

Paradigmas da Programação

MiECom (2º ano)

1º Teste

Data: 03 de Dezembro de 2009
Hora: 14:30H – 16:00H

Questão 1: Modelação de Sistemas de Informação

Considere que se pretende desenvolver um Sistema de Informação para gestão de uma oficina de automóveis, tendo como principal objectivo identificar Viaturas e seus Proprietários e a lista de peças gastas na manutenção de cada viatura (à qual vamos chamar Nota de Reparação, NR). Todas as peças consumidas saem do Stock da oficina, no qual cada peça é identificada por uma referência única, um nome e descrição, tendo associado o preço unitário, a quantidade em armazém e o nome do fornecedor. As viaturas são identificadas pela matrícula, devendo registar-se, entre outros, marca/modelo e kilometragem; enquanto que os proprietários serão identificados pelo BI, registando-se no mínimo o nome, morada e contacto. Cada NR será também identificada por um número de ordem e, além de cada peça gasta e quantidade usada, será associada a uma data de reparação e ao nome do funcionário responsável pelo serviço.

Usando a abordagem seguida nas aulas para modelar sistemas de informação, construa o modelo UML (Cenários-de-Uso e respectivos Diagramas Actores/Operações e o Diagrama de Classes) e depois construa a Base de Conhecimento (BC) que descreve esse modelo em lógica, recorrendo à linguagem de programação Prolog para poder interrogar de seguida o seu universo de discurso.

Após modelar em UML o SI descrito e implementar em Prolog o dito modelo, escreva questões Prolog para:

- saber todas as viaturas (marca/modelo) de um proprietário (dado o seu nome).
- listar todas as peças (nome, descrição e quantidade em stock) vendidas por um dado fornecedor.
- listar todas as peças (referência e nome) gastas na reparação (realizada numa dada data) de uma viatura (dada a sua matrícula).
- saber a matrícula de todas as viaturas (e o BI e nome do seu proprietário) que consumiram uma peça de referência dada.

Questão 2: Bases de Conhecimento

Analise com atenção a Base de Conhecimento (BC) em Prolog a seguir listada.

```
eUm( pescada,peixe ).      ondeVive( pescada,mar ).
eUm( marmota,peixe ).    ondeVive( marmota,mar ).
eUm( atum,peixe ).      ondeVive( atum,mar ).
eUm( salmão,peixe ).    ondeVive( salmão,rio ).
eUm( truta,peixe ).     ondeVive( truta,rio ).
eUm( boga,peixe ).      ondeVive( boga,rio ).
eUm( cobra,réptil ).    ondeVive( cobra,campo ).
eUm( galinha,ave ).     ondeVive( galinha,campo ).
eUm( cabra,mamifero ).   ondeVive( cabra,serra ).
eUm( peru,ave ).        ondeVive( peru,campo ).
eUm( águia,ave ).       ondeVive( águia,campo ).
```

```
desporto( manuel,pesca ).   ondeEsta( manuel,mar ).
desporto( jose,caça ).     ondeEsta( jose,serra ).
desporto( antónio,pesca ). ondeEsta( antónio,rio ).
desporto( alex,pesca ).    ondeEsta( manuel,mar ).
desporto( joão,caça ).     ondeEsta( joão,campo ).
```

```
podePescar( Pes, Peixe ) :- desporto( Pes, pesca ),
                             ondeEsta( Pes, Loc),
                             ondeVive( Peixe, Loc ),
                             eUm( Peixe, peixe).
```

Face à BC acima, responda às seguintes alíneas:

- a) Como construa uma questão para saber todos os peixes do mar ?
 b) Quais as respostas calculadas pelo interpretador de Prolog à questão abaixo (indique a ordem correcta):

```
?- eUm( X,ave ).
```

- c) Diga o que pode pescar o 'antónio' à luz da regra acima.
 d) Qual a resposta calculada pelo interpretador de Prolog à questão abaixo (justique a resposta):

```
?- podePescar( joão,X ).
```

- e) Qual a resposta calculada pelo interpretador de Prolog à questão abaixo (justifique a resposta):

```
?- podePescar( manuel,boga ).
```

- f) Explique claramente o que sucedia se redefinisse o predicado `podePesar/2` como se mostra abaixo; indique se sua resposta a alguma das três alíneas anteriores ficava alterada.

```
podePescar( Pes, Peixe ) :- desporto( Pes, pesca ),
                             ondeEsta( Pes, Loc),
                             eUm( Peixe, peixe),
                             ondeVive( Peixe, Loc ).
```

Questão 3: Manuseamento de Listas

Sobre operações com Listas em Prolog, responda às alíneas seguintes:

- a) Dada uma lista e um elemento, escreva um predicado para contar o número de vezes que o elemento ocorre na lista. Exemplo:

```
?- contar([1,3,4,5,1,7,2,1,4],1,N)
   N = 3.
```

- b) Escreva um predicado que dadas duas listas ordenadas e com elementos distintos, deriva uma lista que é a intersecção das listas dadas. Exemplo:

```
?- intersect([1,3,4,7,9,12,33,89],[1,2,5,7,8,12,15],L).
   L = [1,7,12].
```

- c) Apresente um predicado que gera uma lista que é o "shift" N vezes de uma lista dada. Exemplo (para N=3):

```
?- shift([1,2,3,4,5,6,7],3,L)
   L = [4,5,6,7,1,2,3].
```