

Questão 1. (valor 2 pontos) Considere os seguintes fatos sobre alguns carros:

- 1 carro(fusca, branco, 1974).
- 2 carro(gol, vermelho, 1980).
- 3 carro(voyage, preto, 1978).
- 4 carro(gol, branco, 2001).
- 5 carro(unio, branco, 2004).

Escreva uma consulta usando `findall` para obter uma lista contendo marca e ano dos carros brancos, da seguinte forma:

- 1 `L = [fusca-1974, gol-2001, uno-2004]`

Questão 2. (valor 2 pontos) Escreva o predicado `pares/2` que separa os valores pares de uma lista, conforme o seguinte exemplo de uso:

- 1 `?- pares([1,2,3,4], X).`
- 2 `X = [2, 4].`

Considere a expressão  $(A \bmod 2) = 0$  para testar se o valor de  $A$  é par.

Questão 3. (valor 2 pontos) Implemente o predicado `trunca/3` para truncar um lista de valores, conforme o seguinte exemplo de uso:

- 1 `?- trunca([10,20,30,40,50,60], 4, L).`
- 2 `L = [10, 20, 30, 40].`

Questão 4. (valor 2 pontos) Dados os fatos:

- 1 `d(0).`
- 2 `d(1).`

Qual é o resultado da seguinte consulta `prolog`:

- 1 `?- findall([A,B], (d(A), d(B)), L).`

Questão 5. (valor 2 pontos) Considere o seguinte problema: "Há dois jarros com capacidades de 3 e 4 litros, respectivamente. Nenhum dos jarros contém qualquer medida ou escala, de modo que só se pode saber o conteúdo exato quando eles estão cheios. Sabendo-se que podemos encher ou esvaziar um jarro, bem como transferir água de um jarro para outro, encontre uma sequência de passos que deixe o jarro de 4 litros com exatamente 2 litros de água". Considere que o estado inicial pode ser representado pela lista `[0,0]`, indicando que os jarros de 3 e 4 litros estão vazios inicialmente e a meta é `[_,2]`. Complete os predicados `transforma` abaixo para descrever todas as transformações possíveis de estados desse problema.

- 1 `transforma('encher o jarro 1', [X,Y], [3,Y]) :- X < 3.`
- 2 `transforma('encher o jarro 2', ...`
- 3 `transforma('esvaziar o jarro 1', ...`
- 4 `transforma('esvaziar o jarro 2', ...`
- 5 `transforma('transferir do jarro 1 para o 2', ...`
- 6 `transforma('transferir do jarro 2 para o 1', ...`
- 7 `%— considerando que ainda restara agua no jarro de origem`
- 8 `transforma('transferir do jarro 1 para o 2', ...`
- 9 `transforma('transferir do jarro 2 para o 1', ...`

① ~~consulta~~ <sup>branco</sup> ~~função~~ (S) := fundall (M-A, ~~corres~~ (M, ~~X~~, A), S).

② ~~par~~ (A) := (A mod 2) = 0.

parus (L, S) := fundall (P, (member (P, L), par (P)), S).

trunca (-, 0, []).

③ trunca ([A-], 1, [A]).

trunca (~~[X, Y]~~, N, [X|Z]) := M is N-1, trunca (Y, M, Z).

[X|Y]

↑  
Isto é uma lista com  
2 valores

④ d(0).

d(1).

fundall ([A|B], (d(A), d(B)), L)

L = [[0, 0], [1, 0], [0, 1], [1, 1]].

⑤ interval ([0, 0]).

interval (-, 2]).

[2] [3]

[3] [2] 3+2-3=2

mº do } 1 ... :- X < 3.

limbo } 2 ... , [X, Y], [X, 4]) :- Y < 4.

3 ... , [X, Y], [0, Y]) :- X > 0.

4 ... , [X, Y], [X, 0]) :- Y > 0.

5 ... , [X, Y], [0, X+Y]) :- X > 0, Y < 4, X+Y ≤ 4.

6 ... , [X, Y], [X+Y, 0]) :- X < 3, Y > 0, X+Y ≤ 3.

8 ... , [X, Y], [X+Y-3, 4]) :- X > 0, Y < 4, X+Y-3 ≤ 4.

9 ... , [X, Y], [3, X+Y-4]) :- X < 3, Y > 0, X+Y-4 ≤ 3.

Não é possível  
fazer os cálculos  
aqui!!