

4. Praktische Übung zur

Logische und funktionale Programmierung

GRUPPENÜBUNGEN:

(G 16)

- a) Definieren Sie Operatoren für die Negation, Konjunktion und Disjunktion. Die Präzedenz entspricht genau dieser Reihenfolge.
- b) Schreiben Sie ein Prädikat, welches überprüft, ob eine aussagenlogische Formel syntaktisch korrekt aufgeschrieben worden ist. Die propositionalen Konstanten sind als Mitglieder einer Liste aufzufassen.
- c) Schreiben Sie ein Prädikat, welches die Anzahl der Negationen aus einer Formel aufzählt.
- d) Schreiben Sie ein Prädikat, welches überprüft, ob eine Formel Teilformel einer anderen Formel ist.
- e) Ergänzen Sie das Programm um eine Klausel `anzahl_buchstabe(F, N)`, die die Anzahl der Buchstaben N in der Formel liefert.
- f) Dasselbe nun mit `anzahl_und(F, N)`, ein Prädikat, welches die Anzahl N der \wedge -Zeichen in der Formel F ermittelt.

(G 17)

- a) Gesucht ist ein Prädikat `laenge(L, N)`, welches die Länge N einer Liste L berechnet.
- b) Gesucht ist ein Prädikat `zaehle(L, E, N)`, welches die Anzahl N der Vorkommen des Elementes E in einer Liste L bestimmt. c) Gesucht ist ein Prädikat `gleichlang(L1, L2)`, das genau dann erfüllt ist, wenn $L1$ und $L2$ Listen gleicher Länge sind.

(G 18)

Überlegen Sie sich wie ein Prädikat `klassifiziere/1`, definiert werden könnte, welches erfüllt ist, falls das Argument

- keine Zahl ist
- kein Integer ist
- ein gerader Integer-Wert ist
- eine Liste ist. Dabei soll mit dem Systemprädikat `write/1` jeweils ausgegeben werden, welcher Fall vorliegt.

(G 19)

Es soll das größte Element einer Liste von Zahlen zurückgegeben werden. Definieren Sie dazu das Prädikat `groesstest(L, X)`.