Concurso/Encontro Nacional de Programação em Lógica
CeNPL'02

Universidade de Coimbra / Instituto Politécnico de Coimbra

11-13 de Abril de 2002

Problema nº 6

CAVALEIRO DE EULER

Introdução

O problema que enunciamos de seguida foi colocado pelo matemático Leonhard Euler há mais de 200 anos. Diz respeito à procura de um caminho Hamiltoniano num tabuleiro de xadrez (caminho que passa uma e uma só vez por cada uma das casas do tabuleiro).

Começamos por lembrar que o cavalo de xadrez, numa dada casa do tabuleiro, pode ocupar, depois do seu salto, um máximo de oito casas diferentes. Ele pode movimentar-se duas casas para cima, seguido de uma deslocação para a esquerda ou para a direita; movimentar-se de duas casas para baixo, seguido de uma deslocação para a esquerda ou para a direita; movimentar-se de duas casas para a direita, seguido de uma deslocação para cima ou para baixo; movimentar-se de duas casas para a esquerda, seguido de uma deslocação para cima ou para baixo. A questão colocada por Euler foi a de saber se dado um tabuleiro $N \times N$ é possível, ou não, a um cavalo, partindo de uma casa 1, passar uma só vez por cada uma das restantes $N^2 - 1$ casas. Esta tem sido uma questão bem estudada, com algumas publicações científicas recentes, e algumas variações do enunciado. Sabe-se que para tabuleiros de dimensão N ? 5 existem sempre muitas soluções. Segue-se uma resposta possível para o tabuleiro de xadrez convencional (8×8):

1	42	37	44	25	4	15	18
38	55	40	3	14	17	24	5
41	2	43	36	45	26	19	16
56	39	54	13	48	35	6	23
63	12	57	46	61	22	27	20
58	53	62	49	34	47	30	7
11	64	51	60	9	32	21	28
52	59	10	33	50	29	8	31

O Algoritmo de Warnsdorff

É fácil engendrar um método de pesquisa para encontrar um caminho, se um existir: backtracking. Deslocamos o cavalo tão longe quanto possível e, se chegarmos a um beco sem saída, voltamos atrás de maneira a seguirmos outra direcção. O problema é que para tabuleiros grandes (nem é preciso que N seja tão grande quanto isso...) o backtracking torna-se muito lento.

Em 1823 Warnsdorff propôs um algoritmo com a pretensão de encontrar sempre um caminho (quando ele existe) sem nunca retroceder. Em cada posição do cavalo faz-se uma classificação das posições sucessoras, deslocando-se o cavalo para o sucessor com a maior classificação. As posições sucessoras são as casas do tabuleiro que ainda não foram visitadas e podem ser alcançadas por um único salto do cavalo a partir da posição corrente. A classificação é maior para o sucessor com menor número de sucessores. Desta maneira, as casas que tendem a estar isoladas são visitadas primeiro, evitando-se que fiquem de facto isoladas.

O tempo necessário para a execução deste algoritmo cresce linearmente com o número de casas do tabuleiro, o que é bom, mas infelizmente sabemos hoje, com a ajuda dos computadores (que não estavam disponíveis no tempo de Warnsdorff) que para tabuleiros de lado N ??76 podem ocorrer becos sem saída. Existem algoritmos recentes de complexidade linear para todo o N, mas são extremamente complicados, lidando com muitos casos particulares cujas soluções são já conhecidas à priori. A questão de saber se existe um algoritmo, comparável ao de Warnsdorff em simplicidade, que resolva o problema em tempo linear para todo o N = 5, permanece em aberto.

Tarefa

Na verdade, o algoritmo de Warnsdorff não elimina completamente a necessidade de backtracking. Surgem situações em que existe mais do que um sucessor com menor número de sucessores, alguns dos quais conduzem a becos sem saída. Aqui vamos pedir que implementem um melhoramento do esquema de classificação de Warnsdorff. Acrescentamos a regra que, em caso de empate, escolhe um dos sucessores (X,Y) com maior distância ao centro do tabuleiro: distancia = $(X - C)^2 + (Y - C)^2$, com C = N/2, onde N é a dimensão do lado do tabuleiro. Note-se que, mesmo com esta regra adicional, podem existir vários candidatos à posição seguinte do cavalo (todas elas devem ser usadas, se necessário, por backtracking). Para garantir a unicidade de solução, deve adoptar as seguintes convenções:

- Prioridade dos saltos por ordem decrescente (olhe para o tabuleiro como uma matriz - o primeiro elemento do par denota o incremento na linha, o segundo o incremento na coluna): (-1,2), (-2,1), (-2,-1), (-1,-2), (1,-2), (2,-1), (2,1), (1,2).
- Considere como ponto de partida do cavalo o canto superior esquerdo: (1,1). No caso de não existir solução (com as regras adoptadas) deve procurar uma usando um outro ponto de partida. Partindo de (1,1) deve incrementar a coluna e só depois a linha: (1,1), (1,2), (1,3), …, (N,N).

A sua tarefa consiste então em escrever um programa Prolog que dada a dimensão N do tabuleiro (N = 1) devolva o tabuleiro T, descrito por uma lista de linhas, onde cada linha é uma lista de N números da sucessão $1,\ldots,N^2$, que descreve o caminho descrito pelo ``Cavaleiro de Euler''. Implemente para isso o predicado `wm(N,T)` (iniciais de *Warnsdorff Melhorado*).

Predicados Dados

O seu programa será testado através do seguinte predicado `ce(N)` (iniciais de *Cavaleiro de Euler*):

```
ce(N) :- wm(N,L), mostra(L).

mostra([]).
mostra([L|R]) :- mostraLinha(L), nl, mostra(R).

mostraLinha([]).
mostraLinha([X|R]) :-
  X < 10, write('  '), write(X), write(' '), mostraLinha(R).
mostraLinha([X|R]) :-
  X < 100, write(' '), write(X), write(' '), mostraLinha(R).
mostraLinha([X|R]) :-
```

```
    write(X), write(' '), mostraLinha(R).
```

Os Resultados

Mostramos de seguida alguns exemplos que ilustram os resultados esperados.

```
?- ce(2).

No

?- ce(5).
  1  10  15  20   3
 16  21   2   9  14
 11   8  23   4  19
 22  17   6  13  24
  7  12  25  18   5

Yes

?- ce(8).
  1  42  37  44  25   4  15  18
 38  55  40   3  14  17  24   5
 41   2  43  36  45  26  19  16
 56  39  54  13  48  35   6  23
 63  12  57  46  61  22  27  20
 58  53  62  49  34  47  30   7
 11  64  51  60   9  32  21  28
 52  59  10  33  50  29   8  31

Yes
```