

# Theorie der Programmiersprachen

## 14. Übung

**1. Aufgabe:** In der Vorlesung wurde das Prädikat  $\text{insert}(X, L_1, L_2)$  definiert, das wahr ist, falls sich die Liste  $L_2$  durch das Einfügen des Elementes  $X$  an eine beliebige Position in der Liste  $L_1$  erzeugen lässt. Die Implementation ist gegeben durch

$$\begin{aligned} &\text{insert}(X, L_1, [X|L_1]), \\ &\text{insert}(X, [H|L_1], [H|L_2]) :- \text{insert}(X, L_1, L_2). \end{aligned}$$

Finden Sie eine Definition von  $\text{insert}$  die nur eine Klausel benötigt. Dies ist zum Beispiel mit Hilfe der Funktion  $\text{append}$  möglich.

Führen beide Implementationen in Prolog zu einem Ergebnis? Wenn ja, welche der beiden Funktionen ist schneller?

**2. Aufgabe:** Sei  $\text{sort}(L_1, L_2)$  wahr, falls  $L_2$  durch Sortieren der Liste  $L_1$  entsteht. Eine mögliche Implementation dieses Prädikats ist gegeben durch

$$\begin{aligned} &\text{sort}(L_1, L_2) :- \text{permut}(L_1, L_2), \text{ord}(L_2) \\ &\text{ord}([]) \\ &\text{ord}([X]) \\ &\text{ord}([X|[Y|Z]]) :- X \leq Y, \text{ord}([Y|Z]). \end{aligned}$$

Finden Sie zwei weitere Implementationen und vergleichen Sie die Geschwindigkeiten in Prolog.

### 3. Aufgabe:

Man kann Mengen als Listen sehen, die jedes Element höchstens einmal enthalten und deren Reihenfolge egal ist. Implementieren Sie das Prädikat  $\text{makeSet}(L_1, L_2)$ , das wahr ist, wenn  $L_1$  und  $L_2$  (bis auf ihre Anzahl) die gleichen Elemente enthalten und in  $L_2$  kein Element mehrfach vorkommt.

Sie dürfen verwenden, dass „ $A \setminus = B$ “ genau dann wahr ist, wenn  $A$  und  $B$  ungleich sind.