Primeiro Trabalho de Técnicas Alternativas de Programação (Prof. Alexandre Direne - 2010/1)

Atenção:

Este trabalho é obrigatório e deverá ser entregue, impreterivelmente, até o dia 03 de maio de 2010 (segunda-feira). A solução é individual e deverá ser arquivada no diretório "~alexd/TAP/" onde o nome do arquivo terá como prefixo o seu nome-de-usuário no sistema do laboratório e, como extensão, ".pl" para indicar que seu conteúdo possui um programa em Prolog. Assim, por exemplo, se o seu nome de usuário no sistema fosse "grs07" então o nome o arquivo seria "grs07.pl" (dentro do diretório "~alexd/TAP/"). Não se esqueça de proteger completamente o arquivo criado, de maneira a permitir a leitura do mesmo apenas por você! Isso pode ser feito aplicando `chmod og-rwx grs07.pl` antes de efetuar a cópia com a preservação das permissões (`cp -p grs07.pl ~alexd/TAP/`). Não se preocupe com as permissões do professor que irá corrigir o trabalho. A correção dos trabalhos será parcialmente automatizada, sendo assim, é importante que todos os arquivos com as soluções individuais estejam no diretório citado acima, dentro do prazo estipulado. Não será permitida a entrega do arquivo por e-mail.

Enunciado:

A Figura 1 mostra o exemplo de um grafo que representa vias terrestres e aéreas de mão dupla com as respectivas indicações do custo da viagem de uma cidade para outra. As linhas contínuas indicam vias terrestres e as pontilhadas, vias aéreas. As vias terrestres ligam todas as cidades de um mesmo país (cujo nome não interessa para este problema). Cada país só possui uma cidade com aeroporto. No exemplo do grafo da Figura 1, os aeroportos de cada país são: `e`,`j` e `m`.

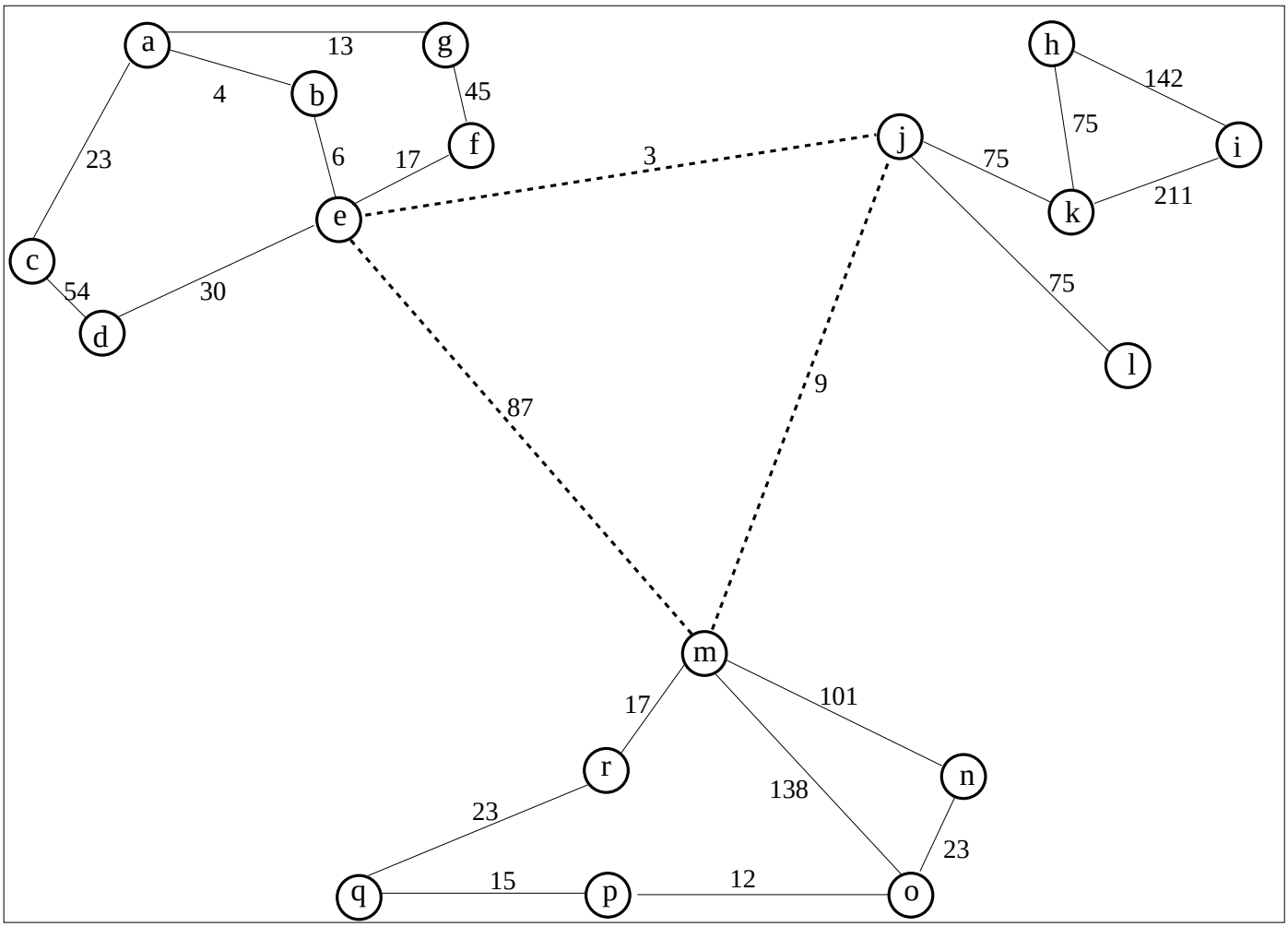

Figura 1: Grafo com vias terrestres e aéreas e seus custos de percurso.

Implementar um predicado ternário em Prolog que relaciona duas cidades de países diferentes com o caminho entre elas. Tal caminho é definido como a sequência das cidades que compõem o trajeto, obedecendo às duas restrições descritas ao longo deste enunciado (atenção para elas!). Para ilustrar uma aplicação do predicado "`caminho`", a consulta a seguir mostra o cômputo do trajeto entre a cidade g e a cidade p para o grafo da Figura 1.

```
?- caminho(g, p, Caminho).
Caminho = [g, a, b, e, m, r, q, p] ?
yes
```

Duas restrições importantes se aplicam a um caminho qualquer (o qual é sempre entre duas cidades de países diferentes):

1. a cobertura das vias terrestres deve ser feita pelo caminho de custo mínimo (dica: use o Algoritmo de Dijkstra para atingir isso de maneira segura pois o custo de percurso de qualquer via é sempre positivo);
2. o caminho completo só pode incluir um único vôo (dica: use o predicado "`subculturas`" para ajudar a separar as cidades de um mesmo país).

Para evitar erros e ainda servir como apoio inicial, o código em Prolog abaixo é uma boa representação do grafo da Figura 1.

```
via_terrestre(a,b,4).
via_terrestre(b,e,6).
via_terrestre(a,c,23).
via_terrestre(a,g,13).
via_terrestre(c,d,54).
via_terrestre(d,e,30).
via_terrestre(e,f,17).
via_terrestre(f,g,45).
via_terrestre(j,k,75).
via_terrestre(j,l,75).
via_terrestre(k,h,75).
via_terrestre(k,i,211).
via_terrestre(i,h,142).
via_terrestre(m,n,101).
via_terrestre(m,o,138).
via_terrestre(m,r,17).
via_terrestre(n,o,23).
via_terrestre(r,q,23).
via_terrestre(q,p,15).
via_terrestre(p,o,12).
via_aerea(e, j, 3).
via_aerea(e, m, 87).
via_aerea(j, m, 9).
```

Obsevações finais:

- Indique, por meio de um comentário de código, qual foi o compilador Prolog que você utilizou (e.g., SWI-Prolog, Poplog-Prolog, GNU-Prolog, etc);
- Não troque o nome do predicado "caminho" pois isto dificultará a correção do trabalho.