

6. Praktische Übung zur

Logische und funktionale Programmierung

GRUPPENÜBUNGEN:

(G 23)

Definieren Sie die Prädikate `zweites/2`, `drittes/2`, die das zweite bzw. dritte Element einer Liste ausgeben.

(G 24)

Definieren Sie ein Prädikat `letztes/2`, das den letzten Term in einer Liste angibt.

(G 25)

Definieren Sie eine Verkettung von n Listen zu einer Liste (Z. Beispiel: `verkette_n([[1],[2,3],[4,5,6]],L)`. `L= [1,2,3,4,5,6]; no.`).

Rufen Sie das Prädikat mit zwei Variablen X und Y sowie der Liste `L= [1,2,3,4,5,6]` als drittem Argument auf. Erzwingen Sie Backtracking und geben Sie alle möglichen Belegungen von X und Y an, die zu L verkettet werden können.

(G 26)

Definieren Sie das Prädikat `austausch/4`, das Term X gegen einen Term Y in einer Liste $L1$ austauscht und die Liste $L2$ als Resultat produziert. Zum Beispiel `austausch(a,b,[a,c,a,e,a,f],L)`. `L= [b,c,b,e,b,f]`.

(G 27) Sieb von Eratosthenes

Implementieren Sie ein Prädikat `primes(UpTo, ListOfPrimes)`, das eine obere Grenze `UpTo` bekommt und eine Liste der Primzahlen kleiner als die Grenze zurück gibt. Als Algorithmus bietet sich das Sieb des Eratosthenes an.