List processing

Representing lists [a, b, c] (atoms)

Empty list []

Pattern [H|T] (H: head, T: tail)

length:

```
length([], 0).
length([H|T], N) :- length(T, N1), N is N1 + 1.
```

('assignment')

[H|T] matches [a] with H = a, T = []

[H|T] matches [a, b, c] with H = a, T = [b, c]

[H|T] does not match []

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

| $H_1 = a$
| $T_1 = [b,c]$
| $N_1 = X$

length([b,c], $N1_1$), X is $N1_1 + 1$

| $H_2 = b$
| $T_2 = [c]$
| $N_2 = N1_1$

length([c], $N1_2$), $N1_1$ is $N1_2 + 1$, X is $N1_1 + 1$

| $H_3 = c$
| $T_3 = []$
| $N_3 = N1_2$

length([], $N1_3$), $N1_2$ is $N1_3 + 1$, $N1_1$ is $N1_2 + 1$,
X is $N1_1 + 1$

| $N1_3 = 0$

$N1_2$ is $0 + 1$, $N1_1$ is $N1_2 + 1$, X is $N1_1 + 1$

$N1_1$ is $1 + 1$, X $N1_1 + 1$ — X is $2 + 1$ — $X = 3$

member:

```
member(X, [X|_]).
member(X, [_|T]) :- member(X, T).
```

```
?- member(b, [a,b,c])
   | X1 = b
   | T1 = [b,c]
member(b, [b,c])
   | X2 = b
  yes
```

```
?- member(d, [a,b])
   | X1 = d
   | T1 = [b]
member(d, [b])
   | X2 = d
   | T2 = []
member(d, [])
   |
  no
```

append([a,b], [c,d], X)

X = [a,b,c,d]

append([], L, L).
append([H|T1], L, [H|T2]) :- append(T1, L, T2).

?- append([a,b],[c,d],X)

- $H_1 = a$
- $T1_1 = [b]$
- $L_1 = [c,d]$
- $X = [a|T2_1]$

append([b], [c,d], $T2_1$)

- $H_2 = b$
- $T1_2 = []$
- $L_2 = [c,d]$
- $T2_1 = [b|T2_2]$

append([], [c,d], $T2_2$)

- $L_3 = [c,d]$
- $L_3 = T2_2$

X = [a,b,c,d]

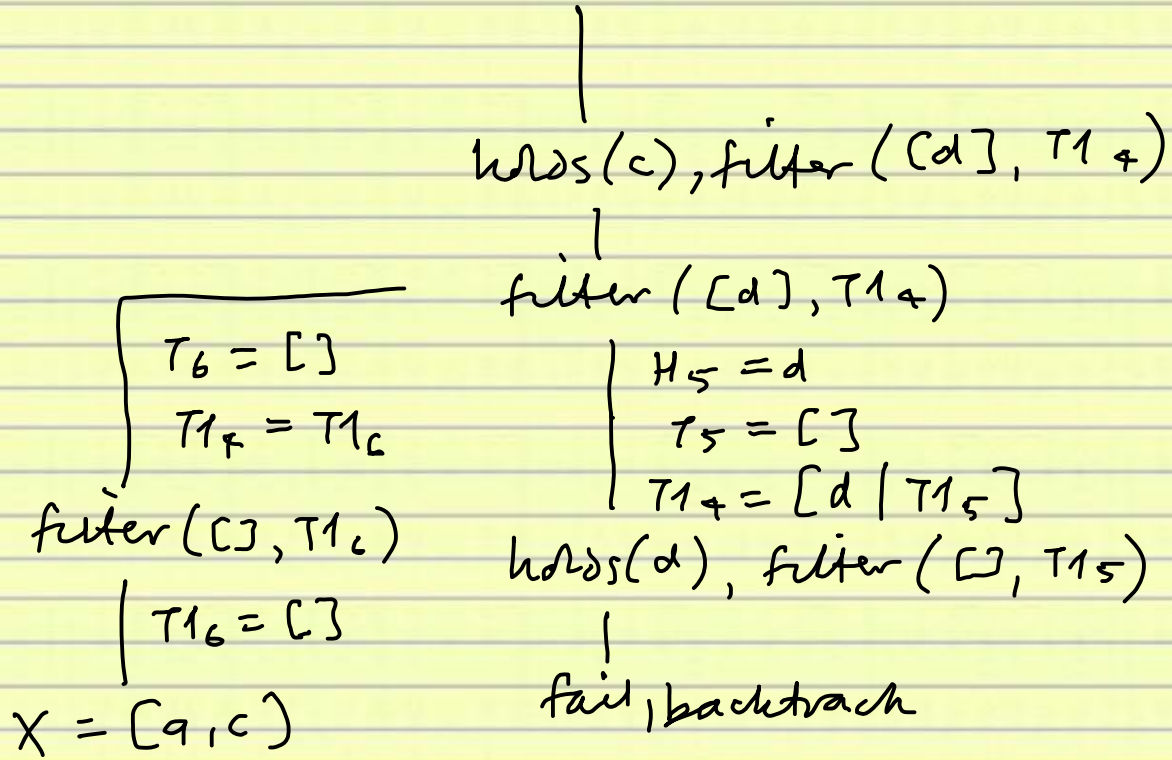
$X = [a|[b|[c,d]]]$

$= [a,b,c,d]$

append can work 'backwards'

```
?- append(X, [c,d], [a,b,c,d])
    | X = [H1|T1]
    | L1 = [c,d]
    | T2_1 = [b,c,d]
    | H1 = a
   append(T1_1, [c,d], [b,c,d])
       | T1_1 = [H2|T1_2]
       | L2 = [c,d]
       | T2_2 = [c,d]
       | H2 = b
      append(T1_2, [c,d], [c,d])
         | T1_2 = []
      X = [a,b]
```

```
?- append(X, Y, [a,b,c,d]).
    X = [], Y = [a,b,c,d] ;
    X = [a], Y = [b,c,d] ;
    X = [a,b], Y = [c,d] ;
    X = [a,b,c], Y = [d] ;
    X = [a,b,c,d], Y = [] ;
    no
```


holds(c), filter([d], T1₄)
filter([d], T1₄)
H₅ = d
T₅ = []
T1₄ = [d | T1₅]
holds(d), filter([], T1₅)
fail, backtrack
T₆ = []
T1₄ = T1₆
filter([], T1₆)
T1₆ = []
X = [a, c]