Softwareparadigmen SS 2017, Übungsblatt 4

Abgabe: 14. Juni 2017, bis 16:00 Uhr vor dem Sekretariat IST, Infeldgasse 16b, 2. OG

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(Y, X, g(X))** und **t' = p(f(a), Z, g(f(Z)))**
2. **t = q(Y, b, f(Y))** und **t' = q(f(X), X, f(f(a)))**
3. **t = r(f(X, f(Y)), X)** und **t' = r(f(g(a), f(X)), Y)**
4. **t = s(Y, b, i(j(b)))** und **t' = s(i(b), X, i(Y))**
5. **t = u(f(a), g(f(a)), Y)** und **t' = u(X, g(X), f(g(a)))**

Beispiel 2 (2, 5 P.)

Definieren Sie ein Prädikat **double(X, Y)** in **LP**, welches genau dann wahr ist, wenn **X** doppelt so groß ist wie **Y**. Verwenden Sie dafür die in der Vorlesung besprochene Repräsentation der natürlichen Zahlen. Argumentieren Sie den Wahrheitswert des folgenden Faktums mit Hilfe der Resolution:
:= double(s(s(s(s(0)))), s(s(0)))).

Beispiel 3 (2, 5 P.)

In Abbildung 1 können Sie einen fiktiven Stammbaum sehen. Folgende Prädikate reichen aus, um den Stammbaum zu beschreiben: **parent(X, Y)**, **male(X)** und **female(X)**.

Anmerkung: **parent(X,Y)** ist genau dann wahr, wenn **X** ein Elternteil von **Y** ist.

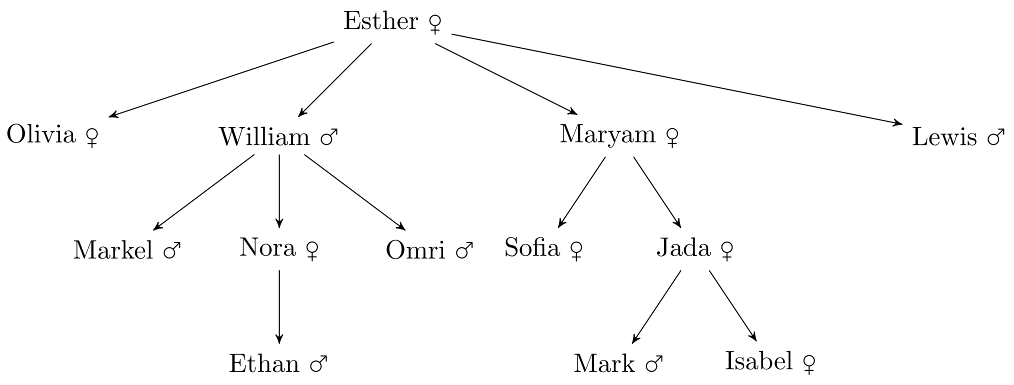

Abbildung 1: Fiktiver-Stammbaum

1. Beschreiben Sie den Stammbaum mit den oben angegebenen Prädikaten so genau wie möglich. (0,5 P.)

2. Schreiben Sie ein Prädikat **niece(X, Y)**, welches genau dann wahr ist, wenn **X** die **Nichte** von **Y** ist und beweisen Sie mittels Resolution, dass Nora die Nichte von Olivia ist. (1 P.)

   := niece(nora, olivia).

3. Schreiben Sie ein Prädikat **greatuncle(X, Y)**, welches genau dann wahr ist, wenn **X** der **Großonkel** von **Y** ist und zeigen Sie mit Hilfe der Resolution, welche Großonkel Ethan besitzt (**Variablenbelegung von X**). (1 P.)

   := greatuncle(X, ethan).

Beispiel 4 (2, 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 **subst(W, X, Y, Z)** in **LP**, welches genau dann wahr ist, wenn die Liste **Z** das Ergebnis des Ersetzens von **X** für alle Vorkommen von **W** in der Liste **Y** ist.

2. Zeigen Sie mittels **Resolution**, dass folgende Anfrage zu **TRUE** evaluiert:
   := subst(a, b, build(a, build(a, build(c, build(a, null)))), build(b, build(b, build(c, build(b, null))))).

Beispiel 5 (2, 5 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 **smaller(A, B)** in LP, welches genau dann wahr ist, wenn **A kleiner als B ist** (**A** $<$ **B**). Dadurch dürfen mathematische größer Beziehungen als Fakten angenommen werden (z.B. smaller(0, s(0)).).

Zur Vereinfachung werden nur **aufsteigend sortierte** Listen betrachtet. Außerdem ist die leere Liste **null** auch immer im Komplement enthalten.

1. Definieren Sie ein Prädikat **complement(X, Y, Z)** in **LP**, welches genau dann wahr ist, wenn **Z** das **Komplement** der Listen **X** und **Y** ist. Das heißt, dass in der Liste **Z** nur jene Elemente enthalten sind, welche nur in der Liste **X**, aber nicht in der Liste **Y** existieren.

   <u>**Basis**</u>
   complement(X, null, X).
   complement(null, Y, null).

2. Zeigen Sie mittels **Resolution**, dass folgende Anfrage zu **TRUE** evaluiert:
   := complement(build(s(0), build(s(s(0)), build(s(s(s(0))), null))),
   build(0, build(s(0), build(s(s(s(0))), null))),
   build(s(s(0)), null)).

Viel Erfolg!