

Trabalho 2

Problemas em Prolog

Disciplina: 5200 – Paradigma de Programação Lógica e Funcional

Professor: Lucas Pupulin Nanni

Introdução

O objetivo deste trabalho é a implementação de quatro problemas em Prolog. Todos os predicados construídos devem ser devidamente documentados com a assinatura e a descrição dos mesmos. A descrição do predicado deve conter o significado da resposta quando satisfeito e o funcionamento lógico do mesmo. O predicado principal de cada problema (o predicado que será consultado para obter a solução do problema) deve possuir um conjunto de testes junto de sua implementação.

O trabalho é em equipe de até duas pessoas. O compartilhamento de informações entre as equipes é permitido (e aconselhado), mas o compartilhamento de código não é permitido. Trabalhos que tenham porções significativas de código iguais, ou copiadas da internet, serão anulados. Veja a [resolução Nº 008/2007-COU](#) para as possíveis sanções disciplinares.

Este trabalho vale 2/3 (dois terços) da nota do primeiro bimestre.

Data de entrega: até o dia 10/08/2016 às 23:55.

Problema #1 – Lógica

O Reitor convocou quatro alunos suspeitos de terem pichado os muros da Universidade para depor. Ele sabe que apenas um deles cometeu o crime, mas não sabe quem foi. Além disso, o Reitor ouviu dizer que, dentre os quatro alunos, apenas um fala a verdade. Quando interrogados sobre o ato, os alunos responderam o seguinte:

- Ana disse que não foi ela.
- Bruno disse que foi o Carlos.
- Carlos disse que foi o Daniel.
- Daniel disse que não foi ele.

Como o Reitor não sabe ao certo quem cometeu o crime, ele pediu auxílio aos alunos da disciplina de PPLF. Portanto, sua tarefa é escrever em Prolog um predicado **solucao(?C, ?V)** que é verdadeiro se C é o aluno criminoso e V é o aluno que fala a verdade.

Dica: Tente primeiramente resolver o problema no papel e então aplique a mesma lógica na escrita do predicado. Como você consegue garantir que apenas um cometeu o crime e apenas um fala a verdade?

Problema #2 – Puzzle 8-Rainhas

O problema das 8-rainhas é um problema clássico da computação. O objetivo é posicionar 8 rainhas em um tabuleiro de xadrez de forma que as rainhas não se ataquem, isto é, duas rainhas não podem estar posicionadas na mesma linha, coluna ou diagonal.

Sua tarefa é escrever dois predicados em Prolog:

- **rainhas_p(?Q)** que é verdadeiro se Q é uma solução para o problema das 8-Rainhas utilizando o processo de gerar e testar a solução. O processo de gerar e testar consiste em construir todas as possíveis configurações do problema (uma de cada vez) e então testar uma a uma se é solução.
- **rainhas_r(?Q)** que é verdadeiro se Q é uma solução para o problema das 8-Rainhas utilizando retrocesso (*backtracking*). O Processo de *backtracking* consiste em resolver um problema de tamanho N-1 recursivamente e, para cada possibilidade de posicionar uma nova rainha no tabuleiro (tornando o tamanho do problema igual a N), testar se o posicionamento é uma solução.

Dica: Utilize uma representação da solução que seja fácil de manipular e construir. Uma estratégia é representar a solução como uma lista de inteiros que indicam a linha de cada rainha, com a coluna implícita na posição do elemento na lista.

Exemplo:

```
?- rainhas_p(Q).  
Q = [1, 5, 8, 6, 3, 7, 2, 4] ;  
Q = [1, 6, 8, 3, 7, 4, 2, 5] ;  
...
```

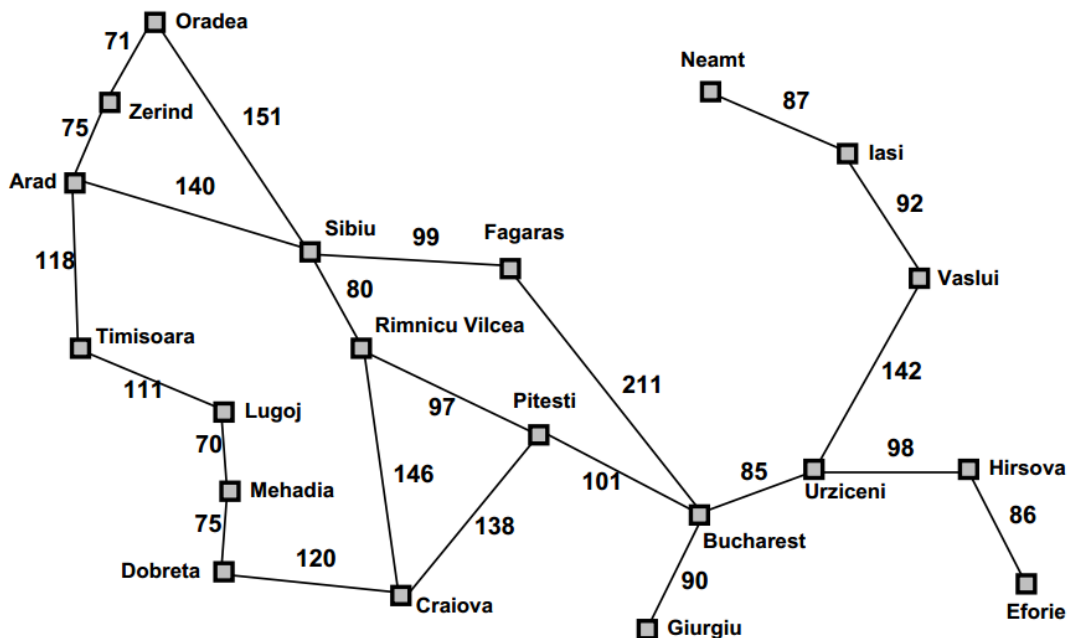
Observe que na primeira resposta, interpretamos o posicionamento das rainhas como: uma rainha na coluna 1 / linha 1, outra rainha na coluna 2 / linha 5, outra rainha na coluna 4 / linha 8, e assim sucessivamente.

Bônus (+1 ponto na prova): Nós podemos generalizar o problema das 8-Rainhas ao permitir um número arbitrário N para a quantidade de rainhas e dimensão do tabuleiro. Dessa forma, escreva um predicado **rainhas(+N, ?Q)** que é verdadeiro se Q é uma solução para o problema das N-rainhas. Faça as duas variações do predicado (utilizando o processo de gerar e testar e o *backtracking*).

Problema #3 – Viagem à Bucareste

Você foi contratado por uma agência de viagens da Romênia que deseja descobrir os caminhos mais curtos partindo das principais cidades até a capital Bucareste. Você recebeu o mapa das principais estradas que ligam as cidades da Romênia, juntamente de uma tabela que relaciona as distâncias em linha reta entre Bucareste e as demais cidades.

Sua tarefa é implementar um predicado `melhor_caminho(?O, ?C)` que é verdadeiro se `C` é o melhor caminho partindo da cidade de origem `O` até Bucareste. O seu predicado deve descobrir o melhor caminho realizando uma busca gulosa com a heurística da distância em linha reta até Bucareste.



Distância em linha reta até Bucareste

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Observação: Assuma a grafia dos nomes das cidades como a definida pelo mapa e pela tabela.

Problema #4 – Puzzle Vizinhos

Na cidade de Fagaras há uma rua em que:

1. Há cinco casas de diferentes cores;
2. Em cada casa mora uma pessoa de uma diferente nacionalidade;
3. Esses cinco proprietários bebem diferentes bebidas, fumam diferentes tipos de cigarros e têm um animal de estimação;
4. Nenhum deles tem o mesmo animal, fumam o mesmo cigarro ou bebem a mesma bebida.

Sobre essa rua, sabe-se que:

1. O Norueguês vive na primeira casa.
2. O Inglês vive na casa Vermelha.
3. O Sueco tem Cachorros como animais de estimação.
4. O Dinamarquês bebe Chá.
5. A casa Verde fica do lado esquerdo da casa Branca.
6. O homem que vive na casa Verde bebe Café.
7. O homem que fuma Pall Mall cria Pássaros.
8. O homem que vive na casa Amarela fuma Dunhill.
9. O homem que vive na casa do meio bebe Leite.
10. O homem que fuma Blends vive ao lado do que tem Gatos.
11. O homem que cria Cavalos vive ao lado do que fuma Dunhill.
12. O homem que fuma BlueMaster bebe Cerveja.
13. O Alemão fuma Prince.
14. O Norueguês vive ao lado da casa Azul.
15. O homem que fuma Blends é vizinho do que bebe Água.
16. Alguém possui um aquário com peixes.

Sua tarefa é escrever um predicado **vizinhos(?Rua)** que é verdadeiro se Rua é a lista das cinco casas, com suas respectivas informações sobre cor, nacionalidade do seu proprietário, bebida e cigarro consumidos por ele, bem como seu animal de estimação.

Dica: consulte a resolução do exercício 4.8.

Exemplo:

```
?- vizinhos(Rua).  
Rua = [casa(amarela, norueguês, ...), casa(azul, ...), ...]
```

Envio do trabalho

Cada problema deve ser implementado em seu arquivo particular. Nomeie os arquivos com da seguinte forma:

Problema #1 → “logica.pl”

Problema #2 → “rainhas.pl”

Problema #3 → “bucaresta.pl”

Problema #4 → “vizinhos.pl”

Para enviar o trabalho, crie uma versão compactada no formato **zip** com os arquivos fonte. Renomeie o arquivo de forma a conter o RA dos integrantes da equipe, como no exemplo a seguir.

Exemplos:

- ra12345_ra54321.zip (dois integrantes)
- ra12345.zip (um integrante)

O arquivo compactado deve ser enviado até as 23:55 do dia 10/08/2016 pelo Moodle. Arquivos compactados que não seguirem esse procedimento serão desconsiderados.

Avaliação

Este trabalho vale 2/3 (dois terços) da nota do primeiro bimestre. O trabalho será avaliado de acordo com os critérios:

- Clareza lógica: a lógica nos programas deve ser clara.
- Corretude e completude: os programas têm que passar nos testes que você escrever.
- Boas práticas de programação: o código deve estar bem escrito e organizado; os recursos da linguagem devem ser usados corretamente.