

# Trabalho Prático

PPIII (LESI)

2002/2003

## Introdução

Este documento apresenta o trabalho prático da disciplina de Paradigmas da Programação III. **Leia-o com atenção** e até ao fim antes de começar a realizar o trabalho.

O enunciado do trabalho procura ser o mais explicativo possível, por forma a facilitar a interpretação do mesmo. De qualquer modo, sempre que sentir que na modelação da solução que pretende apresentar tem necessidade de tomar decisões sobre pontos que o enunciado não prevê, é livre de o fazer desde que justifique convenientemente no relatório, e na apresentação do trabalho, tais tomadas de decisão.

O trabalho foi concebido para ser realizado no espaço de **um mês**. A data de entrega do trabalho será oportunamente afixada na página da disciplina.

Cada grupo de trabalho deve ser constituído por **três elementos**. Serão colocadas na recepção do DI, folhas com a marcação de hora para entrega do trabalho prático, pelo que depois de estas serem retiradas **não serão aceites** entregas não programadas. Para evitar a possibilidade de este tipo de problemas ocorrer consulte regularmente a página da disciplina.

## Enunciado

A empresa de transportes VQC realiza transporte de mercadorias entre diversos destinos num conjunto de rotas pré-definidas. Actualmente opera num sistema de escalonamento fixo de viagens (entre cada destino existe um número pré-definido de viagens em dias e horas estabelecidos). Este sistema acarreta custos elevados pois muitas vezes as viagens são feitas sem a carga completa.

Noutras situações, existe carga em excesso, pelo que alguma tem que esperar pela viagem seguinte. A empresa compromete-se a não cobrar o serviço sempre que a carga não seja entregue no prazo previsto.

**Para melhorar** esta situação a VQC pretende implementar um sistema flexível de gestão da frota e das viagens. No novo sistema o tipo de transporte e a altura para a viagem é definido em função da quantidade de carga disponível para transporte em cada momento.

Importa dizer que a VQC tem uma frota constituída por veículos com diferentes capacidades de carga, que tentará alocar de acordo com as solicitações em cada uma das localidades onde actua. A VQC, com o intuito de reduzir os custos de funcionamento, está a ponderar desenvolver um programa que lhe permita melhorar a sua gestão.

A sua missão, caso decida aceitá-la, é construir um programa Prolog que permita aos gestores da empresa melhorar o rendimento global de VQC, disponibilizando-lhes sugestões de utilização dos transportes em função da carga a transportar.

## Instruções

Para o ajudar na análise do problema apontam-se como principais entidades as seguintes:

- **Localidade** - genericamente com informação sobre o nome da localidade e um código pela qual é designada. A localidade também deve conter informação sobre os veículos que em determinado momento lá estão localizados;
- **Ligação** - contém informação sobre a localidade origem e destino da ligação. Uma ligação contém informação sobre o tráfego automóvel num sentido, isto é, o facto de existir ligação entre A e B, não implica a existência de caminho directo entre B e A (embora este seja o cenário mais provável). Além da informação das localidades, a ligação contém também informação sobre a distância entre as localidades e o tempo que em média a viagem demora;
- **Veículo** - agrega informação sobre a capacidade de carga que um determinado veículo pode transportar. Quando em utilização, um veículo guarda também dados sobre a carga que está a transportar;
- **Carga** - esta entidade regista a informação sobre os items que se querem transportar de dada origem para um destino. É uma lista constituída

pela informação sobre a origem e o destino, peso e data limite de entrega no destino de cada item.

O transporte de um determinado item de um sítio para outro implica um custo de operação para o transportador. Este tenta minimizar ao máximo esse custo e simplisticamente podemos dizer que o lucro do transportador é função do custo de operação. Para o cliente o valor que paga pelo transporte de um determinado bem de uma cidade para outra, é estabelecido caso a caso, em função da distância, mas também da própria natureza do bem (tem um valor diferente transportar 1 Ton. de Ouro ou 1 Ton. de Tecido). Neste programa não queremos dar suporte a este tipo de raciocínio (se o quiser fazer, será por sua conta e risco!).

Para o transportador (entidade para a qual estaremos a trabalhar) o custo de operação incorrido durante a viagem de um veículo é calculado pela fórmula:

$$CO = Distancia * Tempo * (0.1 * Tara + 0.4 * PesoTransportado)$$

Ao elaborar uma possível solução, tenha em atenção que ao calcular a rota de um camião, pode ter que tomar a decisão de efectuar a ligação para uma determinada carga, ou então esperar por outro camião, que vá para o mesmo destino, optimizando assim os recursos.

É objectivo que o seu programa estabeleça um conjunto de rotas para a melhor distribuição possível das cargas, para um determinado dia. Embora seja discutível o que se entende por melhor distribuição possível, neste contexto queremos cingir-nos ao conjunto de rotas que permita satisfazer os pedidos, com o menor custo de operação possível. Para tal deverá calcular as soluções possíveis e dentre elas escolher a mais rentável.

## Exemplo

Por forma a ser mais fácil a compreensão do enunciado, junta-se uma modelação exemplo (que constitui uma boa base para a resolução do problema) e um exemplo de cálculo de rotas.

A solução que desenvolver deverá funcionar com a Base de Conhecimento apresentada neste exemplo. Poderá inclusivé utilizar este exemplo para testar a solução que implementar (mas utilize também exemplos maiores).

A figura 1 apresenta um mapa de várias cidades onde a VQC tem escritórios. Apresenta-se de seguida uma base de conhecimento para este mapa. Para além da informação que se pode inferir da figura (localidades e ligações entre elas), sabe-se que existe uma viatura de 3Ton. em Braga, uma de

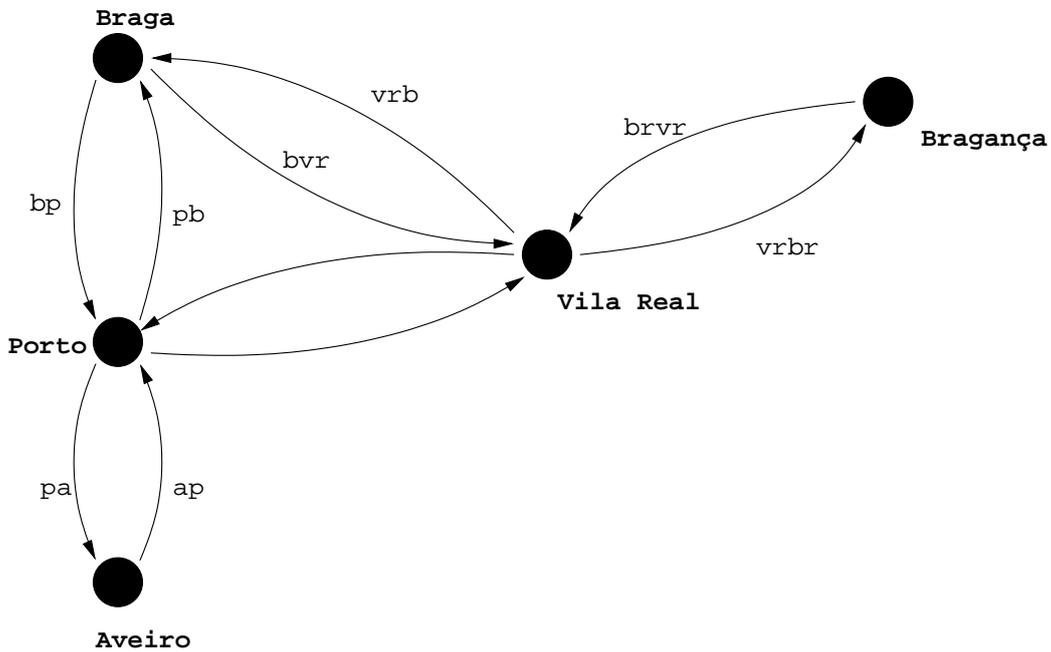


Figura 1: Mapa exemplo.

2Ton. em Aveiro e uma de 3Ton. em Vila Real. São também conhecidas as distâncias e os tempos médios de duração das deslocações entre as diferentes localidades.

A informação relativa às localidades é definida com factos `localidade/3`:

```
%localidade(Cod,Descr,ListaVeículos) :- A localidade com código
%                                     Cod tem descrição Descr
%                                     e estão lá estacionadas
%                                     as viaturas da lista
%                                     ListaVeículos.
%
```

```
localidade(av, 'Aveiro', [v1]).
localidade(p, 'Porto', []).
localidade(b, 'Braga', [v2]).
localidade(vr, 'Vila Real', [v3]).
localidade(br, 'Bragança', []).
```

a informação relativa às ligações entre cidades com factos `ligação/4`:

```
%ligação(Orig, Dest, Dist, Temp) :- Existe uma ligação entre
%                                     Orig e Dest que tem a
%                                     extensão de Dist (em Km) e
```

```
% demora Temp (em Horas).
```

```
ligação(av, p, 70, 1).  
ligação(p, av, 70, 1).  
ligação(b, p, 60, 1).  
ligação(p, b, 60, 1).  
ligação(b, vr, 120, 4).  
ligação(vr, b, 120, 4).  
ligação(p, vr, 100, 2).  
ligação(vr, p, 100, 2).  
ligação(vr, br, 100, 2).  
ligação(br, vr, 100, 2).
```

É agora necessário representar a informação relativa aos veículos:

```
%veículo(Cod, T):- o veículo de código Cod tem capacidade para  
% transportar T toneladas.
```

```
veículo(v1, 2).  
veículo(v2, 3).  
veículo(v3, 3).
```

Finalmente é necessário representar quais as cargas que devem ser despachadas:

```
%carga(Cod, Orig, Dest, Peso, D):- a carga com código Cod deve  
% ser enviada de Orig para  
% Dest, tem peso Peso e deve  
% ser entregue no prazo máximo  
% de D dias.
```

```
carga(c1, b, vr, 1, 2).  
carga(c2, av, br, 2, 1).
```

É agora necessário escrever um predicado que determine qual a melhor utilização a dar aos veículos existentes de modo a satisfazer os pedidos dos clientes com o mínimo custo.

A solução para o problema deverá ser encontrada sob a forma de uma lista de viagens, sendo cada viagem representada por um termo *viagem/2* tal que *viagem(Cod, L)* indica que o veículo *Cod* fez a viagem descrita em *L*. Por sua vez *L* é uma lista de termos *etapa/3* tal que *etapa(Orig, Dest, Lcarga)* indica que o veículo efectuou a ligação entre *Orig* e *Dest* com a carga indicada na lista *Lcarga*. Assim, o termo:

```
viagem(v2, [etapa(b, p, [c1]), etapa(p, vr, [c1,c2])])
```

indica que o veículo v2 viajou de Braga até ao Porto transportando a carga c1 e daí até Vila Real transportando as cargas c1 e c2.

Uma primeira possibilidade de resolução seria escrever um predicado `planosimples/2` que encontrasse a melhor atribuição de viagens aos veículos existentes sem considerar partilha de cargas, calculando também o custo dessa distribuição.

No caso presente esse predicado deveria suceder para a seguinte *query*:

```
planosimples([viagem(v1, [etapa(a, p, [c2]),
                        etapa(p, vr, [c2]),
                        etapa(vr, br, [c2])]),
             viagem(v2, [etapa(b, p, [c1]),
                        etapa(p, vr, [c1])])], 568).
```

No entanto, esta não é, ainda, a melhor solução. Na verdade, tanto o veículo v1, como o veículo v2 estão a realizar a viagem do Porto para Vila Real, quando bastaria que apenas o veículo v2 o fizesse (levando a sua carga original mais a carga que v1 transportava).

Para ser aceitável pela empresa o sistema deverá detectar estas possibilidades de partilha de carga e pô-las em prática. É então necessário definir um predicado `plano/2` que deverá suceder para a *query*:

```
plano([viagem(v1, [etapa(a, p, [c2])]),
      viagem(v2, [etapa(b, p, [c1]),
                  etapa(p, vr, [c1, c2]),
                  etapa(vr, br, [c2])])], 548).
```

o que representa uma poupança de 20.

Note-se que esta solução assume que é sempre possível combinar as cargas dos veículos. Numa situação 100% real isso estará dependente das horas de partida e chegada de cada veículo. O tratamento dessa informação é uma melhoria que poderá fazer no trabalho.

Após se ter identificado um plano de viagens, torna-se necessário registá-lo no sistema e fazer as alterações necessárias sobre a localização dos veículos e das cargas. Para isso, deverá ser definido o predicado `executa/1`, que dado uma lista de viagens efectua na base de conhecimento as alterações por elas produzidas.

## Avaliação

A avaliação do trabalho terá três componentes:

- o programa desenvolvido;
- o relatório do trabalho;
- a apresentação/discussão do trabalho.

Relativamente ao programa, será tida em conta a sofisticação dos predicados `plano/2` e `executa/1`. Se não for considerada a partilha de cargas, a nota final poderá ir até 12 valores. Com a partilha de cargas, a nota poderá ir até 17 valores. Para obter notas superiores será necessário ter considerado a questão temporal.

A Base de Conhecimento do programa deverá, **obrigatoriamente**, ser definida utilizando factos `localidade/3`, `ligação/4`, `veículo/2` e `carga/5`; os predicados `plano/2` e `executa/1` deverão ser definidos. Outros factos e/ou regras poderão ser definidos conforme necessário.

O relatório deverá descrever (em traços gerais) o problema e a solução proposta, descrever/justificar as opções tomadas durante o desenvolvimento dessa solução e indicar quais as vantagens/desvantagens e/ou problemas encontrados na utilização do Prolog para a resolução do trabalho. A capa do trabalho deverá indicar o nome da cadeira e do curso, o nome dos elementos do grupo e o ano lectivo (2002/2003).

## Notas Finais

**Tenha em atenção** que deve efectuar o trabalho por etapas de modo a que não lhe aconteça chegar ao fim do prazo sem que tenha uma versão a funcionar.

Para tal **sugere-se** que o tratamento de partilhas pode ser efectuado com base na solução inicial sem tratamento de partilhas.

Como sempre acontece quando se começa a utilizar um novo paradigma de programação, existe uma fase inicial de adaptação que poderá durar mais ou menos tempo. Por forma a evitar problemas, comece a resolver o trabalho o mais cedo possível. Dessa forma terá mais margem de manobra caso a fase de adaptação se revele mais longa.

**Bom trabalho**