

Minitest WS 2000/2001

Programmiertechniken in der Computerlinguistik I · 5. Februar 2001 · Dauer 45 Min

Name:

Vorname:

Adresse mit Ort:

1. Anfragen und Unifikation (30 Punkte)

Gib jeweils an, ob die folgenden Anfragen beweisbar sind oder nicht. Falls ja, gib die entsprechenden Variablenbelegungen an. Falls nein, gib den Grund an. (Je 5 Punkte)

- a) ?- blume(Kalla) = blume(rose, narzisse).
- b) ?- Blume = [STENGEL|[bluete]].
- c) ?- [a,b,c] = .(a,.(b,C)).
- d) ?- sprache(prolog) = L, L =.. [sprache,Prolog].
- e) ?- t(1,2,3) = t(A,B) ; T = t(A,B).
- f) ?- \+ r(C,D) = r(d,c).

2. Was machen diese Prädikate? (35 Punkte)

a) Schreibe in die unterste Zeile die Ausgabe, die beim Beweis der Anfrage aus der mittleren Zeile auf dem Bildschirm entsteht. Verwende das Zeichen ¶, um zu verdeutlichen, wo ein Zeilenende steht. (Je 15 Punkte)

Definition	<pre>p1([]). p1([X Xs]) :- write(X), nl, p1(Xs).</pre>	<pre>p2([]). p2([X Xs]) :- p2(Xs), write(X), nl.</pre>
Anfrage	?- p1(['Ein', 'Satz', '.']).	?- p2(['Ein', 'Satz', '.']).
Ausgabe		

b) Welche Art von Rekursion liegt in p1/1 bzw. p2/1 vor? (5 Punkte)

3. Backtracking (10 Punkte)

Was ist Backtracking?

4. Parsingstrategie (10 Punkte)

Warum parst der Standard-DCG-Parser von Prolog top-down und von links nach rechts? (Kurzes konzeptuelles Statement, keine technischen Details bzgl. Implementation von DCGs.)

5. Wortarten bestimmen (35 Punkte)

Gegeben sei ein Wortartenlexikon der Formlex (*Wort, Wortart*):

```
lex(der, art).  
lex('Hund', n).  
lex('trinkt', v).  
etc. etc.
```

Definiere das Prädikat `wortart/2`, das eine Liste von Wörtern nimmt im 1. Argument und daraus im 2. Argument die entsprechende Liste von Wortarten berechnet.

```
?- wortart([der, 'Hund', trinkt], WAs).  
WAs = [art, n, v]  
yes
```

Falls ein Wort nicht im Lexikon vorkommt – was wegen der Sprachkreativität immer wieder vorkommen kann – soll die Wortart mit '?' markiert werden.

```
?- wortart([der, 'HAL-2001-X34', trinkt], WA).  
WA = [art, '?', v]  
yes
```

Total: 120 Punkte

Erreicht:

Punkte