

Prof. Dr. Jürgen Giesl
Stephan Swiderski

Präsenzübung
Logikprogrammierung
18. 07. 2008

Vorname: _____

Nachname: _____

Matrikelnummer: _____

Studiengang (bitte ankreuzen):

- Informatik Bachelor Informatik Diplom Informatik Lehramt
- Mathematik Bachelor
- Sonstige: _____

- Schreiben Sie bitte auf jedes Blatt **Vorname**, **Nachname** und **Matrikelnummer**.
- Geben Sie Ihre Antworten bitte in lesbarer und verständlicher Form an. Schreiben Sie bitte nicht mit roten Stiften oder mit Bleistiften.
- Bitte beantworten Sie die Aufgaben auf den **Aufgabenblättern**. Benutzen Sie ggf. auch die Rückseiten der **zur jeweiligen Aufgabe gehörenden** Aufgabenblätter.
- Antworten auf anderen Blättern können nur berücksichtigt werden, wenn **Name, Matrikelnummer und Aufgabennummer** deutlich darauf erkennbar sind.
- Was nicht bewertet werden soll, kennzeichnen Sie bitte durch **Durchstreichen**.
- Werden Täuschungsversuche beobachtet, so wird die Präsenzübung mit **0 Punkten** bewertet.
- Geben Sie bitte am Ende der Klausur **alle Blätter** zusammen mit den Aufgabenblättern ab.

	Anzahl Punkte	Erreichte Punkte
Aufgabe 1	4	
Aufgabe 2	7	
Aufgabe 3	8	
Aufgabe 4, Teil 1	2	
Aufgabe 4, Teil 2	4	
Summe	25	
Prozentzahl		

Vorname	Nachname	Matr.-Nr.

2

Aufgabe 1 (3+1 Punkte)

Gegeben sei folgendes Prolog-Programm, das hintereinanderstehende Duplikate aus Listen löscht:

```
erasePairs([], []).
erasePairs([X,X|XS],YS) :- !,erasePairs(XS,YS).
erasePairs([X|XS],[X|YS]) :- erasePairs(XS,YS).
```

a) Zeichnen Sie den SLD-Baum **ohne Beachtung des Cuts** zur Anfrage:

```
erasePairs([1,2,2],YS).
```

b) Markieren Sie die Teile Ihres SLD-Baums aus Aufgabe a), die der Cut löscht.

Vorname	Nachname	Matr.-Nr.

Aufgabe 2 (3+4 Punkte)

a) Implementieren Sie das 4-stellige Prädikat `replacePosition`.

Ein Anfrage der Form

$$\text{replacePosition}(t, s^i(0), [q_0, \dots, q_n], [r_0, \dots, r_n])$$

für $n \geq 0$ soll genau dann erfüllt sein, wenn $q_j = r_j$ für alle $j \neq i$ gilt und darüber hinaus $r_i = t$ gilt. Hierbei steht $s^i(0)$ für den Term $\underbrace{s(s(\dots s(0))\dots)}_{i\text{-mal}}$.

Das heißt, der Term im ersten Parameter wird an die i -te Position der ersten Liste geschrieben und das Ergebnis wird mit der zweiten Liste unifiziert. Das erste Element einer Liste hat den Index 0.

Ihr Programm sollte sich wie folgt verhalten:

```
?- replacePosition(d,0,[a,b,c],XS).
```

```
XS = [d,b,c]
```

```
?- replacePosition(d,s(0),[a,b,c,d,e],XS).
```

```
XS = [a,d,c,d,e]
```

```
?- replacePosition(d,s(s(s(0))),[a,b,c],XS).
```

```
XS = [a,b,c]
```

```
?- replacePosition(d,s(s(0)),[],XS).
```

```
XS = []
```

b) Sei \mathcal{P} Ihr Programm aus a). Geben Sie an, wie die Menge $\text{trans}_{\mathcal{P}}^n$ für ein beliebiges $n \in \mathbb{N}$ in Ihrem Beispiel aussieht.

Hierbei ist $\text{trans}_{\mathcal{P}}$ wie üblich die Funktion, die zur Berechnung der Fixpunktsemantik verwendet wird.

Vorname	Nachname	Matr.-Nr.

4

Aufgabe 3 (2+2+2+2 Punkte)

Sei $\varphi = p(\mathbf{b}, \mathbf{a}) \wedge \forall X, Y (p(X, Y) \rightarrow p(X, f(Y)))$ eine Formel über der Signatur (Σ, Δ) mit $\Sigma_0 = \{\mathbf{a}, \mathbf{b}\}$, $\Sigma_1 = \{f\}$ und $\Delta_2 = \{p\}$. Zusätzlich sei $\psi = \exists X p(X, f(f(\mathbf{a})))$.

- Überführen Sie φ und $\neg\psi$ in hierzu jeweils äquivalenten Klauselmengen $\mathcal{K}(\varphi)$ und $\mathcal{K}(\neg\psi)$.
- Um zu prüfen, ob $\varphi \models \psi$ gilt, leiten Sie die leere Klausel \square mit Hilfe von Input-Resolution aus der Klauselmenge $\mathcal{K}(\varphi) \cup \mathcal{K}(\neg\psi)$ her.
- Geben Sie ein Herbrandmodell S für die Formel φ an.
- Ist Ihr Modell S auch ein Modell der Formel $p(\mathbf{a}, \mathbf{b})$? Falls nein, existiert ein Herbrandmodell von φ , das auch Modell von $p(\mathbf{a}, \mathbf{b})$ ist?

Vorname	Nachname	Matr.-Nr.

5

Aufgabe 4, Teil 1 (1+1 Punkte)

Gegeben sei folgendes Prolog-Programm \mathcal{P} :

```
inc(A, [], []).
inc(A, [X|XS], [Y|YS]) :- Y =:= A+X, inc(A, [X|XS], [Y|YS]).
```

- a) Warum bricht die Anfrage `inc(5, [1,2,3,4,5], XS)` mit einem Programmfehler ab?
 b) Ändern Sie das Programm so ab, dass es alle Anfragen der Form

$$\text{inc}(a, [q_1, \dots, q_n], XS)$$

für $a, q_1, \dots, q_n \in \mathbb{N}$ mit der Antwortsstitution $XS/[r_1, \dots, r_n]$ beantwortet, so dass $r_i \in \mathbb{N}$ die Zahl $a+q_i$ ist.

Vorname	Nachname	Matr.-Nr.

Aufgabe 4, Teil 2 (3+1 Punkte) (nicht für Studiengang Bachelor)

Die folgenden Aufgabenteile c) und d) brauchen von Studierenden des *Studiengangs Bachelor* **nicht** bearbeitet werden.

- c) Implementieren Sie das Prädikat `apply`, welches ähnlich wie das Prädikat `inc` aus Programm \mathcal{P} funktioniert, aber im Gegensatz dazu eine beliebige Funktion auf Zahlen benutzt. Das heißt, eine Anfrage der Form

$$\text{apply}(\circ, a, [q_1, \dots, q_n], [r_1, \dots, r_n])$$

mit $a, q_i, r_i \in \mathbb{N}$ und $\circ \in \{+, -, *, /\}$ ist genau dann erfüllt, wenn jeweils r_i die Zahl $a \circ q_i$ ist. Benutzen Sie hierzu das Prädikat `=..` in geeigneter Weise.

- d) Wir ergänzen das Programm um die Direktive `:-use_module(library(clpfd))`. Ändern Sie die Implementierung von `apply` so ab, dass es bidirektional verwendet werden kann. Das heißt unter anderem, dass es sich wie folgt verhält:

```
?- apply(*,2,[1,Y],[X,6]).
X = 2, Y = 3
```

```
?- apply(+,3,[X,Y],[4,5]).
X = 1, Y = 2
```