



UNIVERSIDADE ESTADUAL DE MARINGÁ

DEPARTAMENTO DE INFORMÁTICA

Prof. Dr. Wagner Igarashi

6902 – Paradigma de Programação Lógica e Funcional

Juliano César Chagas Tavares

RA: 87940

O Problema das 10 Rainhas

Maringá

10 de outubro de 2016

Elaboração do Algoritmo

O algoritmo foi elaborado pensando na verificação das rainhas par a par para saber se há conflito ou não. A função **foraAtaque** é a responsável por extrair todas as rainhas da lista e chamar a função **conferePar** que irá verificar na linha, coluna e diagonal se há conflito. Como dito anteriormente a função **conferePar** irá fazer a verificação de igualdade nas posições de linha, coluna ou diagonal. A função **noAlcance** é responsável por controlar os limites do tabuleiro durante a execução do código para que não haja estouro. A função **printTabuleiro** é a responsável por exibir o tabuleiro na tela. Para tal ela faz uso de **noAlcance** e da função **printLinha** que será explicada a seguir. Além do **printTabuleiro**, há também a **printTabuleiroConflito**, pois dependendo da chamada que é realizada para as rainhas, a lista é montada de uma forma diferente, havendo a necessidade de processar a lista de trás para frente. A função **printLinha** possui duas variações principais, a que pode exibir um 'R' caso a rainha esteja fora de perigo e a que pode exibir um 'C' quando a rainha está sob ataque. Além disso ambas funções podem imprimir '_' caso não seja uma célula ocupada.

Para a construção do algoritmo das rainhas propriamente dito, se fez necessário a construção de cinco variações:

- Uma com apenas um argumento **rainhas(ListaInicial)** utilizada para a verificação de validade da lista passada como parâmetro
- **Rainhas(N, Solucao)**, utilizada para passar um valor de N com a quantidade de rainhas e gerar todas as soluções possíveis.
- **Rainhas(A,B,C,D,E,F,G,H,I,J)**, que é a função que verifica os valores dados, faz os testes de validade, retorna caso não estejam corretos e, caso estejam, verifica a validade da combinação, se não estiver correto, imprime uma combinação possível na tela.
- **Rainhas(M, SolucaoParcial, Final)**, que é a função que realmente resolve o problema das rainhas.
- **Rainhas(0, Solucao, Solucao)**, que é o caso base.

Prints das Telas

```
?- rainhas(7,4,2,9,5,10,8,6,3,1).  
  
      R  
    R  
  R      R  
    R      R  
R          R  
      R      R  
    R      R  
  
Nao ha conflitos!  
true.
```

Figura 1 - Teste 10 Rainhas Com Disposição Correta

```
?- rainhas(2,3,1,4,5,7,6,8,10,9).  
  
      R  
C      C  
      C      C  
      C      C  
      C      R  
      R      R  
      C  
  
Ha conflitos no tabuleiro. Possivel resposta:  
      R  
    R      R  
R          R  
      R      R  
    R      R  
      R      R  
      R  
  
true.
```

Figura 2 - Teste 10 Rainhas Com Disposição Errada

```

?- rainhas(7,4,B,9,5,10,8,6,3,12).
Corrija o valor do argumento C!
false.

?- rainhas(7,4,2,9,5,10,8,6,3,3).
Ha valores repetidos nos argumentos!
false.

?- rainhas(7,4,B,9,5,10,8,6,3,A).
Corrija o valor do argumento C!
false.

?- rainhas(7,4,2,9,5,10,8,6,3,12).
O valor de J deve ser menor do que 10!
false.

```

Figura 3 - Exemplos de Erros Informados

```

?- rainhas(8,N).

  R
  R
  R
  R
  R
  R
  R
  R
N = [4, 2, 7, 3, 6, 8, 5, 1] ;

  R
  R
  R
  R
  R
  R
  R
  R
N = [5, 2, 4, 7, 3, 8, 6, 1] ;

  R
  R
  R
  R
  R
  R
  R
  R

```

Figura 4 - Teste 8 Rainhas Gerando Possíveis Respostas