

## Übungen zur Vorlesung *Logikprogrammierung*

### Aufgabe 1 Lexikalische Analyse (2 Punkte)

Definieren Sie ein Prädikat `lexan(String,TokenList)`, welches einen String, der Zeichen der Aussagenlogik enthält, in eine Liste von Tokens übersetzt. Dabei sollen die Zeichen gemäß folgender Tabelle übersetzt werden:

A	→	aT
B	→	bT
C	→	cT
D	→	dT
E	→	eT
F	→	fT
(	→	klaT
)	→	klzT
-	→	negT
&	→	konT
v	→	disT
->	→	impT
<->	→	gdwT

Leerzeichen im Eingabestring sollen bei der lexikalischen Analyse ignoriert werden. Es sollen sich also z.B. folgende Dialoge ergeben:

```
?- lexan("A -> (AvB)", L).  
L = [aT, impT, klaT, aT, disT, bT, klzT]  
Yes  
?- lexan(")A -> A(vB", L).  
L = [klzT, aT, impT, aT, klaT, disT, bT]  
Yes  
?- lexan("A <- (AvB)", L).  
No
```

### Aufgabe 2 Differenzlisten (1 Punkt)

Betrachten Sie die beiden folgenden Fakten:

$$a(L - [A|Z], A, L - Z).$$

$$c(L - Z1, Z1 - Z2, L - Z2).$$

Was wird bei den folgenden beiden Zielen in L berechnet?

$$(a) \text{ ?- } a([a_1, a_2, \dots, a_n | X] - X, b, L).$$

$$(b) \text{ ?- } c([a_1, a_2, \dots, a_m | X] - X, [b_1, b_2, \dots, b_n | Y] - Y, L).$$

### Aufgabe 3 Quick-Sort (3 Punkte)

Eine Liste  $L$  von natürlichen Zahlen soll nach der folgenden Methode sortiert werden:

„Man nimmt das erste Element  $X$  von  $L$  und spaltet den Rest von  $L$  in die Liste *Small*, der Elemente  $\leq X$ , und in die Liste *Big*, der Elemente  $> X$ , auf. Dann werden *Small* und *Big* sortiert und mit  $X$  geeignet zusammengefügt, so daß eine geordnete Liste entsteht.“

Programmieren Sie Quick-Sort mit Hilfe von Differenzlisten.

*Hinweis:* Schreiben Sie ein Prädikat  $q$ , so daß  $q([a_1, a_2, \dots, a_n], L)$  in  $L$  die geordnete Differenzliste  $[a_{\pi(1)}, a_{\pi(2)}, \dots, a_{\pi(n)} | Z] - Z$  berechnet. Dann definiert man:

$$\text{quicksort}(L, S) \text{ :- } q(L, S - []).$$