
eric.kunze@mailbox.tu-dresden.de

TU Dresden, 10. Mai 2019

Unifikationsalgorithmus – Regeln

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

$$\begin{pmatrix} \delta(s_1, \ldots, s_k) \\ \delta(t_1, \ldots, t_k) \end{pmatrix} \quad \leadsto \quad \begin{pmatrix} s_1 \\ t_1 \end{pmatrix}, \ldots, \begin{pmatrix} s_k \\ t_k \end{pmatrix}$$

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

$$\begin{pmatrix} x \\ x \end{pmatrix} \quad \leadsto \quad \emptyset$$

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

$$\begin{pmatrix} t \\ x \end{pmatrix} \quad \leadsto \quad \begin{pmatrix} x \\ t \end{pmatrix}$$

- **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 1

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

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

$$\stackrel{\text{Dek.}}{\Longrightarrow}^3 \left\{\begin{pmatrix}x_1\\x_1\end{pmatrix}, \begin{pmatrix}\alpha\\x_2\end{pmatrix}, \begin{pmatrix}x_2\\x_2\end{pmatrix}, \begin{pmatrix}x_3\\\gamma(x_2)\end{pmatrix}\right\}$$

$$\stackrel{\text{El.}}{\Longrightarrow}^2 \left\{\begin{pmatrix}\alpha\\x_2\end{pmatrix}, \begin{pmatrix}x_3\\\gamma(x_2)\end{pmatrix}\right\}$$

$$\stackrel{\text{Vert.}}{\Longrightarrow} \left\{\begin{pmatrix}x_2\\\alpha\end{pmatrix}, \begin{pmatrix}x_3\\\gamma(x_2)\end{pmatrix}\right\}$$

$$\stackrel{\text{Subst.}}{\Longrightarrow} \left\{\begin{pmatrix}x_2\\\alpha\end{pmatrix}, \begin{pmatrix}x_3\\\gamma(\alpha)\end{pmatrix}\right\}$$

allgemeinster Unifikator:

$x_1 \mapsto x_1 \qquad x_2 \mapsto \alpha \qquad x_3 \mapsto \gamma(\alpha)$

Teilaufgabe (b)

t_1 = `(a , [a])`

t_2 = `(Int , [Double])`

t_3 = `(b , c)`

- t_1 und t_2 sind **nicht** unifizierbar
- t_1 und t_3 sind unifizierbar mit
  `a` $\mapsto$ `a` `b` $\mapsto$ `a` `c` $\mapsto$ `[a]`
- t_2 und t_3 sind unifizierbar mit
  `b` $\mapsto$ `Int` `c` $\mapsto$ `[Double]`

Aufgabe 2

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

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

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

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

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

x_1 kommt nicht in $\gamma(x_2)$ vor

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

allgemeinster Unifikator:

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

weitere Unifikatoren:

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

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

Strukturelle Induktion

- **Allgemeine Hinweise.** Es müssen alle Variablen quantifiziert werden!
- **Was wollen wir?** Wir wollen zeigen, dass eine Eigenschaft (ein Prädikat) P für jede Liste `xs :: [a]` gilt, d.h. dass $P(\texttt{xs})$ gilt.
- **Induktionsanfang.** Wir zeigen $P(\texttt{xs})$ für `xs == []`.
  Achtung: hat P weitere freie Parameter, dann müssen auch diese quantifiziert werden!
- **Induktionsvoraussetzung.** Wir nehmen an, dass $P(\texttt{xs'})$ für eine Liste `xs' :: [a]` gilt.
  Achtung: freie Parameter!
- **Induktionsschritt.** Nutze die Induktionsvoraussetzung um zu zeigen, dass $P(\texttt{x : xs'})$ für alle `x :: a` gilt.

Aufgabe 3

- **zu zeigen.**

  `sum (foo xs) = 2 * sum xs - length xs` für alle `xs :: Int`

- **Induktionsanfang.** Sei `xs :: [Int]` mit `xs == []`.

  linke Seite: $\texttt{sum (foo [])} \overset{(2)}{=} \texttt{sum []} \overset{(6)}{=} \texttt{0}$

  rechte Seite: $\texttt{2 * sum [] - length []} \overset{(10)}{=} \texttt{2 * sum [] - 0} \overset{(6)}{=} \texttt{2 * 0 - 0 = 0}$

- **Induktionsvoraussetzung.** Sei `xs :: [Int]`, sodass gilt

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

Aufgabe 3

- **Induktionsschritt.** Für alle `x :: Int` zeigen wir, dass gilt

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

<u>Beweis.</u>

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

□