ARBEITSBLATT ZU LISTEN

Aufgabe 1: Wie antwortet Prolog auf die folgenden Anfragen?

1. `[a,b,c,d] = [a,[b,c,d]].`
2. `[a,b,c,d] = [a|[b,c,d]].`
3. `[a,b,c,d] = [a,b,[c,d]].`
4. `[a,b,c,d] = [a,b|[c,d]].`
5. `[a,b,c,d] = [a,b,c,[d]].`
6. `[a,b,c,d] = [a,b,c|[d]].`
7. `[a,b,c,d] = [a,b,c,d,[]].`
8. `[a,b,c,d] = [a,b,c,d|[]].`

Aufgabe 2: Angenommen wir haben die folgende Wissensbasis:

```
tran(eins,one).
tran(zwei,two).
tran(drei,three).
tran(vier,four).
tran(fuenf,five).
tran(sechs,six).
tran(sieben,seven).
tran(acht,eight).
tran(neun,nine).
```

Schreibe ein Prädikat `listtran(G,E)`, dass eine Liste aus deutschen Zahlwörtern in eine Liste aus englischen Zahlwörtern übersetzt.

Beispiel: `listtran([eins,neun,zwei],X).` sollte `X = [one,nine,two].` ergeben.
Das Program sollte auch in die andere Richtung funktionieren. Auf die Anfrage `listtran(X,[one,seven,six,two]).` sollte es `X = [eins,sieben,sechs,zwei].` antworten.

Tipp: Um diese Aufgabe zu lösen, überlege zuerst, wie die leere Liste übersetzt werden muss. Das ist der Basisfall. Bei nicht-leeren Listen, übersetze zuerst den Kopf (Head) der Liste und benutze dann Rekursion, um den Rest (Tail) der Liste zu übersetzen.

Aufgabe 3: Schreibe ein Prädikat twice/2 [diese Notation bedeutet, dass twice zwei Variablen enthält], dessen erstes Argument eine Liste ist und dessen zweites Argument eine Liste ist, die alle Elemente der ersten Liste zweimal hintereinander enthält. Die Anfrage `twice([a,4,auto],X).` sollte zum Beispiel `X = [a,a,4,4,auto,auto]).` ergeben. Und die Anfrage `twice([1,2,1,1],X).` sollte `X = [1,1,2,2,1,1,1,1].` ergeben.

Tipp: Um diese Aufgabe zu lösen, überlege zuerst wieder, was passieren soll, wenn das erste Argument die leere Liste ist. Das ist der Basisfall. Für nicht-leere Listen, überlege, was mit dem Kopf (Head) der Liste passieren sollte, und benutze Rekursion, um den Rest (Tail) zu behandeln.

Aufgabe 4: Schreibe ein 3-stelliges Prädikat combine/3, das drei Listen als Argumente nimmt und die Elemente der ersten zwei Listen folgendermaßen in der dritten Liste kombiniert: Die Anfrage `combine1([a,b,c],[1,2,3],X).` sollte als Lösung `X = [a,1,b,2,c,3].` ergeben.

Aufgabe 5: Veranschauliche den Backtracking-Vorgang für folgende Anfragen, indem du den zugehörigen Suchbaum aufzeichnest.

```
?- element(a,[c,b,a,y]).
?- element(x,[a,b,c]).
?- element(X,[a,b,c]).
```