

Aufgabe 3 (Unifikation):

(5 + 5 + 5 + 5 + 5 = 25 Punkte)

In dieser Aufgabe sollen allgemeinste Unifikatoren bestimmt werden. Sie sollten diese Aufgabe ohne Hilfe eines Rechners lösen, da Sie zur Lösung von Aufgaben dieses Typs in der Klausur keinen Rechner zur Verfügung haben.

Nutzen Sie den Algorithmus zur Berechnung des allgemeinsten Unifikators (MGU) aus der Vorlesung, um die folgenden Termpaare auf Unifizierbarkeit zu testen.

Geben Sie neben dem Endergebnis σ auch die Unifikatoren $\sigma_1, \sigma_2, \dots, \sigma_n$ für die direkten Teilterme der beiden Terme an. Sollte ein σ_i nicht existieren, so begründen Sie kurz, warum die Unifikation fehlschlägt. Geben Sie in diesem Fall an, ob es sich um einen *clash failure* oder einen *occur failure* handelt.

- (i) $f(g(a, X), Y)$ und $f(g(a, h(Y)), c)$
- (ii) $f(g(Z, h(X)), X, g(Y, h(Y)))$ und $f(g(h(Y), h(a)), X, g(h(X), h(a)))$
- (iii) $f(X, Y, Z)$ und $f(h(h(Y)), g(Z), X)$
- (iv) $f(g(a), Z, h(X, X))$ und $f(g(Y), b, h(Y, b))$
- (v) $f(g(X, Y), Y, h(Z, Z))$ und $f(g(g(Y), Y), g(Z), h(a, Z))$

Aufgabe 5 (Beweisbäume):

(48 + 4 = 52 Punkte)

Betrachten Sie die Anfrage `?- a(X,s(0)).` zu folgendem Prolog-Programm:

```
a(s(s(X)),Y) :- a(X,s(Y)).
a(0,s(X)) :- a(0,X).
a(0,s(s(0))).
```

- a) Geben Sie den zugehörigen Beweisbaum (SLD-Baum) bis einschließlich Höhe 4 an (für die Definition der Höhe siehe Aufgabe 4). Markieren Sie unendliche Pfade mit ∞ und Fehlschläge mit (*fail*). Anstelle von $s(s(s(0)))$ können Sie auch $s^3(0)$ schreiben. Geben Sie alle im Teilbaum vorhandenen Lösungen (Antwortsubstitutionen) zur obigen Anfrage an. Wie viele Lösungen gibt es im gesamten Beweisbaum?
- b) Strukturieren Sie das gegebene Programm so in ein logisch äquivalentes Programm um, dass Prolog mit seiner Auswertungsstrategie **alle** Lösungen zur gegebenen Anfrage findet. Der Beweisbaum (SLD-Baum) muss nicht endlich sein! Sie brauchen den SLD Baum nicht angeben. Bei dieser Umstrukturierung dürfen Sie nur die Reihenfolge der Prolog-Klauseln verändern.

Aufgabe 7 (Arithmetik mit Prolog):

(23 Punkte)

Formulieren Sie ein Prolog-Programm mit einem Prädikat `multiples`, so dass `multiples(X, N, R)` genau dann wahr ist, wenn $X \geq 1$ und $N \geq 1$ gelten und R die absteigende Liste der positiven Vielfachen $[X * N, X * (N - 1), \dots, X]$ ist. Beispielsweise soll `multiples(2, 5, [10, 8, 6, 4, 2])` wahr sein. Verwenden Sie dafür keine anderen vordefinierten Prädikate als `is/2`, `</2` und `>/2` sowie die Addition.