

Lista 2

Sistemas Baseados em Conhecimento (MAC0444/MAC5778)

Data de entrega - 17 de outubro

1. Formalize as seguintes sentenças usando os predicados unários `fezEx`, `vaiBem`, `mediaAlta` e o predicado binário `aprovado`:

“Quem fez os exercícios vai bem na prova.”

“Quem vai bem na prova fica com média alta.”

“Quem fica com média alta é aprovado em mac444.”

“João fez os exercícios.”

“Maria foi bem na prova.”

Use **resolução SLD** para mostrar que João e Maria foram aprovados em mac444.

2. Mostre passo a passo que o procedimento de **encadeamento para trás** (*backward chaining*) produz resposta SIM com objetivo $P(a)$ e uma base de conhecimento formada pelas seguintes cláusulas:

- $[\neg A_1(x), \neg A_2(x), P(x)]$
- $[\neg B_1(x), \neg B_2(x), A_1(x)]$
- $[\neg B_3(x), \neg B_4(x), A_2(x)]$
- $[B_1(a)]$
- $[B_2(a)]$
- $[B_3(a)]$
- $[B_4(a)]$

3. Considere o seguinte programa em Prolog:

```
result([_, E | L], [E | M]) :- !, result(L, M).  
result(_, []).
```

(a) Após ter carregado este programa, qual seria a resposta do Prolog para a seguinte consulta?

```
?- result([a, b, c, d, e, f, g], X).
```

(b) Descreva brevemente **o que** o programa faz e **como** ele o faz quando o primeiro argumento do predicado `result/2` é instanciado com uma lista e uma variável é passada no segundo argumento, assim como no item (a). Suas explicações devem incluir respostas às seguintes perguntas:

- Qual(is) caso(s) é(são) coberto(s) pelo fato?
- Qual efeito tem o corte na primeira linha do programa?

- Por que foi utilizada a variável anônima?

4. Suponha a seguinte base de conhecimento:

```
homem(americo).
homem(daniel).
homem(paulo).
homem(carlos).
homem(joaquim).
homem(filipe).
mulher(teresa).
mulher(sonia).
mulher(ana).
mulher(carla).
mulher(barbara).
mulher(maria).
idade(americo, 18).
idade(daniel, 60).
idade(paulo, 25).
idade(carlos, 37).
idade(joaquim, 80).
idade(filipe, 32).
idade(teresa, 18).
idade(sonia, 28).
idade(ana, 17).
idade(carla, 26).
idade(barbara, 51).
idade(maria, 79).
irmaos(americo, paulo).
irmaos(carlos, sonia).
pai(carlos, teresa).
pai(daniel, americo).
pai(daniel, paulo).
pai(joaquim, daniel).
mae(maria, daniel).
mae(barbara, ana).
casados(filipe, carla).
casados(americo, teresa).
casados(joaquim, maria).
```

Escreva regras para os seguintes predicados:

- (a) avof(Mul, Pess) em que Mul seja avó de Pess.
- (c) bisavom(Hom, Pess) que é verdadeiro se Hom for bisavô de Pess.
- (d) primo_1(P1, P2) que é verdadeiro se P1 e P2 forem primos em primeiro grau.
- (f) maior_de_idade(Pess) que é verdadeiro se Pess for maior de idade.
- (g) pessoas(Lista) que devolve a Lista de todas as pessoas existentes na base de conhecimento.
- (h) mais_velho(Pess) que retorna a pessoa mais velha que consta na base de conhecimento.

(i) `lista_pessoas(Lista, Sexo)` que retorna uma `Lista` de todas as pessoas do `Sexo` indicado (m/f), incluindo as suas respectivas idades. Por exemplo, `lista_pessoas(Lista, m)` deveria retornar:

```
Lista=[[americo,18], [daniel,60], [paulo,25],  
       [carlos,37], [joaquim,80], [filipe,32]].
```

(j) `adequados(Hom, Mul)` que é verdadeiro se `Hom` for um homem, `Mul` for uma mulher e o homem for (no máximo) 2 anos mais novo do que a mulher ou 10 anos mais velho do que ela e se ambos não tiverem nenhuma relação de parentesco nem nenhum deles for casado.

Vocês podem utilizar o comando `findall` do Prolog, que devolve uma lista com todas as soluções. Vejam nesta página do livro de Prolog como utilizá-lo:

<http://www.learnprolognow.org/lpnpage.php?pagetype=html&pageid=lpn-htmlse49>