Concurso/Encontro Nacional de Programação Lógica CNPL'98

Dep. de Matemática, Univ. de Aveiro

27–28 de Abril de 1998

Cobre

Coberturas

Considere que tem um tabuleiro de `N x M` casas e um número ilimitado de pedras de dominó (todas iguais e sem pintas). As dimensões das casas do tabuleiro e das pedras de dominó são tais que uma pedra de dominó cobre precisamente duas casas do tabuleiro.

Se tivermos um tabuleiro `2x2` bastam duas pedras de dominó para cobrir o tabuleiro e todas as pedras do dominó estão a cobrir alguma casa do tabuleiro, havendo assim uma cobertura perfeita. No entanto, se retirarmos um canto do tabuleiro já não será possível cobrir o tabuleiro de um modo perfeito.

Se tivermos um tabuleiro `4x4` é fácil de verificar que são necessárias exactamente 8 pedras para cobrir perfeitamente o tabuleiro. O que acontecerá se fizermos dois buracos no tabuleiro, isto é, se retirarmos duas casas quaisquer do tabuleiro? Será ainda possível cobrir perfeitamente o tabuleiro?

Tarefa

Implemente um programa Prolog para determinar se um tabuleiro (eventualmente com buracos) pode ser coberto perfeitamente por peças de dominó.

Os Dados

Toda a informação que é necessária para executar o programa que lhe é solicitado, será fornecida através dos argumentos do predicado usado para o activar, como se detalha a seguir.

O programa deve ser invocado com o predicado

```
cobertura(NLinhas,NColunas,Lista_Buracos)
```

em que `NLinhas` indica o número de linhas do tabuleiro, `NColunas` o número de colunas do tabuleiro e `Lista_Buracos` é uma lista com as coordenadas `b(L,C)` dos buracos do tabuleiro.

Os Resultados

O predicado `cobertura/3` deve ser executado com sucesso se for possível cobrir perfeitamente o tabuleiro com pedras de dominó e deve falhar se tal não for possível.

Exemplos: Dado que é possível cobrir perfeitamente um tabuleiro de `2x2` que tem 0 buracos (`[]`), `cobertura(2,2,[])` deve ser executado com sucesso:

```
?-cobertura(2,2,[]).
  yes
```

Uma vez que um tabuleiro de `2x2` com um buraco em `1x1`, ou seja, ao qual foi retirado um canto, não pode ser coberto perfeitamente, `cobertura(2,2,[b(1,1)])` deve falhar:

```
?-cobertura(2,2,[b(1,1)]).
  no
```