

# Concurso/Encontro Nacional de Programação Lógica CeNPL'2004

Departamento de Matemática  
Instituto Superior Técnico

16–18 de Abril de 2004

## DECISÃO IMPOSSÍVEL

### Introdução

No contexto do desenvolvimento de um sistema informático para apoio ao trabalho colaborativo de uma equipe de decisores distribuída, pretende-se implementar uma funcionalidade para *suporte à decisão*.

Essa funcionalidade permite fazer a escolha de uma situação entre um conjunto de várias alternativas com base na avaliação que os decisores (os gestores, membros de uma equipa directiva, ou selectiva) dão a cada um dos parâmetros que permitem comparar as ofertas alternativas, parâmetros esses que poderão ter pesos diferentes (de acordo com um critério estabelecido pelo Gestor Principal do processo).

A título de exemplo, suponha que se quer escolher os telemóveis a comprar para os directores, secretárias e funcionários de marketing de uma determinada empresa. Essa decisão vai ser da responsabilidade distribuída dos 5 sócios fundadores da empresa, os quais escolhem como critérios, ou parâmetros, de avaliação: a *companhia operadora* de telecomunicações móveis; a *marca e o modelo* do equipamento proposto; o *prazo de entrega*; o *custo das chamadas para a rede*; o *custo das chamadas para as outras redes móveis*; o *custo das chamadas para a rede fixa*; a existência de algumas *facilidades* para uma rede de empresa; e o *plano de manutenção e apoio* ao cliente.

Voltando ao caso geral, o produto escolhido será o que reunir maior pontuação (somatório da classificação atribuída por cada membro da equipe decisora). A classificação dada por cada decisor obtém-se somando os pontos atribuídos a cada parâmetro (numa escala de 0 a 4) depois de multiplicados pelo respectivo peso.

### Tarefa

A sua tarefa consiste em escrever um programa Prolog que, dado um *cenário de escolha*—o qual estabelece as alternativas (associando a cada uma um identificador), o número de parâmetros de avaliação e o peso de cada um—e as *avaliações de cada decisor* — não-satisfaz (ns), satisfaz-mínimo(sm), satisfaz (s), bom (b) e muito-bom (mb)

— para cada alternativa, determina a alternativa (o produto) a escolher de acordo com o processo de decisão exposto acima.

## Os Dados

Os dados para este problema devem ser carregados a partir do ficheiro 'escolha.pl', o qual define os predicados `cenario/2` e `avaliacao/3` através de factos da forma:

```
cenario( Alts, Pesos ).
avaliacao( Dec, Alt, Classifs ).
```

em que: `Alts` é a lista de alternativas (identificadores de casos a comparar); `Pesos` é a lista com os pesos a atribuir a cada parâmetro no cálculo da pontuação (note que o número de parâmetros a usar no processo de decisão está implícito nesta lista); `Dec` é o nome do decisor; `Alt` é o nome da alternativa votada (suponha que foi previamente validade e portanto existirá sempre na lista `Alts`); `Classifs` é a lista de avaliações atribuídos por esse decisor a cada um dos parâmetros da alternativa em causa (também neste caso, deve supor que houve uma validação prévia e, portanto, cada classificação indicada será sempre um valor correcto dentro dos 5 possíveis, acima listados).

## Os Resultados

O seu programa deve ser invocado através do predicado `escolhe/2`:

```
escolhe( -Decisao,-Erros ).
```

Ao ser invocado, o programa instancia a variável `Decisao` com um par de valores na forma do functor `decisao(Alt,Pontos)` em que a primeira variável trará o identificador da alternativa mais bem pontuada e a segunda corresponderá à pontuação dessa escolha.

Caso haja mais de uma com a mesma pontuação máxima, a variável `Decisao` indicará 'conflito'.

Além disso, o programa a desenvolver deve rejeitar (desprezar no cálculo da pontuação) as avaliações erradas, reportando as anomalias através da variável `Erros`, uma lista com os seguintes funtores: `classifMaisMenos(Dec,Alt,N)` – na alternativa `Alt` o decisor `Dec` classificou `N` parâmetros (acima ou abaixo do número esperado); `votouRepetido(Dec,Alt)` o decisor `Dec` votou a votar na alternativa `Alt` (já votada); `naoVotou(Dec,Alt)` o decisor `Dec` não votou na alternativa `Alt`.

## Exemplos

Supondo que o ficheiro 'escolha.pl' continha o seguintes factos:

```
cenario( ['Vodafone-a', 'Vodafone-b', 'TMN-a', 'TMN-b', 'TMN-c'],
         [1,2,1,2] ).
avaliacao( prh, 'Vodafone-a', [b,b,s,mb] ).
avaliacao( prh, 'Vodafone-b', [mb,mb,b,mb] ).
avaliacao( prh, 'TMN-a', [b,b,b,b] ).
avaliacao( prh, 'TMN-b', [s,s,b,s] ).
avaliacao( prh, 'TMN-c', [sm,sm,ns,sm] ).
avaliacao( prh, 'TMN-c', [sm,sm,ns,sm] ).
avaliacao( jcr, 'Vodafone-a', [b,b,s,b] ).
avaliacao( jcr, 'Vodafone-b', [mb,mb,mb,mb] ).
avaliacao( jcr, 'TMN-b', [s,s,s] ).
avaliacao( jcr, 'TMN-c', [s,sm,ns,s] ).
```

Segue-se um exemplo ilustrativo do programa pretendido:

```
?- escolhe(D,E).
   D = decisao('Vodafone-b',47)
   E = [votouRepetido(prh, 'TMN-c'),
        classifMaisMenos(jcr, 'TMN-b', 3),
        naoVotou(jcr, 'TMN-a')].
```