Paradigmas da Programação I
LECom (2º ano)

Exame de 1ª Época – 2ª Chamada

Data: 25 de Janeiro de 2005
Hora: 14:00

Dispõe de 2:30 horas para realizar este exame
Leia as questões com toda a atenção e responda com calma e clareza em folha convencional

Questão 1: Representação de Conhecimento

Suponha que lhe pedem para desenvolver o sistema automático (completo, com HW e SW) de armazenamento e entrega de roupa numa lavandaria a seco (tipo do que já existe na `5-a-Sec`), baseado num carril circular móvel onde se dependuram os vários cabides; esse carril desloca-se de modo a trazer para a frente (posição onde o empregado está colocado) o cabide com a peça que o cliente está a solicitar.
Usando a abordagem seguida nas aulas para modelar sistemas de informação em lógica, recorrendo à linguagem de programação Prolog para poder interrogar de seguida o seu universo de discurso, construa uma Base de Conhecimento (BC) que descreva o sistema de gestão pedido.
Para isso guarde informação sobre: os clientes; cada peça de roupa que é deixada para lavar ou engomar; e sobre a posição (número do cabide) onde a peça é colocada no carril rolante quando está pronta para entrega. Registe ainda informação sobre o carril rolante (número de cabides que suporta, número de andares que tem, etc.) e o número do cabide que no momento está à frente (pronto para entrega). Note que o deslocamento que o carril rolante tem de fazer é calculado tendo em consideração a posição actual em que está parado e a posição da peça que se quer entregar.

Após identificar o tipo de cláusulas (factos ou regras) que deve usar para modelar o sistema, indique algum tipo de perguntas que poderia ver respondidas com o seu programa lógico.

Questão 2: Bases de Conhecimento, Árvores de Prova e de Procura

Suponha que construiu uma BC com várias cláusulas (referentes a diversos predicados) que descrevem um jogo de palavras cruzadas (tabuleiro, questões, dicionário, etc) e que abaixo se apresenta apenas um excerto dessa BC com os seguintes factos e regras

```
atribuicao(1, b, vertical, pato). atribuicao(2, a, horizontal, gata).

dicionario(pato, 4).     dicionario(gata, 4).
dicionario(ostra, 5).    dicionario(rato, 4).
dicionario(tia, 3).

letra(gata, 1, g).       letra(rato, 1, r).
letra(gata, 2, a).       letra(rato, 2, a).
letra(gata, 3, t).       letra(rato, 3, t).
letra(gata, 4, a).       letra(rato, 4, o).
.............................
```

```
letra(pato, 3, t).       letra(ostra, 5, a).

cruzam( P1, P2, Pos )  :- letra( P1, Pos, L ), letra( P2, Pos, L ).
cruzaveis( P1, P2, L ) :- letra( P1, _, L ), letra( P2, _, L ).
```

Responda, então, às alíneas seguintes:

a) Usando uma *Árvore de Prova*, prove que o interpretador Prolog daria a seguinte resposta

```
?- dicionario( ostra, C ).
   C = 5
   yes
```

b) Mostre, recorrendo a uma *Árvore de Prova*, que uma possível resposta calculada pelo interpretador de Prolog à questão

```
cruzam( gata, P, 3 ).
```

era

```
 P = pato
```

c) Usando uma *Árvore de Procura*, mostre a primeira resposta calculada pelo interpretador de Prolog à seguinte questão

```
cruzam( gata, rato, Pos ).
```

d) Usando uma *Árvore de Procura*, mostre todas as respostas calculadas pelo interpretador de Prolog à seguinte questão

```
cruzaveis( gata, rato, L ).
```

e) Considere uma lista da forma `ListaDeLetras=[s/2, a/5, ···, letra/posicao]`.
Defina um predicado `satisfazpalavra(Palavra,ListaDeLetras)` que garanta que Palavra é uma palavra válida (existente no dicionário) que satisfaz a exigência de ter todas as letras contidas na ListaDeLetras na posição indicada.

f) Recorrendo ao predicado standard do Prolog `assertz/1`, que adiciona dinamicamente cláusulas a uma BC, escreva um predicado `acrescenta/1` que recebe como argumento uma palavra e junta à BC mais uma entrada no dicionário (isto é, um facto `dicionario/2`) caso essa entrada ainda não exista.
Por exemplo

```
acrescenta(pato).
```

não teria qualquer repercussão, mas já

```
acrescenta(anedota).
```

adicionaria à BC o facto

```
dicionario(anedota,7).
```

Questão 3: Manuseamento de Listas

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

a) Pretende-se que implemente um predicado `filtro/3` que, dada um valor de referência e uma lista de números inteiros, retorne outra lista só com os números múltiplos dessa referência (não se esqueça que pode usar o operador `mod/2` do Prolog). Exemplo:

```
?- filtro(4,[1,2,3,4,5,6,7,8,9,10],L).
   L = [4,8]
```

b) Pretende-se que implemente um predicado `inverte/2` que, dada uma lista, devolva a lista invertida (em que o último é o primeiro). Exemplo:

```
?- inverte([1,2,3],L).
   L = [3,2,1]
```

c) Pretende-se que implemente um predicado `deriva/3` que, dado o grau (expoente máximo) de um polinómio e o polinómio representado como uma lista de coeficientes, ordenados por ordem decrescente de expoente (do grau máximo dado até 0), calcule o polinómio derivada (note que termos não existentes serão representados pelo coeficiente 0). Exemplo:

```
?- soma( 5, [2,0,1,0,-7,-4], L ).
   L = [10,0,3,0,-7]
```

d) Pretende-se que implemente um predicado `soma/3` que, dados dois polinómios representados como listas de pares (coeficiente,expoente) ordenados por ordem decrescente de expoente, calcule o polinómio soma. Exemplo:

```
?- soma( [(2,5),(1,3),(-7,1),(-4,0)], [(3,4),(8,3),(-3,2),(3,1)], L ).
   L = [(2,5),(3,4),(9,3),(-3,2),(-4,1),(-4,0)]
```