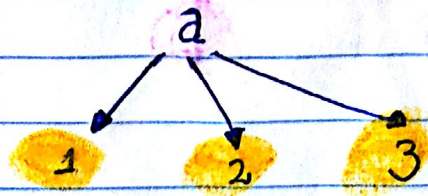


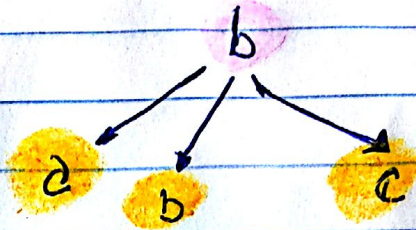
# **Prológ**

①

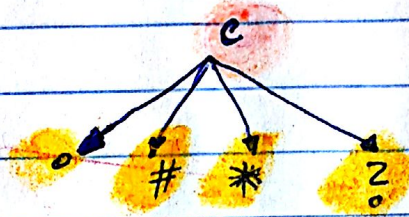
a (1)  
a (2)  
a (3)



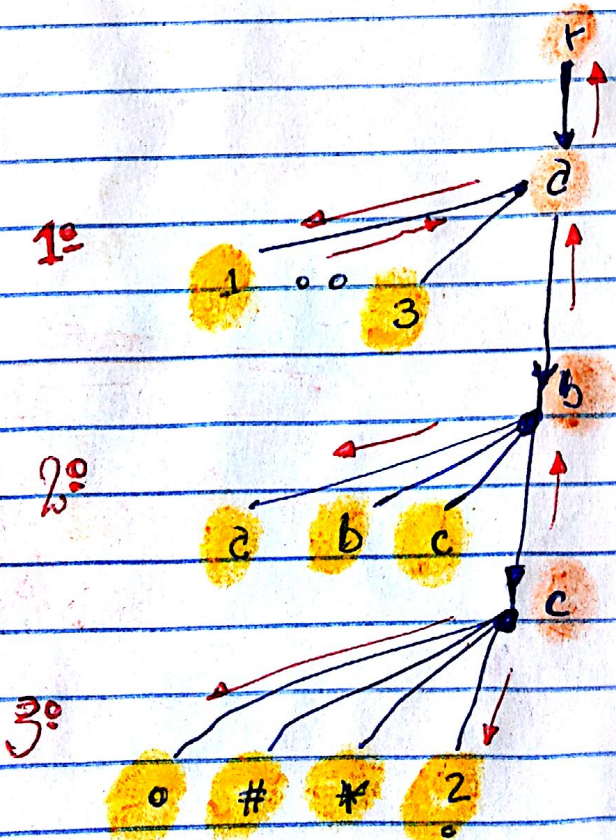
b (a)  
b (b)  
b (c)



c ('o')  
c ('#')  
c ('\*')  
c ('z')



Assim  $r(x,y,z) :- a(x), b(y), c(z).$



RESPOSTAS :

- 1, a, o
- 1, a, #
- 1, a, \*
- 1, a, z
- 1, b, o
- 1, b, #
- ...
- 3, c, \*
- 3, c, z

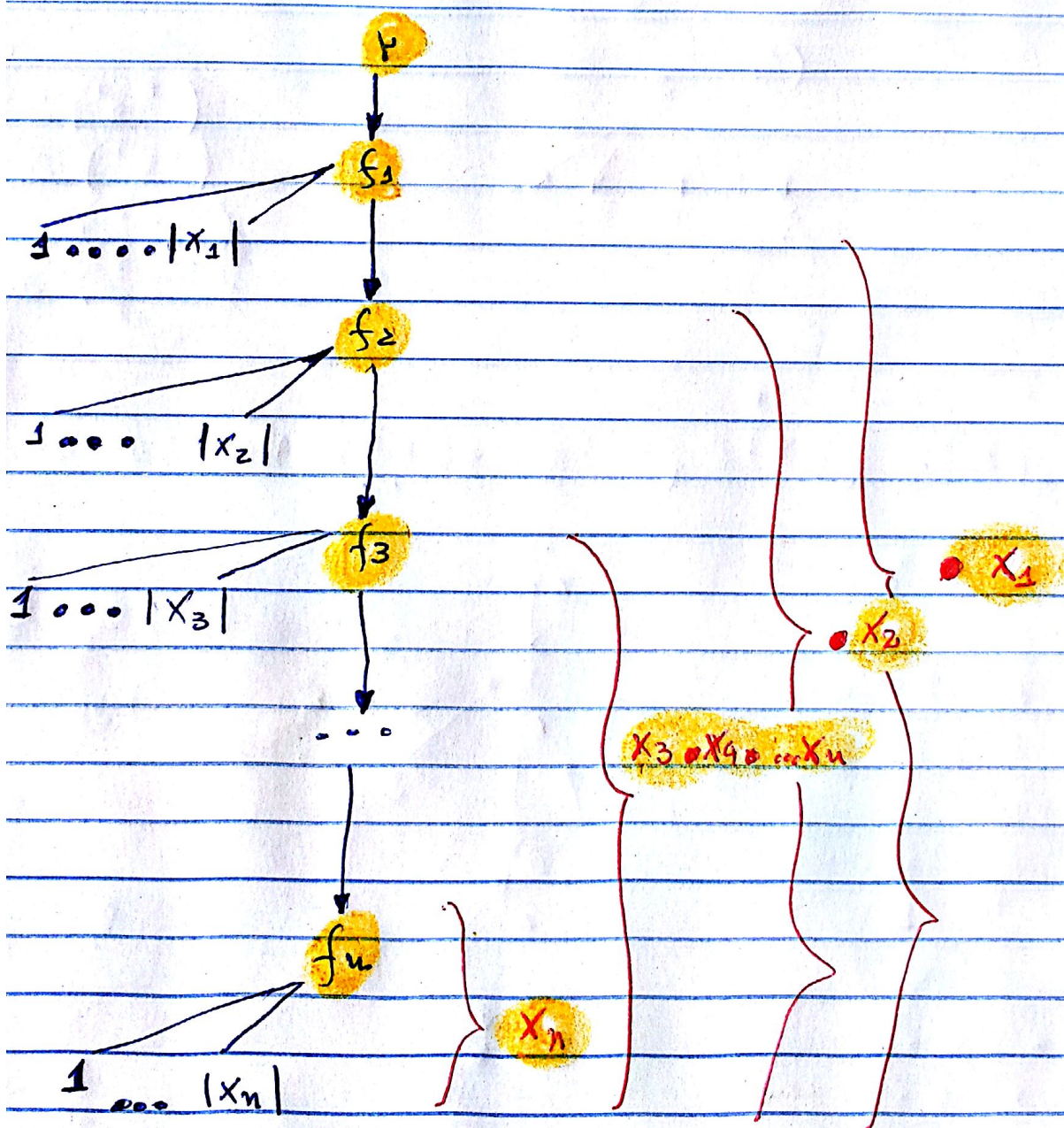
TOTAL DE RESPOSTAS :  $3 \times 3 \times 4 = 36$



Confusão:

$$\text{Regra}(x_1, x_2, x_3, \dots, x_n) :- \begin{matrix} f_1(x_1), \\ f_2(x_2), \\ \dots \\ f_n(x_n). \end{matrix}$$

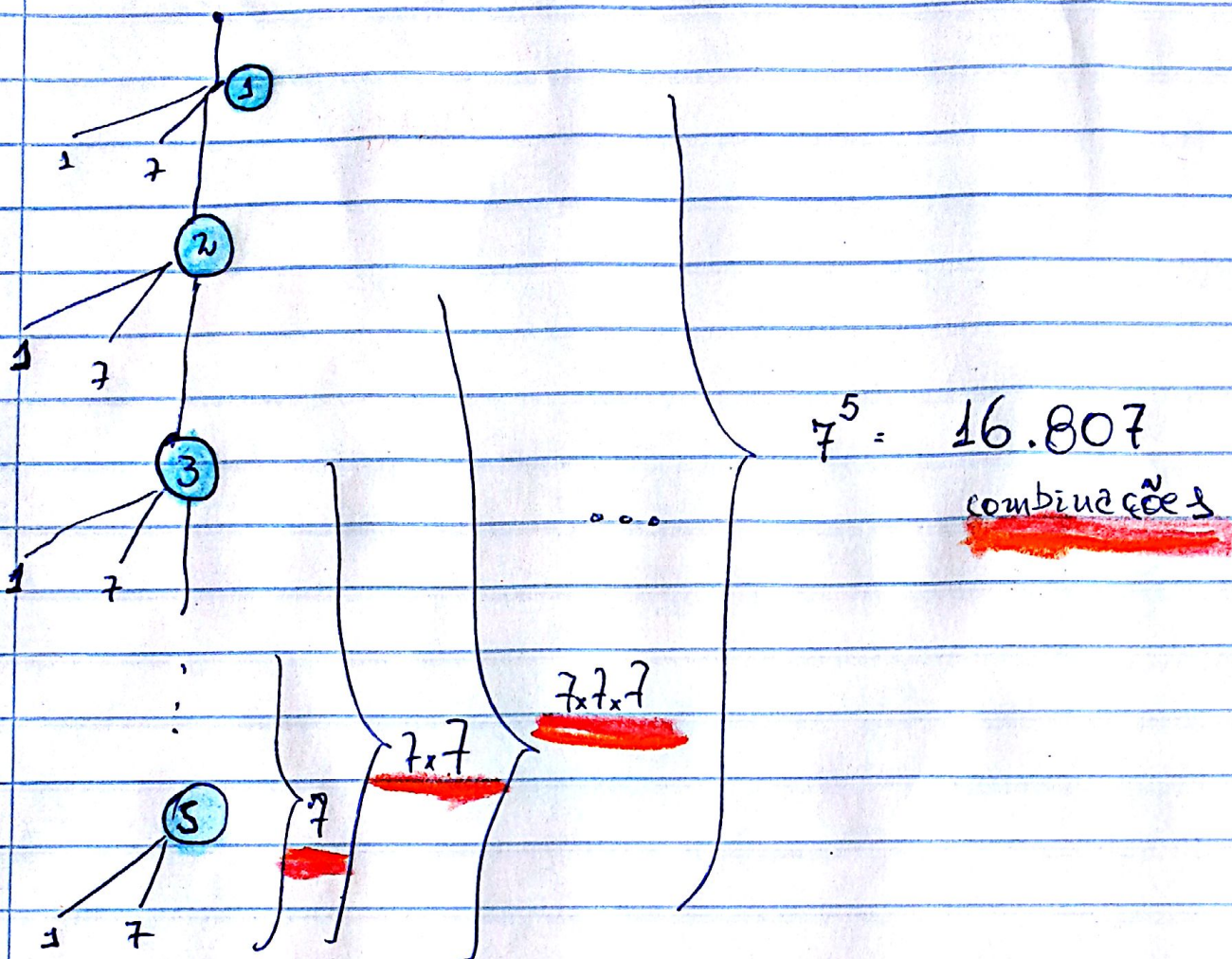
Qual o TAMANHO DESTA ÁRVORE?



Se  $x_1 \approx x_2 \approx \dots \approx x_n \Rightarrow x^n$  Uma EXPONENCIAL!  
 $\uparrow$  # qtdade de combinações



# Exemplicando:



Se um teste ou combinação gastar  $10^{-9}$  segundos  
 então  $\uparrow$  rápido

$$\left[ \begin{array}{l} 16.807, - t \\ 1 - 10^{-9} \end{array} \right] \Rightarrow t_{\text{célula}} = 1,6 \times 10^{-5} \text{ seg.}$$

rápido!

logo algo lento, como  $10^{-3}$  seg (1 miliseg.)  $\Rightarrow$

$$16.807, 10^{-3} = 16,8 \text{ segundos}$$



Assim um problema do RACHA-CUCA em  
tese dá uma resposta abaixo de 1 seg.  
Uma pior hipótese  $\approx$  20 seg.

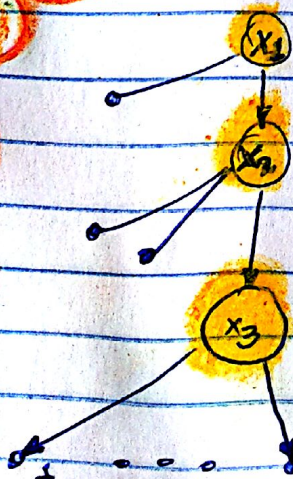
$$10^{-5} \text{ seg.} \leq t_{\text{RACHA-CUCA}} \leq 20 \text{ seg.}$$

O que fazer para melhorar o desempenho  
do Prolog para estes problemas?

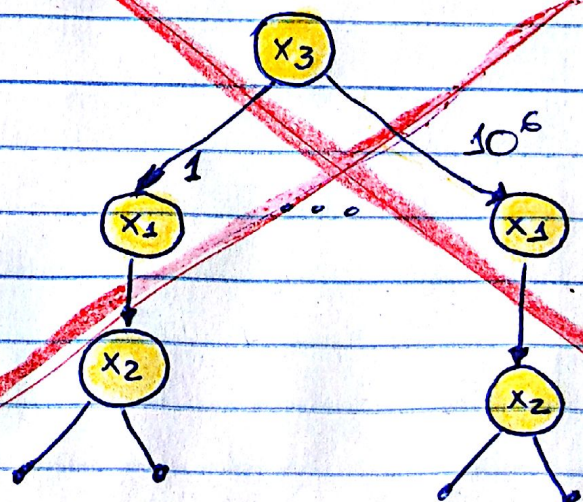
1º Atribuir o maior número de fatos  
conhecidos nas regras  
 $\Rightarrow$  Reduz o nº de combinações na árvore

2º Definir a chamada de variáveis com o  
menor domínio antes das demais.  
 $\Rightarrow$  A forma da árvore é diferente, ...  
logo ...

Ex:  $x_1, x_2, x_3$



$\neq$







prolog