

Concurso/Encontro Nacional de Programação em Lógica CeNPL'02

Universidade de Coimbra / Instituto Politécnico de Coimbra

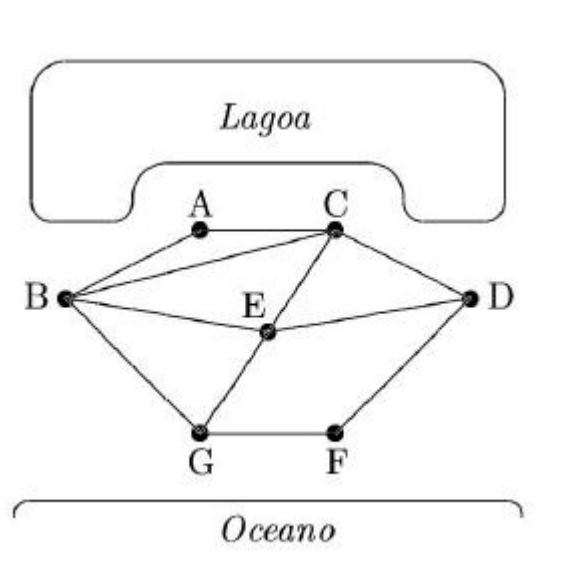
11-13 de Abril de 2002

Problema nº 4

DETECTIVES¹

Introdução

O local, uma estância de verão, tinha um aspecto tão agradável que seria impossível imaginá-la como palco de um crime de odor tão sórdido. A estância era composta por sete *bungalows*, quatro junto à lagoa (A, B, C, D), dois junto ao oceano (F e G) e um no meio (E), ligados por caminhos de acordo com a figura:



Um empregado viu um homem de aspecto sinistro, carregando um balde grande, aproximar-se da estância vindo da lagoa e esconder-se dentro de um dos *bungalows* próximos desta. O homem percorreu os caminhos da estância deixando rastros de peixe podre por todo o lado.

A polícia, chamada ao local, detectou pelas pegadas lamacentas deixadas pelo criminoso, que este tinha percorrido todos os caminhos da estância exactamente uma vez. Os detectives repararam também que não havia pegadas que conduzissem para fora da estância, concluindo por isso que o infractor se devia ainda encontrar escondido dentro de um dos *bungalows*. Infelizmente, as pegadas eram tão pouco nítidas que os detectives foram incapazes de determinar a direcção destas. Para piorar ainda mais as coisas, o empregado da estância que tinha avistado o infractor não se lembrava do *bungalow* em que este tinha entrado. A polícia ficou impossibilitada de reproduzir o caminho do infractor. O máximo que conseguiu apurar foi que ele nunca percorreu o mesmo caminho duas vezes.

Tarefa

Escrever em Prolog um programa para auxiliar a polícia a descobrir em que *bungalow* o infractor se encontra escondido. O programa deverá poder ser aplicado em estâncias com topologias diferentes da do exemplo, mas mantendo sempre o requisito de cada caminho ser percorrido exactamente uma vez.

Os Dados

O programa deverá receber como argumentos a lista dos possíveis pontos de partida (no caso do exemplo, a lista [a, b, c, d]) e a topologia da estância, representada por uma lista de pares correspondentes aos vários caminhos (no caso do exemplo, [(a,b), (a,c), (b,c), (b,e), (b,g), (c,e), (c,d), (d,e), (d,f), (e,g), (f,g)]).

Os Resultados

O programa deverá ser chamado através do predicado `detective/2` que mostrará no ecrã o par (ou pares, caso exista mais do que uma solução) de *bungalows* em que o primeiro elemento do par é o *bungalow* de onde o infractor partiu e o segundo elemento do par é o *bungalow* onde o infractor está escondido.

Exemplos:

```
?- detective([a,b,c,d],
            [(a,b), (a,c), (b,c), (b,e), (b,g),
             (c,e), (c,d), (d,e), (d,f), (e,g), (f,g)]).
[(d, g)]
```

```
?- detective([a,b], [(a,b), (a,c), (b,c)]).
[(a, a), (b, b)]
```

¹ Inspirado no problema "Something Fishy" da revista Scientific American de Maio de 2001.