

?- suffix([a], L), prefix(L, [a, b, c]).

```

1   append([], Y, Y).
2   append([H|X], Y, [H|Z]) :- append(X, Y, Z).

3   prefix(P, L) :- append(P, _, L).
4   suffix(S, L) :- append(_, S, L).

```

(4) |  $S = [a]$

$\text{append}(-1, [a], L)$ ,  $\text{prefix}(L, [a, b, c])$ .

①  $-1 \rightarrow \{\}$   
 $L \rightarrow [a]$

②  $Y \rightarrow [a]$   
 $L \rightarrow [H|Z]$

$\text{prefix}([a], [a, b, c])$ .

③  $P \rightarrow [a]$   
 $L \rightarrow [a, b, c]$

$\text{append}([a], -2, [a, b, c])$ .

②  $H \rightarrow a$   
 $X \rightarrow \{\}$

②  $Z \rightarrow [b, c]$

$\text{append}(\{\}, -2, [b, c])$ .

①  $Y \rightarrow [b, c]$   
 $-2 \rightarrow \{\}$

$L = [a]$  ;

$\text{prefix}([H, a], [a, b, c])$ .

③ |

$\text{append}([H, a], -3, [a, b, c])$ .

② |  $X \rightarrow [a]$   
 $Z \rightarrow [b, c]$

② |  $H \rightarrow a$

$\text{append}([a], -3, [b, c])$ .

backtrack

$\text{append}(X, [a], Z)$ ,  $\text{prefix}([H|Z], [a, b, c])$ .

① |  $X \rightarrow \{\}$   
 $Z \rightarrow [a]$

② |  $X \rightarrow [H, X]$ ,  $Z \rightarrow [H, Z]$   
 $Y \rightarrow [a]$

infinite subtree with  
no solutions  $\Rightarrow$  infinite  
computation

$\text{append}(X_1, [a], Z_1)$ ,  $\text{prefix}([H|Z_1], [a, b, c])$ .

① |  $X_1 \rightarrow \{\}$   
 $Z_1 \rightarrow [a]$

② |

$\text{prefix}([H, H_1, a], [a, b, c])$ .

③ |

$\text{append}([H, H_1, a], -4, [a, b, c])$ .

② |  $H \rightarrow a$

$\text{append}([H_1, a], -4, [b, c])$ .

② |  $H_1 \rightarrow b$

$\text{append}([a], -4, [c])$ .

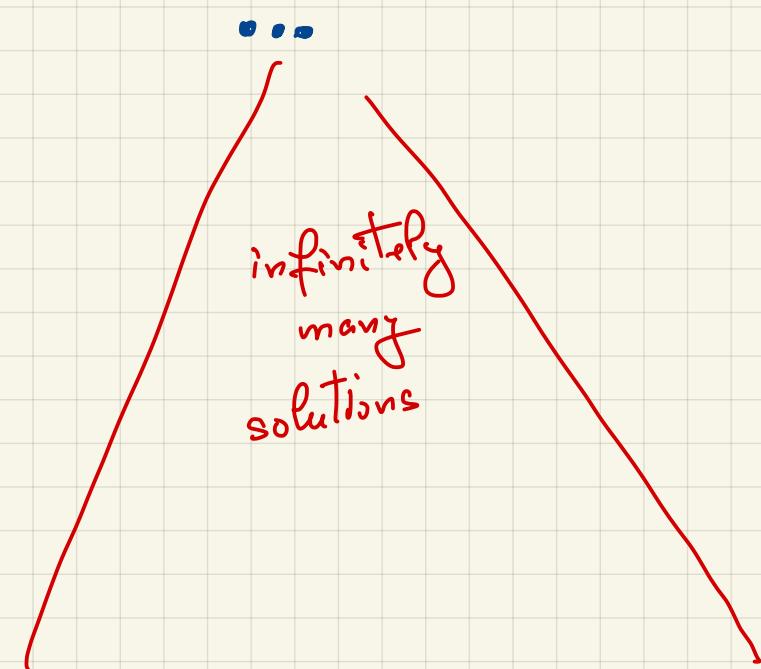
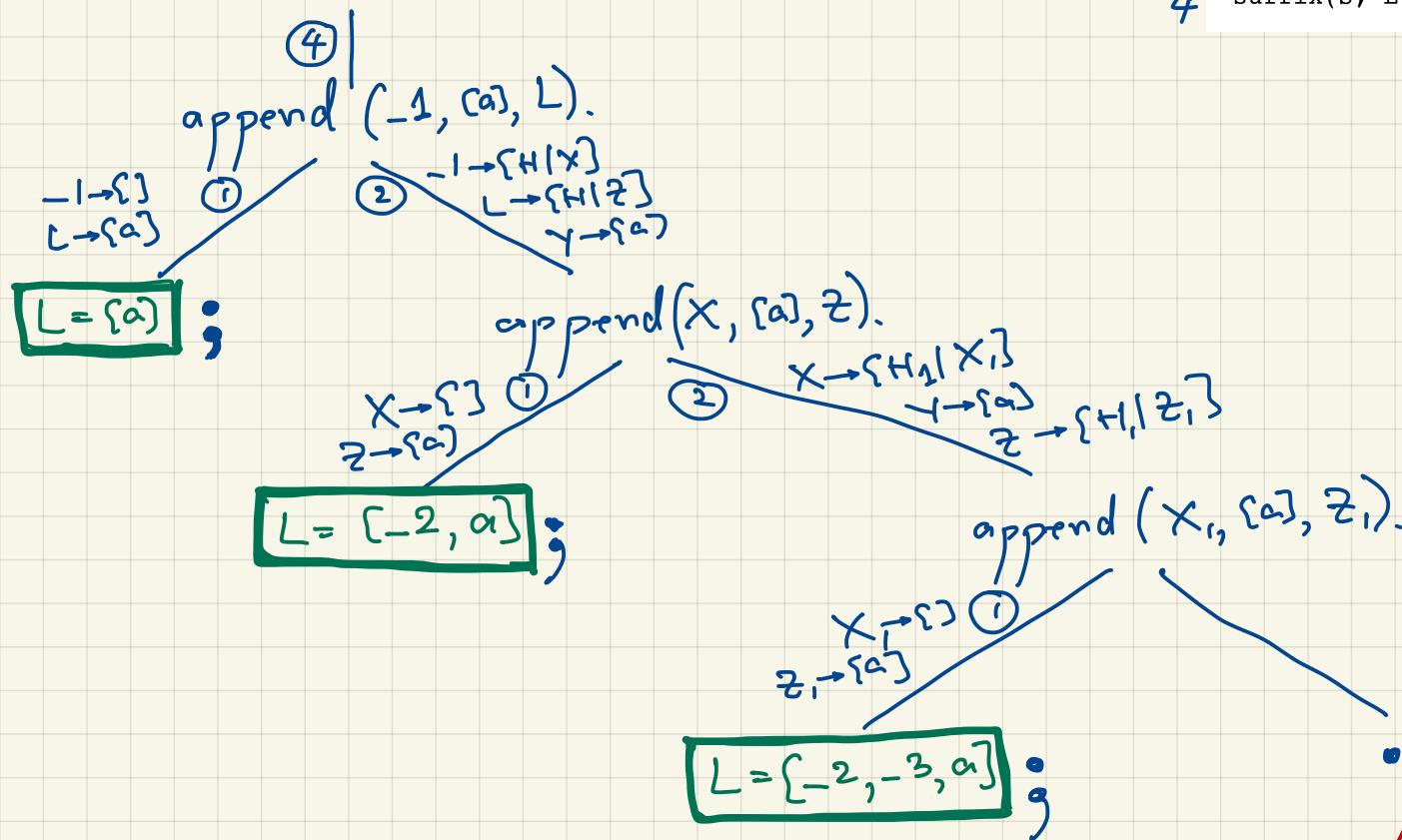
backtrack

backtrack

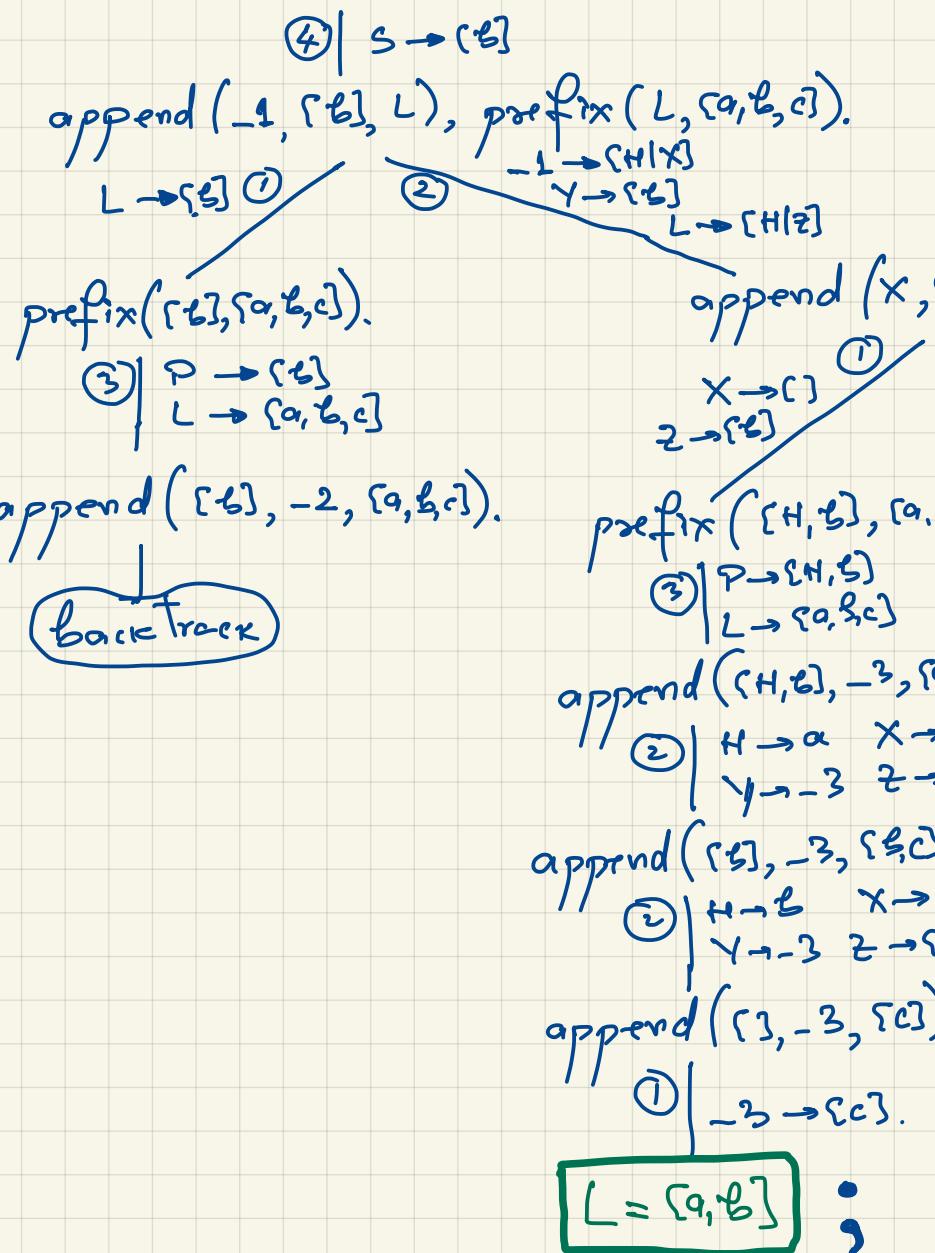
backtrack

```
?- suffix([a], L).
```

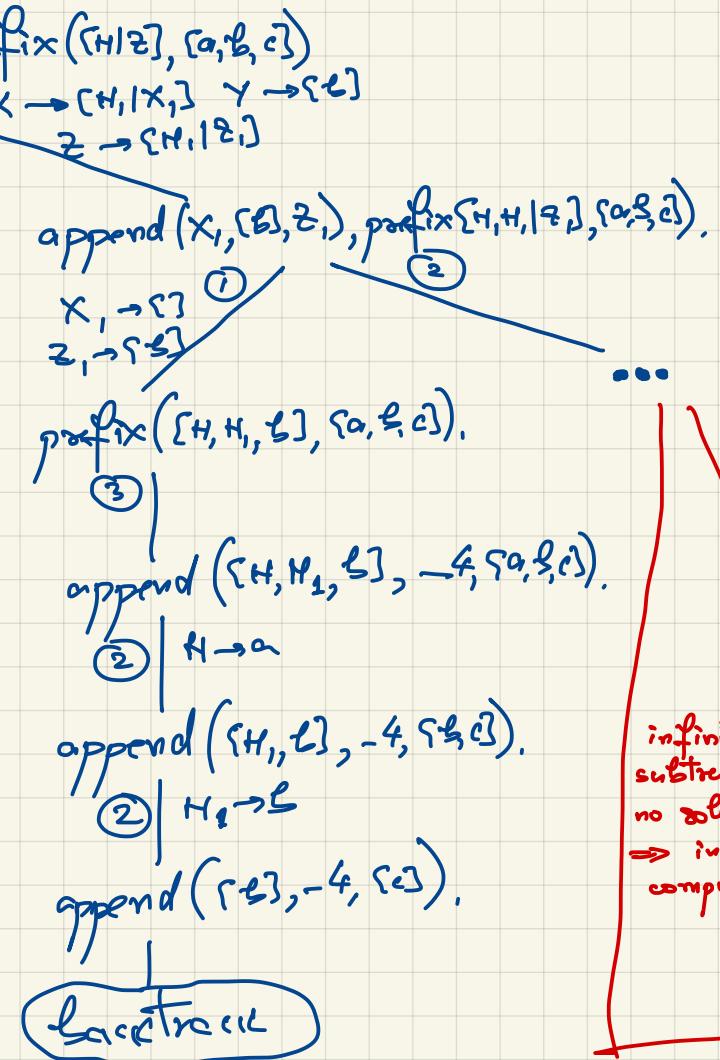
```
1 append([], Y, Y).  
2 append([H|X], Y, [H|Z]) :- append(X, Y, Z).  
3 prefix(P, L) :- append(P, _, L).  
4 suffix(S, L) :- append(_, S, L).
```



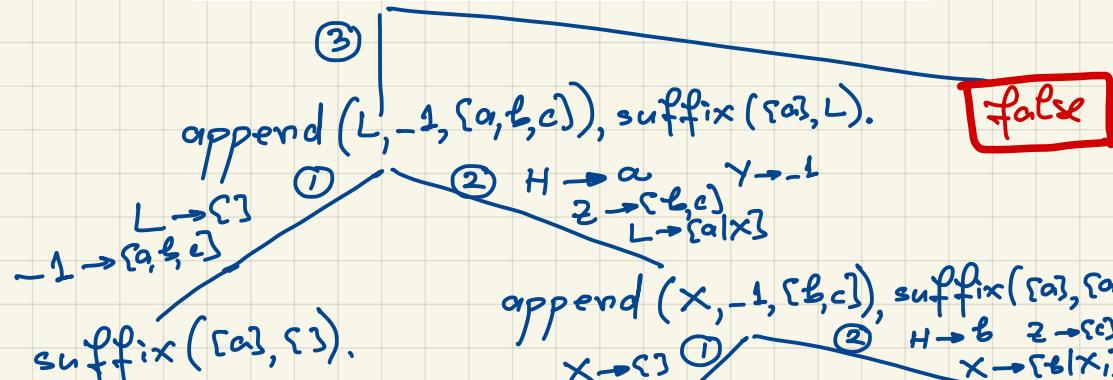
?- suffix([b], L), prefix(L, [a, b, c]).



1.  $\text{append}([], Y, Y).$
2.  $\text{append}([H|X], Y, [H|Z]) :- \text{append}(X, Y, Z).$
3.  $\text{prefix}(P, L) :- \text{append}(P, \_, L).$
4.  $\text{suffix}(S, L) :- \text{append}(\_, S, L).$

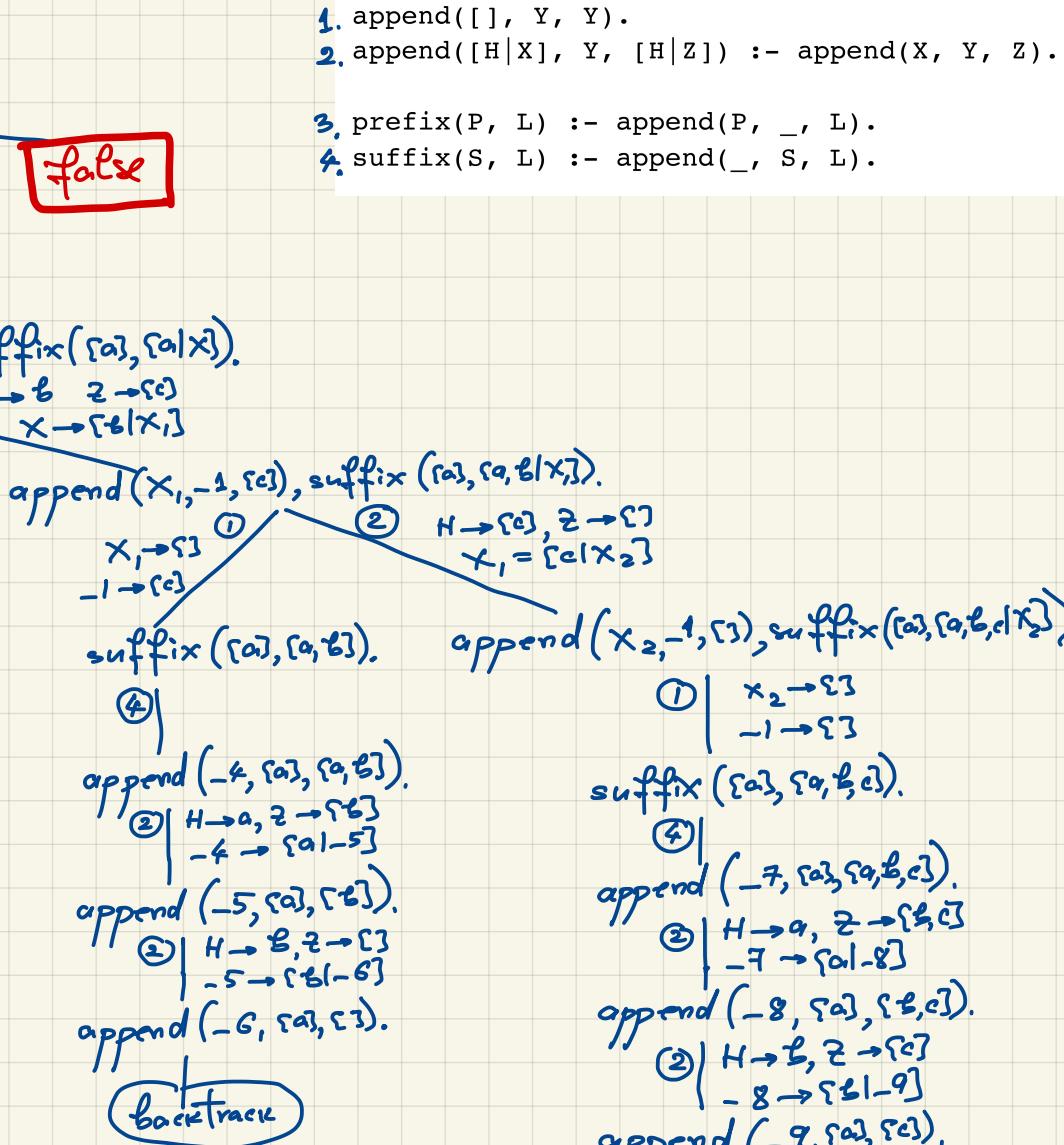
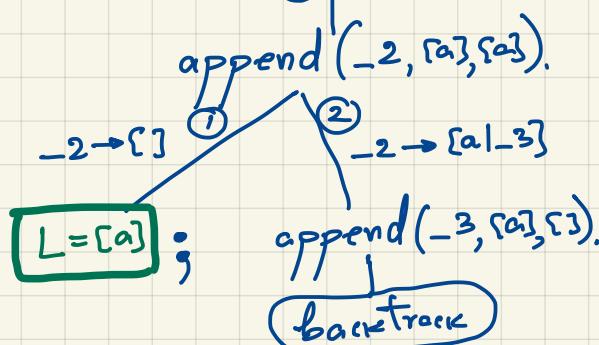


?- prefix(L, [a, b, c]), suffix([a], L).



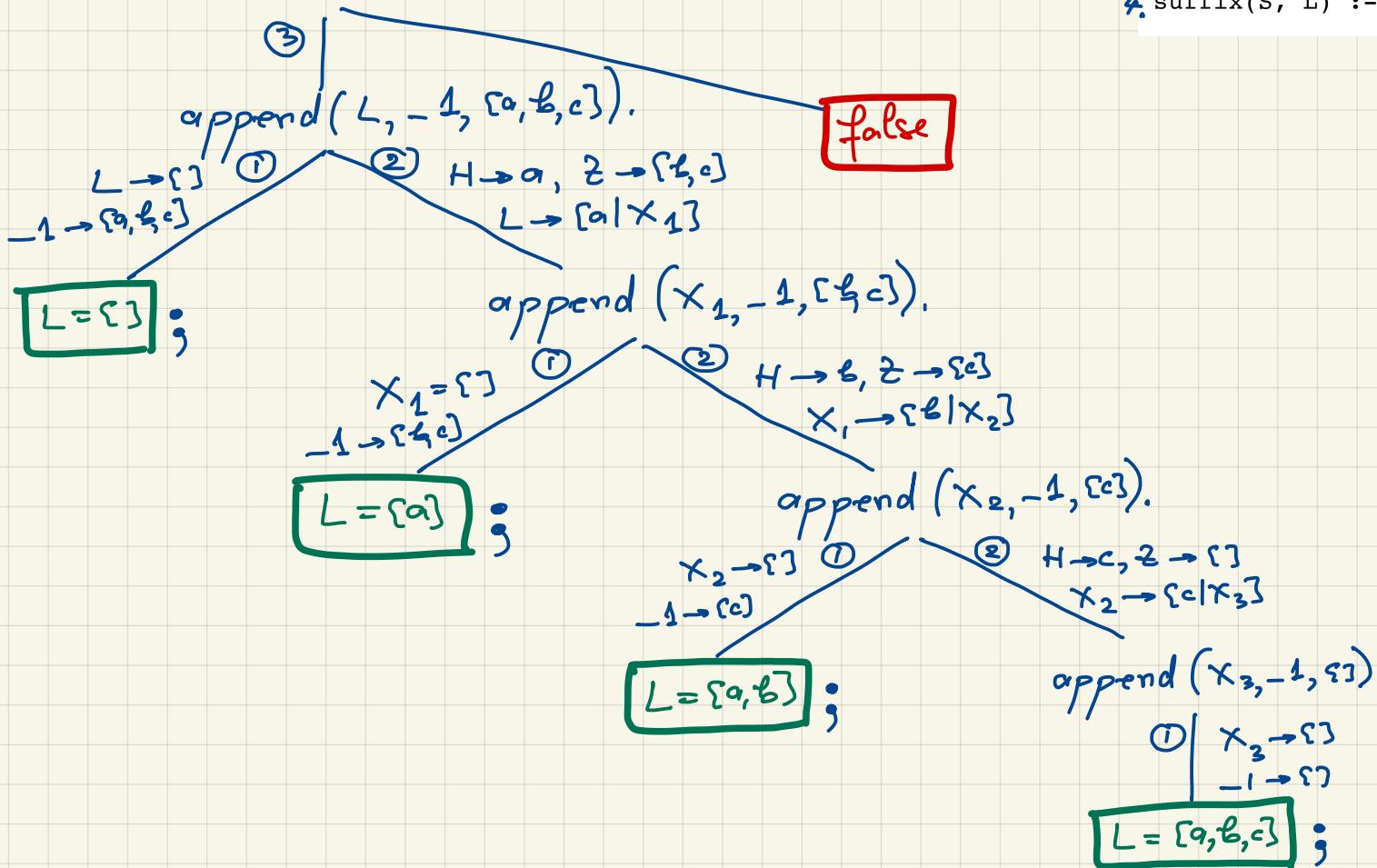
④ |  
append(-2, [a], [a]).

backtrack



?- prefix(L, [a, b, c]).

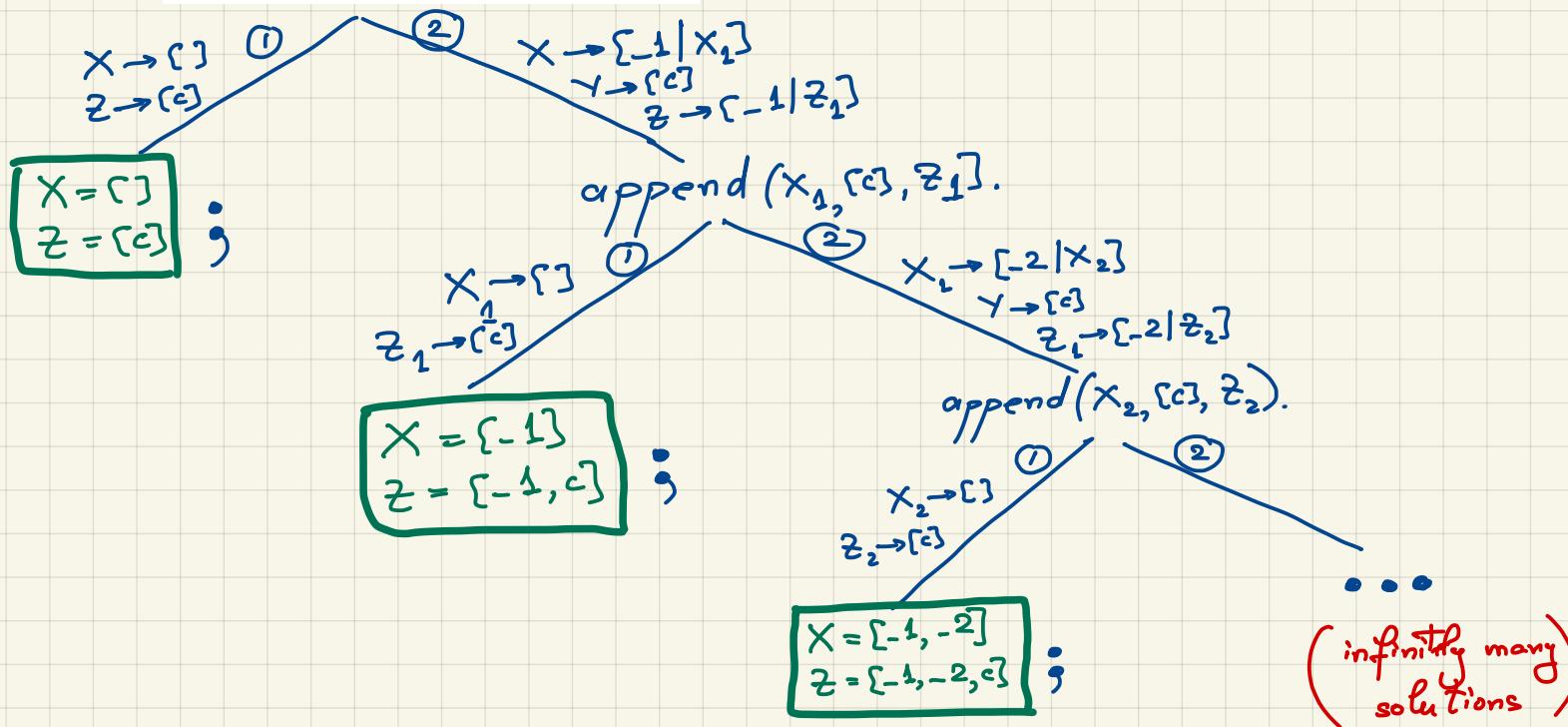
1. `append([], Y, Y).`
2. `append([H|X], Y, [H|Z]) :- append(X, Y, Z).`
3. `prefix(P, L) :- append(P, _, L).`
4. `suffix(S, L) :- append(_, S, L).`



1.  $\text{append}([], Y, Y).$

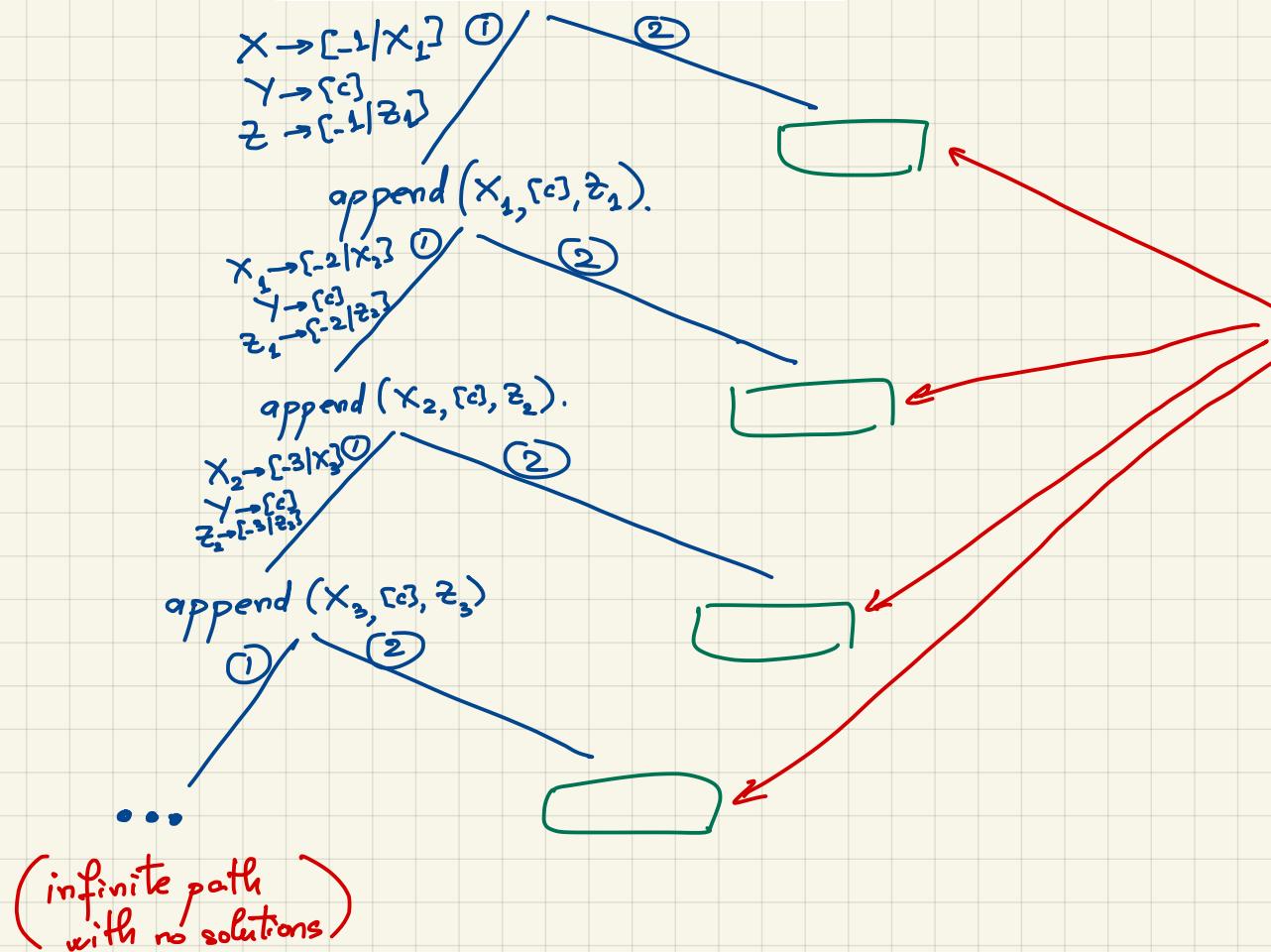
2.  $\text{append}([H|X], Y, [H|Z]) :- \text{append}(X, Y, Z).$

?-  $\text{append}(X, [c], Z).$



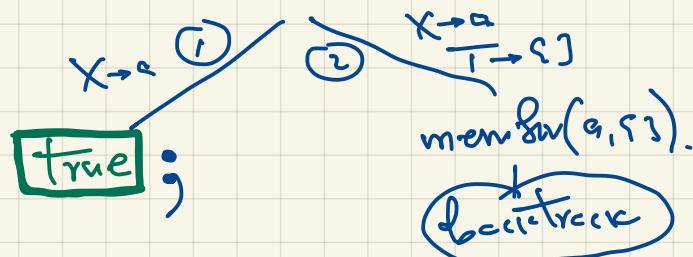
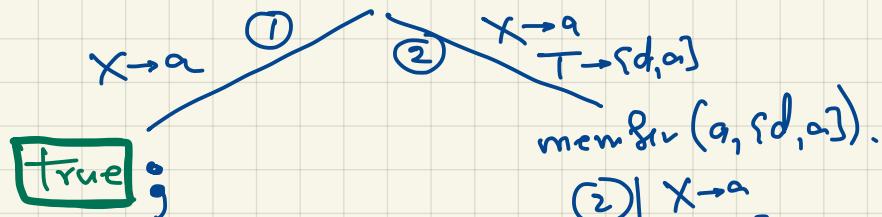
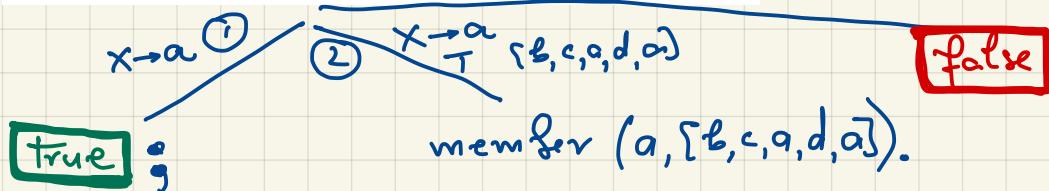
?- append(X, [c], Z).

1.  $\text{append}([H|X], Y, [H|Z]) :- \text{append}(X, Y, Z).$   
2.  $\text{append}([], Y, Y).$



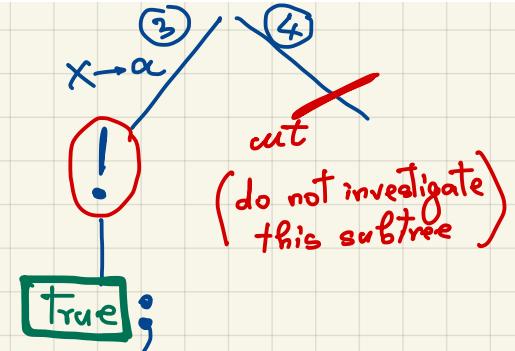
"solutions" are on  
this side of the tree  
but are never reached

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



1. `member(X, [X|_]).`  
 2. `member(X, [_|T]) :- member(X, T).`  
 3. `member1(X, [X|_]) :- !.`  
 4. `member1(X, [_|T]) :- member1(X, T).`

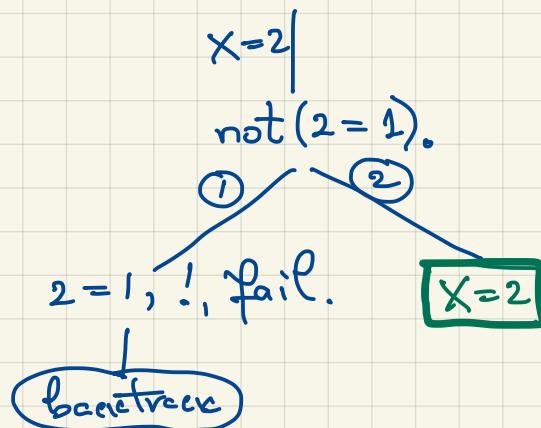
?- member1(a, [a,b,c,a,d,a]).



*backtrack*

1. `not(X) :- X, !, fail.`  
 2. `not(_).`

?- `X=2, not(X=1).`



?- `not(X=1), X=2.`

