



$k(x)$  set  $\mathcal{U}$  (Unify)

$$\sigma[] (k(x), f(a)) \longrightarrow \times$$

$$\sigma[] (k(x), k(x)) \longrightarrow \checkmark \sigma = ()$$

$$k(x) := f(x), g(x), h(x)$$

$$\sigma[], f(x), f(a)$$

$$\sigma[], f(x), f(a)$$

$$\sigma[x \mapsto a], g(x), f(a)$$

$$\times$$

$$\sigma[], f(x), f(b)$$

$$\sigma[x \mapsto a], g(x), g(b)$$

$\times$

$$\sigma[x \mapsto b], g(x), g(b)$$

$$\sigma[x \mapsto b], h(x), h(b)$$



find pseudo code

database clause (list)  
query clause (list)  
 $\text{solve}(q', q, \text{env} = [])$

if  $q$  is  $[]$

use env & print solution.  
return true // useful for backtracking in future

else  
iterate database  
if  $q[0]$  is fact

$\text{env}' = \text{unify}(q', q[0])$

if  $\text{solve}(q', q, \text{env}')$  is true

return true // useful for backtracking in future

else

back track

(i.e. make  $q' = q'[1:]$ )

& return it sol with  $q[0]$ .

(for this iterating database should be a reverse fn.)

Backtracking happens if fails

to this step

note

if  $q[0]$  is a rule

(this time also change  $q$  list) i.e.

$\text{env}' = \text{unify}(q', q[0])$

if  $\text{solve}(q', q + \text{rule body}, \text{env}')$   
return true

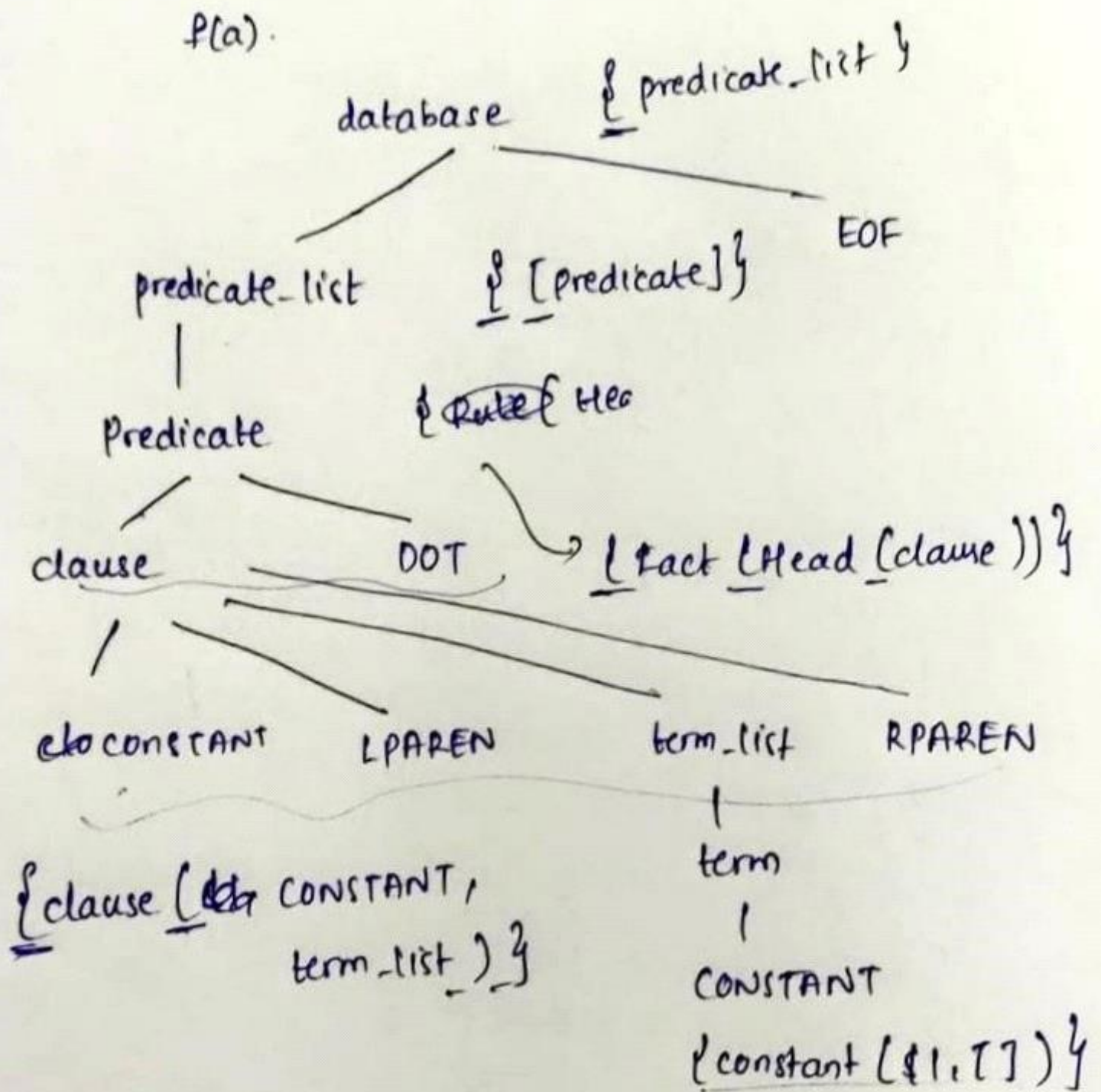
else

back track



$P(a)$   
 $P(b)$   
 $P(c)$   
 $\vdots$

} predicate list



$\{ \{ [ \{ \text{fact (Head ( \{ \text{clause (constant, \{ \text{constant (constant, [ ])} \} )} )} )} \} ] \} \}$