

## Exercise Session Informatik III

### 6. Love Match in Prolog

#### append/3

```
% Fügt eine Liste an eine andere an
append([], A, A).
append([A|B], C, [A|D]) :-
    append(B, C, D).

| ?- append([1,2,3], [4,5], R).
R = [1,2,3,4,5] ?
```



#### remove/3

```
% Entfernt ein Element aus einer Liste
remove(A, [A|B], B).
remove(A, [B|C], D) :-
    remove(A, C, E), append([B], E, D).
remove(_, [], []).

| ?- remove(1, [1,2,3], R).
R = [2,3] ?
```



#### delete/3

```
% Löscht alle Elemente der ersten Liste in der zweiten
delete([], A, A).
delete([A|B], C, D) :-
    delete(B, C, E), remove(A, E, D).

| ?- delete([1,2,3], [1,2,3,4,5], R).
R = [4,5] ?
```



#### resultant/3

```
% Bildet den Resultat String
resultant(A, B, C) :-
    delete(A, B, D), delete(B, A, E), append(D, E, C).

| ?- resultant("michael", "moira", R).
R = [111,114,99,104,101,108] ?
% [orchel]
```



#### sequence/3

```
% Iteriert über den Strings und findet das Resultat
sequence([], A, A).
sequence([_|A], love, B) :- sequence(A, hate, B).
sequence([_|A], hate, B) :- sequence(A, friendship, B).
sequence([_|A], friendship, B) :-
    sequence(A, marriage, B).
sequence([_|A], marriage, B) :- sequence(A, love, B).
```



## game/3

```
game(A, B, C) :-  
    resultant(A, B, D),  
    sequence(D, mariage, C).  
  
| ?- game("michael", "moira", R).  
R = hate ?  
yes
```



That's all folks!



Viel Spass!  
Bis nächste Woche!

