

Übungsaufgaben

Logik in der Praxis - Logikprogrammierung (Prolog)

Abgabe bis 28.10.08

Exercise 4.1 (2 Punkte)

How does Prolog respond to the following queries?

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. $[a,b,c,d] = [a,b,c,d|[]]$.
9. $[] = []$.
10. $[] = [_]$.
11. $[] = [_|[]]$.

Exercise 4.2 (2 Punkte)

Suppose we are given a knowledge base with the following facts:

```
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).
```

Write a predicate `listtran(G,E)` which translates a list of German number words to the corresponding list of English number words. For example:

```
listtran([eins,neun,zwei],X).
```

should give:

```
X = [one,nine,two].
```

Your program should also work in the other direction. For example, if you give it the query `listtran(X,[one,seven,six,two])`.

it should return:

```
X = [eins,sieben,sechs,zwei].
```

Hint: to answer this question, first ask yourself 'How do I translate the empty list of number words?'. That's the base case. For non-empty lists, first translate the head of the list, then use recursion to translate the tail.

Exercise 4.3 (3 Punkte)

Write a predicate `twice(In,Out)` whose left argument is a list, and whose right argument is a list consisting of every element in the left list written twice. For example, the query `twice([a,4,buggle],X)`.

should return

`X = [a,a,4,4,buggle,buggle]`.

And the query

`twice([1,2,1,1],X)`.

should return

`X = [1,1,2,2,1,1,1,1]`.

Hint: to answer this question, first ask yourself 'What should happen when the first argument is the empty list?'. That's the base case. For non-empty lists, think about what you should do with the head, and use recursion to handle the tail.

Exercise 4.4 (3 Punkte)

Draw the search trees for the following three queries:

?- `member(a,[c,b,a,y])`.

?- `member(x,[a,b,c])`.

?- `member(X,[a,b,c])`.