

Vorlesung Perlen der Informatik

Aufgabe 1 Prädikate auf Listen

Eine Liste, die aus den Anfangselementen x_1, \dots, x_n sowie der Restliste xs zusammengesetzt ist, wird in Prolog durch die Notation

$[x_1, \dots, x_n | xs]$

dargestellt. Schreiben Sie Prolog-Prädikate zum

- Zusammenhängen zweier Listen
- Umdrehen einer Liste
- Testen, ob ein Element in einer Liste enthalten ist

Aufgabe 2 Erfüllbarkeitsprüfer

Schreiben Sie ein Prolog-Prädikat `sat(f)`, das eine aussagenlogische Formel f auf Erfüllbarkeit testet. Formeln sollen hierbei durch die zweistelligen Konstruktoren `and` und `or`, sowie die einstelligen Konstruktoren `not` und `var` dargestellt werden, wobei das Argument des Konstruktors `var` eine Prolog-Variable ist, die im Verlauf der Berechnung mit einem Wahrheitswert, d.h. `true` oder `false` belegt werden soll. So wird z.B. die Formel $P \vee (Q \wedge R)$ durch den Prolog-Term `or(var(P), and(var(Q), var(R)))` dargestellt. Das Prädikat `sat` sollte verschränkt rekursiv zusammen mit einem ebenfalls einstelligen Prädikat `unsat` definiert werden, das eine Formel auf Unerfüllbarkeit testet.

Aufgabe 3 Zahldarstellung

Eine Möglichkeit, natürliche Zahlen in Prolog darzustellen besteht darin, einen nullstelligen Konstruktor `zero` sowie einen einstelligen Konstruktor `suc` zu verwenden. Damit kann z.B. die Zahl 4 durch den Term `suc(suc(suc(suc(zero))))` dargestellt werden.

Lösen Sie nun unter Verwendung der obigen Darstellung für natürliche Zahlen folgende Aufgaben:

- Schreiben Sie ein Prädikat `le(X, Y)`, das genau dann erfüllt ist, wenn die natürliche Zahl X kleiner oder gleich Y ist.
- Schreiben Sie ein Prädikat `gt(X, Y)`, das genau dann erfüllt ist, wenn die natürliche Zahl X größer als Y ist.

Aufgabe 4 Permutation Sort

Für die Implementierung des folgenden Sortierverfahrens sollen Listen von natürlichen Zahlen in der obigen Darstellung verwendet werden.

- Schreiben Sie ein Prädikat `permute(Liste1, Liste2)`, das genau dann erfüllt ist, wenn `Liste2` eine Permutation von `Liste1` ist.
- Schreiben Sie ein Prädikat `sorted(Liste)`, das genau dann erfüllt ist, wenn `Liste` sortiert ist.

- Schreiben Sie mit Hilfe der Prädikate `permute` und `sorted` ein Prädikat `psort(Liste, SortierteListe)`

Was können Sie über die Effizienz dieses Sortierverfahrens sagen?

Aufgabe 5 Suche im Labyrinth

Es soll ein Weg im folgenden Labyrinth von einem Raum zu einem anderen gefunden werden, wobei jeder Raum nur einmal betreten werden darf.

a	b	c	d	e	
	f	g	h	i	
	j	k	l	m	
	n	o	p	q	r

Definieren Sie ein Prolog-Prädikat

`maze(Raum1, Raum2, Weg)`

das den Weg von *Raum₁* nach *Raum₂* berechnet und die Lösung *Weg* als Liste von Räumen liefert. Stellen Sie das Labyrinth von Aufgabe 1 durch mehrere Fakten dar. Die Existenz einer Tür zwischen zwei Räumen soll hierbei durch ein Faktum `door(raum1, raum2)` beschrieben werden (z.B. `door(a, b)`).

Beispiel:

?- maze(a,e,W).

W = [a, b, c, g, k, j, n, o, p, l, h, d, e]