Rekursion in Prolog am Beispiel des Systemprädikats `append`

Datenbasis:

```
myAppend([],F,F).
myAppend([H|T],L,[H|R]):-myAppend(T,L,R).
```

Aufruf mit `trace.` in Prolog:

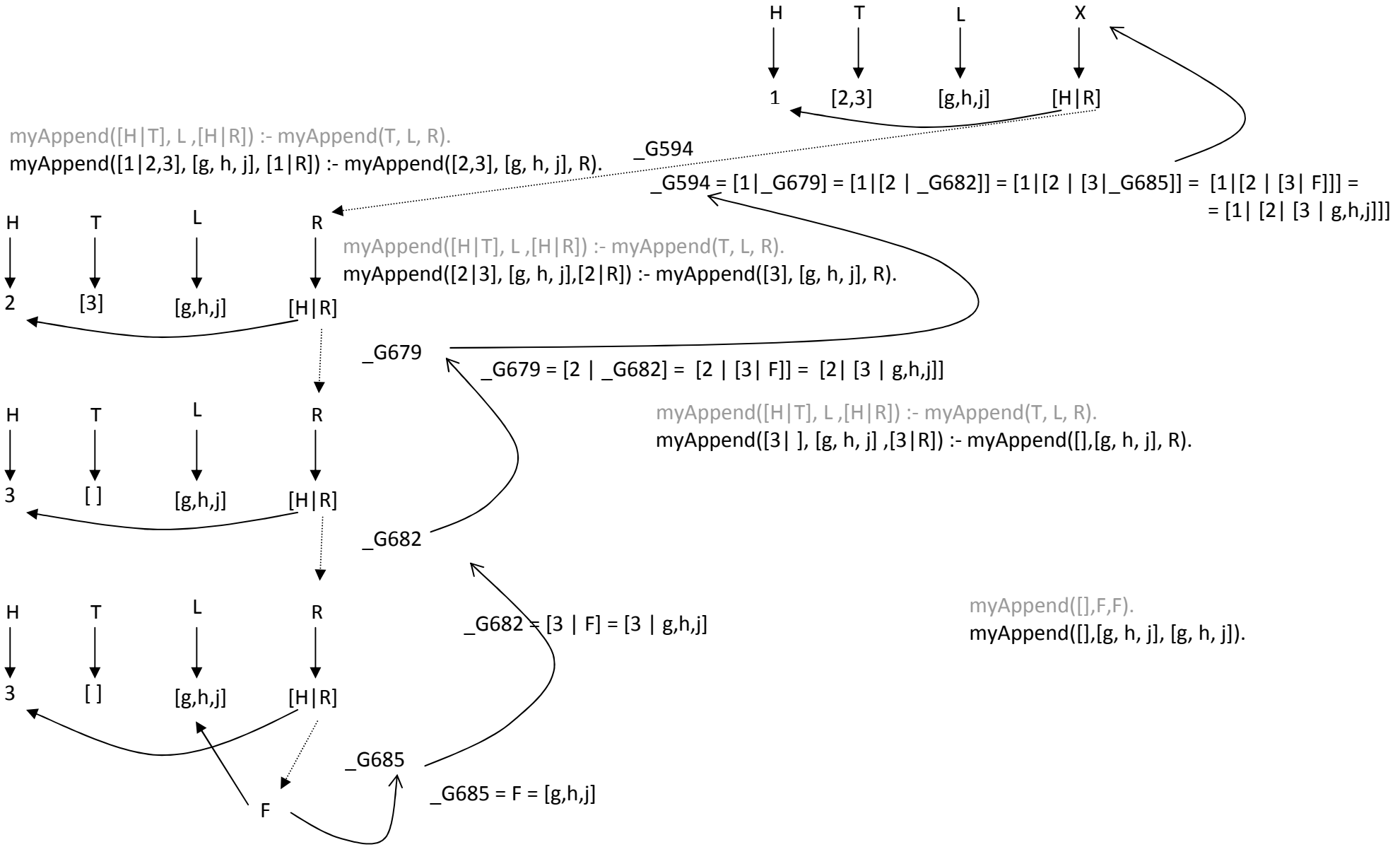
```
[trace] 2 ?- myAppend([1,2,3],[g,h,j],X).
   Call: (6) myAppend([1, 2, 3], [g, h, j], _G594) ? creep
   Call: (7) myAppend([2, 3], [g, h, j], _G679) ? creep
   Call: (8) myAppend([3], [g, h, j], _G682) ? creep
   Call: (9) myAppend([], [g, h, j], _G685) ? creep
   Exit: (9) myAppend([], [g, h, j], [g, h, j]) ? creep
   Exit: (8) myAppend([3], [g, h, j], [3, g, h, j]) ? creep
   Exit: (7) myAppend([2, 3], [g, h, j], [2, 3, g, h, j]) ? creep
   Exit: (6) myAppend([1, 2, 3], [g, h, j], [1, 2, 3, g, h, j]) ? creep

X = [1, 2, 3, g, h, j].
```

Marian Fietta

```
myAppend([],F,F).
myAppend([H|T],L,[H|R]) :- myAppend(T,L,R).

myAppend([1,2,3],[g,h,j],X).
```



Marian Fietta