

Projecto de Lógica para Programação  
**Os suspeitos do costume**  
Licenciatura em Engenharia Informática e de Computadores

Instituto Superior Técnico

2012-2013

## 1 Introdução

A Avó Einstein vive num bairro problemático, pelo que quase todos os dias é testemunha de um delito, passando grande parte da semana na esquadra, a identificar criminosos. No entanto, a Avó Einstein não gosta de soluções fáceis, pelo que sempre que vai identificar um criminoso, nunca o faz diretamente, apresentando um conjunto de pistas que levam, com algum raciocínio, à sua identificação. Ora o detective Horácio é o chefe dessa esquadra e costuma perder um tempo precioso a resolver os enigmas da Avó Einstein, pelo que pediu aos alunos do Técnico que implementassem um programa capaz de os resolver. Este projecto visa a implementação de um programa em Prolog que, com base num conjunto de pistas, permite responder ao desafio do detective Horácio



Figura 1: Os suspeitos do costume

Este enunciado está organizado como se segue: na secção 2 descrevem-se os suspeitos, na secção 3 definem-se os predicados a implementar, na secção 4 precisam-se alguns detalhes da implementação e, na secção 5, explica-se como deve ser feita a entrega e como vai ser avaliado o projecto.

## 2 Os suspeitos

Apenas para fins ilustrativos, define-se um conjunto específico de suspeitos, bem como os atributos que os caracterizam. Assim sendo, cada suspeito é definido pelo predicado `suspeito` de aridade 6 (isto é, com 6 argumentos):

```
suspeito(Nome, Idade, Arma, Forma, Objecto, Calçado)
```

Em que:

- `Nome` é o nome do suspeito (`Nome`  $\in$  {'Valdemort', 'Sauron', 'Darth Vader', 'Agent Smith', 'Hannibal Lecter', 'Malvina Cruela'});
- `Idade` é a idade do suspeito (`Idade`  $\in$  {50, 8000000, 150, 45, 78, 67});
- `Arma` é a arma favorita do suspeito (`Arma`  $\in$  {winchester, bazuca, fígua, varinha, catana, 'agulhas de tricot'});
- `Forma` representa a forma física do suspeito (`Forma`  $\in$  {deploravel, fracote, media, boa, 'muito boa', maravilhosa});
- `Objecto` é um objecto transportado pelo suspeito (`Objecto`  $\in$  {luva, capacete, mac, anel, 'escova de dentes', livro});
- `Calçado` é o tipo de calçado do suspeito (`Calçado`  $\in$  {botas, sandalias, pantufas, havaianas, 'saltos altos', tenis}).

Pode assumir como fixo o número de argumentos do predicado `suspeito`, bem como o número de suspeitos (6 em ambos os casos). Pode igualmente assumir que existem exactamente 6 valores diferentes para cada campo. No entanto, tem de ser claro que o código a implementar **não** deve estar preso aos valores em si, dado que nada impede que lhe seja apresentado um teste com um suspeito chamado, por exemplo, 'Freddy Krueger').

## 3 As pistas e o desafio do detective Horácio

### 3.1 As pistas

Considerando que a sequência de suspeitos é representada através de uma lista em Prolog (denotada por `Suspeitos`), ocupando o suspeito mais à esquerda a posição 1 da lista, isto é, a sua cabeça, o primeiro tipo de pista indica a posição ocupada por um suspeito específico nessa lista. Assim, a pista que se segue indica que o suspeito *Darth Vader* tem uma *luva* e que se encontra na posição 3 na sequência de suspeitos:

```
Suspeitos=[_, _, suspeito('Darth Vader',_,_,_, luva,_), _, _, _]
```

O segundo tipo de pistas serve para indicar características específicas de um dado suspeito, não sendo nada dito em relação ao lugar que ocupa na fila.

Por exemplo, para especificar que o suspeito que tem a *catana* calça *sandálias*, e que o suspeito que tem 45 anos está numa forma física *deplorável*, há que escrever, respectivamente:

```
existe(suspeito(_, _, catana, _, _, sandalias), Suspeitos)
existe(suspeito(_, 45, _, deploravel, _, _), Suspeitos)
```

As restantes pistas também permitem especificar características dos suspeitos, mas servem especialmente para descrever as relações entre as posições ocupadas pelos suspeitos. São estas:

- **lado(s1, s2, Suspeitos)**: indica que os suspeitos s1 e s2 estão ao lado um do outro (nada é dito sobre o que está à esquerda/direita), na lista **Suspeitos**;
- **entre(s1, s2, s3, Suspeitos)**: indica que um suspeito s2 está entre os suspeitos s1 e s3 na lista **Suspeitos**. Note que nada é dito sobre o que está à esquerda/direita, mas pode-se inferir que os três suspeitos estão seguidos (s1 s2 s3 ou s3 s2 s1);
- **esquerda(s1, s2, Suspeitos)**: indica que o suspeito s1 está à esquerda do suspeito s2 (não necessariamente na posição imediatamente a seguir);
- **direita(s1, s2, Suspeitos)**: indica que o suspeito s1 está à direita do suspeito s2 (não necessariamente na posição imediatamente a seguir);
- **naoEntre(s1, s2, s3, Suspeitos)**: indica que o suspeito s2 não está entre os suspeitos s1 e s3. Note que nada é dito sobre o que está à esquerda/direita. No entanto, pode inferir que entre s1 e s3 há apenas uma posição;
- **ou(s1, s2, s3, Suspeitos)**: indica que o suspeito s1 é “o mesmo” (ver exemplo abaixo) que s2 ou que s3, mas não os dois.

Para melhor ilustrar estes conceitos, seguem-se alguns exemplos. Para indicar que o suspeito com a *catana* está ao lado do suspeito com *saltos altos*, escreve-se:

```
lado(suspeito(_, _, catana, _, _, _),
     suspeito(_, _, _, _, _, 'saltos altos'), Suspeitos)
```

Do mesmo modo, para indicar que o suspeito que tem 78 anos está entre o suspeito que tem uma forma física *muito boa* e o suspeito que tem um *mac*, escreve-se:

```
entre(suspeito(_, _, _, 'muito boa', _, _),
      suspeito(_, 78, _, _, _, _),
      suspeito(_, _, _, _, mac, _), Suspeitos)
```

Para indicar que o suspeito com 50 anos tem um *anel* ou então um *mac*, mas não os dois, usa-se:

```
ou(suspeito(_, 50, _, _, _, _),
   suspeito(_, _, _, _, anel, _),
   suspeito(_, _, _, _, mac, _), Suspeitos)
```

Pode encontrar mais exemplos no ficheiro de testes (secção 4).

## 3.2 O desafio do detective Horácio

O detective Horácio quer saber tudo sobre cada cenário/desafio. Para obter informações relativas ao desafio N, usa-se o predicado:

```
desafio(N, Suspeitos)
```

A variável `Suspeitos` conterá uma solução: uma lista com 6 elementos, contendo cada elemento um predicado `suspeito/6`, tal como definido anteriormente, e cujos campos são os vários atributos que caracterizam os suspeitos. Por exemplo:

```
?- desafio(1, Suspeitos).
Suspeitos =
[suspeito('Valdemort', '800000', fiska, 'muito boa', capacete,
          sandalias),
 suspeito('Sauron', '78', 'agulhas de tricot', deploravel,
          'escova de dentes', 'saltos altos'),
 suspeito('Hannibal Lecter', '67', catana, maravilhosa, mac, tenis),
 suspeito('Darth Vader', '150', winchester, fracote, luva, pantufas),
 suspeito('Agent Smith', '50', varinha, boa, livro, botas),
 suspeito('Malvina Cruela', '45', bazuca, media, anel, havaianas)]
```

Note que se em algum desafio existir mais do que uma solução, deverá ser apresentada apenas uma (que será aceite desde que satisfaça as restrições do desafio).

## 4 Detalhes de implementação

### 4.1 Ferramenta

Recomenda-se o `swi-prolog`, dado que é o que vai ser usado na avaliação do projecto (em <http://www.swi-prolog.org/> encontrará ferramentas para as várias plataformas).

### 4.2 Testes e formato dos dados

Todo o código dos alunos deve estar no ficheiro *suspeitosDoCostume.pl*, onde deverão estar definidos os predicados relativos às pistas.

Está disponível um conjunto de ficheiros que permitem testar o projecto. O ficheiro *bateria\_testes.pl* contém 2 desafios e o ficheiro *resultadosTestes.txt* a sua solução. O ficheiro *testaProjecto.pl* contém o código abaixo.

```
testes :-
    % Carrega testes
    ['bateria_testes.pl'],
    ['suspeitosDoCostume.pl'],

    desafio(1, Suspeitos1),
    write('teste 1'), nl,
    ...
```

Para além de poder testar o projecto em ambiente Prolog, pode ainda usar a seguinte linha de comando<sup>1</sup> que permitirá guardar os resultados do programa no ficheiro *resultadosTestes.txt*:

```
swipl -s testaProjecto.pl -g testes -t halt > resultadosTestes.txt
```

## 5 Entrega e avaliação

O projecto deve ser entregue via Fénix e em papel, nos serviços apropriados, até às **15h do dia 20 de Maio de 2013**. A ter em conta:

- Entrega via Fénix: ficheiro num-grupo.zip (ex: 1.zip, 34.zip) – contendo:
  - o ficheiro *suspeitosDoCostume.pl* (com o código do projecto);
  - o relatório (formato .pdf) – máximo de 5 páginas e 10 000 caracteres (incluindo espaços). Será dado um modelo para o relatório.
- Entrega em papel do relatório e do código: folhas simplesmente agramadas (sem capa), na sala de estudo do DEI (alunos da Alameda) ou na portaria do edifício do IST-Taguspark (alunos do Tagus).
- Se a entrega (quer em papel, quer via Fénix) tiver até 24 horas de atraso serão retirados 2 valores à nota obtida no projecto; se a entrega tiver até 48 horas de atraso, o projecto será penalizado em 4 valores. Não serão aceites projectos a partir das 48 horas de atraso.

Será levada a cabo uma **avaliação automática** com vários desafios (12 valores); será igualmente tida em conta a **qualidade do código** (4 valores – uso adequado do Prolog, comentários, paragrafação, ausência de *warnings*, nomes de predicados e de variáveis cuidadosamente escolhidos, etc.); o **relatório** vale 4 valores e os parâmetros de avaliação são a sua organização, clareza e qualidade da escrita.

Os alunos devem ainda ter em conta que:

- Projectos que não respeitem os formatos descritos no enunciado terão 0 na avaliação automática.
- Caso se detectem cópias, os alunos terão 0 no projecto e não poderão fazer LP este semestre.
- Poderá haver uma discussão oral do projecto e/ou uma demonstração do seu funcionamento (a ser decidido caso a caso).

Bom trabalho e uma dica: *em Roma sê Romano*, ou seja, se vai trabalhar em Prolog, aproveite o que o Prolog tem para oferecer.

---

<sup>1</sup>Atenção que a chamada ao Prolog pode mudar de acordo com o sistema operativo.