

ECM253 – Linguagens Formais, Autômatos e Compiladores

Lista de Exercícios

Prolog

Marco Furlan

20 de março de 2019

1. Elaborar uma base de dados Prolog com o nome `familia.pl` que contenha regras para definir:
 - (a) **Filho:** X é filho de Y se Y é pai de X e X é homem.
 - (b) **Filho:** X é filho de Y se Y é mãe de X e X é homem.
 - (c) **Filha:** X é filha de Y se Y é pai de X e Y é mulher.
 - (d) **Filha:** X é filha de Y se Y é mãe de X e Y é mulher.
 - (e) **Um dos pais de:** Y é um dos pais de X se Y é pai de X .
 - (f) **Um dos pais de:** Y é um dos pais de X se Y é mãe de X .
 - (g) **Um avô ou avó:** X é um avô ou avó se X for um dos pais de alguém que é um dos pais de alguma outra
 - (h) **Ancestral:** X é ancestral de Y se X for um dos pais de Y .
 - (i) **Ancestral:** X é ancestral de Y se X for um dos pais de alguém que seja ancestral de Y . pessoa.

Defina consultas que testem todas as regras pedidas acima. Se quiser, utilize os fatos a seguir (você pode propor outros, se quiser):

- José é pai de João.
- José é pai de Nachor.
- José é pai de Marcio.
- João é pai de Flavio.
- Marcio é pai de Ricardo.
- Marcio é pai de Julia.
- Marcio é pai de Maria.
- Silvia é mãe de Flavio.
- São homens: José, João, Marcio, Nachor, Flavio, Ricardo.
- São mulheres: Silvia, Julia, Maria.

2. No estado ABC existem diversas cidades ligadas diretamente por trem. O predicado `trem_direto` é utilizado a seguir para descrever tais cidades:

```
trem_direto(A,B).
trem_direto(C,A).
trem_direto(D,E).
trem_direto(E,A).
trem_direto(B,F).
trem_direto(G,D).
trem_direto(H,G).
```

Elaborar um predicado recursivo binário denominado `trem_entre` que indicará se é possível viajar de trem entre as cidades passadas como argumento. Testar este predicado na base de dados apresentada.

3. Imagine que a base de dados Prolog a seguir descreva um labirinto:

```
conectado(1,2).
conectado(3,4).
conectado(5,6).
conectado(7,8).
conectado(9,10).
conectado(12,13).
conectado(13,14).
conectado(15,16).
conectado(17,18).
conectado(19,20).
conectado(4,1).
conectado(6,3).
conectado(4,7).
conectado(6,11).
conectado(14,9).
conectado(11,15).
conectado(16,12).
conectado(14,17).
conectado(16,19).
```

Esses fatos determinam que pontos estão conectados, isto é, que ponto se pode alcançar a partir de um outro, em um único passo. Além disso, são caminhos de mão-única: só se pode caminhar em uma única direção.

Definir o predicado binário `caminho` que responderá se um determinado ponto pode ser alcançado a partir de um outro qualquer, em um número arbitrário de passos. Testar este predicado de diversas formas, por exemplo perguntando se há um caminho entre dois pontos quaisquer e que pontos podem ser alcançados a partir de um ponto inicial. Salve este programa com o nome `labyrinth.pl`.

4. Considere o programa Prolog a seguir (teste-o para ver os resultados):

```
person(socrates).
person(plato).
person(aristotle).

mortal(X):-
    person(X).

mortal_report:-
    write('Known mortals are :') ,nl,
    mortal(X),
    write(X), nl,
    fail.
```

Explicação sobre este programa. O predicado `write` escreve uma cadeia de caracteres constante ou o valor de uma variável na saída padrão (console). O predicado `nl` escreve o caractere de fim de linha na saída padrão. O predicado `fail` força a execução de “backtracking” pelo Prolog, de modo que ele vai tentar outras soluções até que não seja mais possível. Desse modo, ele tentará automaticamente unificar *mortal* e *write* até que não seja mais possível, exibindo, portanto todos os valores de pessoas, ou mortais.

Pede-se: alterar o programa do exercício 3 para que ele exiba os caminhos percorridos de um ponto de partida a um ponto de chegada. Você precisará alterar os predicados existentes. Por exemplo, você pode adicionar um predicado que dará início a execução denominado `listar_caminho` e usá-lo assim:

```
?- listar_caminho(16,20).
Caminho:
19-20
16-19
false.
```

Não importa que a solução apareça de modo reverso.

5. Declare uma base de dados Prolog que forneça informações sobre estados e suas capitais. Algumas cidades são grandes, outras pequenas. Alguns estados estão ao sul, outros ao norte.
- (a) Escreva uma pergunta para achar todas as cidades capitais.
 - (b) Escreva uma pergunta para achar todos os estados cujas capitais sejam cidades pequenas.
 - (c) Escreva uma pergunta para achar todos os estados ao norte cujas capitais sejam grandes cidades.
 - (d) Formule uma regra para definir as cidades cosmopolitas como as cidades grandes que sejam capitais dos estados do sul.
 - (e) Escreva uma pergunta para achar todas as cidades cosmopolitas.

Defina os predicados da sua maneira, de modo que seja possível responder as questões apresentadas.