

Softwareparadigmen SS 2018, Übungsblatt 4

Abgabe: 20. Juni 2018, bis 16:00 Uhr in Ihrem Gruppen-SVN des Institutes.

Ordnerstruktur: <Repository-URL>/sol4_xx.pdf \rightarrow xx = Gruppennummer

Es sind lediglich Abgaben im PDF Format gültig. Bitte geben Sie nur **ein** Dokument mit dem Namen "sol4_xx.pdf" ab, das alle Lösungen für das dritte Aufgabenblatt beinhaltet. Die Übung findet in den bereits beim ersten Aufgabenblatt registrierten 3er-Gruppen statt.

Beispiel 1 (2,5 P.)

Berechnen Sie, falls möglich, den **Most General Unifier (MGU)** für folgende Beispiele (je 0,5 P.). Geben Sie dabei jeden Einzelschritt explizit an und begründen Sie das Ergebnis.

1. $t = p(f(Y), f(Y,b), f(a))$ und $t' = p(X, f(a,b), X)$
2. $t = q(Y, g(Y), Y)$ und $t' = q(a, g(f(U)), U)$
3. $t = r(f(a,b), g(Z), h(Z))$ und $t' = r(f(a,Y), X, h(X))$
4. $t = s(X, b, f(X))$ und $t' = s(f(Y), Y, f(a))$
5. $t = u(Y, f(a,b), g(Y))$ und $t' = u(f(U), f(X,b), g(f(X)))$

Beispiel 2 (2,5 P.)

Definieren Sie ein Prädikat **leq**(**X**, **Y**) in **LP**, welches genau dann wahr ist, wenn **X** kleiner oder gleich **Y** ist. Verwenden Sie dafür die in der Vorlesung besprochene Repräsentation der natürlichen Zahlen. Argumentieren Sie den Wahrheitswert des folgenden Faktums mit Hilfe der Resolution:
 $:= \text{leq}(s(s(0)), s(s(s(0))))$.

Beispiel 3 (3,5 P.)

Gegeben sei der Funktor **build**(**V**, **R**), der eine Liste über einen **Wert V** und eine **Restliste R** beschreibt. Für die leere Liste wird die Konstante **null** verwendet. Eine Liste [a,b] wird somit durch **build**(**a**, **build**(**b**, **null**)) repräsentiert.

1. Definieren Sie ein Prädikat **remove**(**X**, **LS1**, **LS2**) in **LP**, welches genau dann wahr ist, wenn **LS2** die Liste **LS1** ohne das Element **X** ist (1,5 P.).
2. Zeigen Sie mittels **Resolution**, dass folgende Anfrage zu **TRUE** evaluiert (2 P.):
 $:= \text{remove}(a, \text{build}(a, \text{build}(c, \text{build}(a, \text{null}))), \text{build}(c, \text{null}))$.

Beispiel 4 (4 P.)

Gegeben sei der Funktor **build**(**V**, **R**), der eine Liste über einen **Wert V** und eine **Restliste R** beschreibt. Der Wert **V** ist eine natürliche Zahl inklusive Null, die wiederum mit dem Funktor **s**(**X**) dargestellt wird. Für die leere Liste wird die Konstante **null** verwendet. Eine Liste [0,1] wird somit durch **build**(**0**, **build**(**s**(**0**), **null**)) repräsentiert.

Geben sei weiters ein Prädikat **greater**(**A**, **B**) in **LP**, welches genau dann wahr ist, wenn **A** größer

als **B** ist (**A** > **B**). Dadurch dürfen mathematische größer Beziehungen als Fakten angenommen werden (z.B. `greater(s(0), 0)`).

Zur Vereinfachung werden nur **aufsteigend sortierte** Listen betrachtet, in welchen jedes Element nur einmal vorkommt.

1. Definieren Sie ein Prädikat **merge**(**LS1**, **LS2**, **R**) in **LP**, welches genau dann wahr ist, wenn **R** die **Vereinigung** der Listen **LS1** und **LS2** ist. Dabei sollen Elemente, die sowohl in **LS1** als auch in **LS2** vorhanden sind, in der vereinigten Liste **R** nur einmal vorkommen. Die Liste **R** muss außerdem aufsteigend sortiert sein (2 P.).

Basis

merge(**LS1**, **null**, **LS1**).

merge(**null**, **LS2**, **LS2**).

2. Zeigen Sie mittels **Resolution**, dass folgende Anfrage zu **TRUE** evaluiert (2 P.):

```
:= merge( build( s(0), build( s(s(0)), null ) ),  
          build( 0, build( s(0), null ) ) ,  
          build( 0, build( s(0), build( s(s(0)), null ) ) ) ).
```

Viel Erfolg!