

Paradigmas da Programação I

LECom (2º ano)

Exame de 2ª Época

Data: 15 de Fevereiro de 2006
Hora: 14:30 (Sala B3.35A)

Dispõe de 2:30 horas para realizar este exame

Questão 1: Bases de Conhecimento

Um grupo de estudantes da UM, designado por EUMAN – *Estudantes da UM Amigos da Natureza*, dedica-se a todo o tipo de actividades ligadas com a preservação da fauna e flora das zonas naturais (parques nacionais). Dentro do EUMAN criaram um sub-grupo para protecção dos animais em extinção, como o Tigre da Malcata ou os Urso rosa da Peneda-Gerês.

Considere, então, a base de conhecimento (BC) que suporta a actividade do referido grupo, da qual se lista abaixo apenas um extracto.

```
habita(urso, geres).          habita(cavalo, geres).
habita(javali, geres).       habita(cavalo, peneda).
habita(javali, malcata).     habita(tigre, malcata).
habita(javali, soajo).       habita(cao, malcata).
densidade(urso, geres, 3).   densidade(cavalo, geres, 8).
densidade(javali, geres, 30). densidade(cavalo, peneda, 30).
densidade(javali, malcata, 70). densidade(tigre, malcata, 2).
densidade(javali, soajo, 90). densidade(cao, malcata, 70).
especie(urso, mamifero).     especie(cao, mamifero).
especie(javali, mamifero).   especie(pica-pau, ave).
especie(cavalo, mamifero).   especie(truta, peixe).
especie(tigre, mamifero).   especie(safio, peixe).
feroz(urso).                feroz(javali).
feroz(tigre).               feroz(cavalo).
domestico(cao).              domestico(cavalo).

parqueNac(geres).           parqueNac(malcata).
parqueNac(peneda).         parqueNac(soajo).

aConservar( A,Z ) :- mamifero(A), habita(A,Z), densidade(A,Z,D), D < 10.
amigoHomem( A ) :- domestico(A).
contradiz( A ) :- especie(A,E1), especie(A,E2), E1 \= E2, !.
contradiz( A ) :- feroz(A), domestico(A).
```

Responda, então, às alíneas seguintes:

- a) Se colocar ao Interpretador de Prolog a questão `amigoHomem(X)`, qual será a resposta calculada, isto é, com que valor a variável X irá unificar? O Interpretador de Prolog só calcula um valor, ou descobre mais do que um (se sim, explique como/em que circunstâncias) ?

- b) Defina um predicado `habitat(A)` que escreve na consola todas as zonas habitadas pelo animal `A`.
- c) Justifique a seguinte afirmação: “*Sempre que se invoque o predicado `aConservar/2` a prova falha (a resposta é `No`) independentemente do tipo ou valor dos argumentos*”.
Como pode alterar a cláusula referente a esse predicado, ou a própria BC, de modo a corrigir a situação acima justificada, fazendo com que o Interpretador de Prolog passe a inferir correctamente quando é que um animal deve ser conservado numa determinada zona.
- d) Relativamente ao predicado `contradiz/1`, definido acima através de 2 cláusulas:
1. explique por palavras suas o seu significado.
 2. diga para que serve o operador `cut ("!")`, no fim do corpo da cláusula (se é que serve para alguma coisa).
 3. defina (através de 1 ou mais cláusulas) o predicado `remove` que elimina da BC as cláusulas contraditórias.
- e) Escreva um predicado `verifica/0` que assinala erro sempre o local habitado por um animal não é um parque nacional.
- f) Para representar na BC as zonas habitadas por cada animal, também podia ter recorrido ao uso de listas. Diga como seria alterada a BC acima, caso fizesse essa escolha e discuta vantagens/desvantagens de cada uma das representações referidas.

Questão 2: Manuseamento de Listas

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

- a) Pretende-se que implemente um predicado `multiplica/2` que, dada uma lista, devolva como segundo argumento uma nova lista em que cada número da primeira é multiplicado pelo distância a que está do fim da lista (o último está à distância zero e o primeiro à distância que é o número de elementos menos 1). Exemplo:

```
?- multiplica([1,2,3,4], C).
   C = [3,4,3,0]
```

- b) Pretende-se que implemente um predicado `procura/3` que, dada uma palavra e uma lista de palavras, retorne no terceiro argumento a posição em que essa palavra se encontra na lista (1 se for a primeira e 0 se não existir). Exemplo:

```
?- procura(quente, [bom,good,mau,bad,quente,hot,bom,nice], T).
   T = 5
```

- c) Pretende-se que implemente um predicado `derivada/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), calcule e devolva como terceiro argumento o polinómio derivada (note que termos não existentes serão representados pelo coeficiente 0). Exemplo:

```
?- derivada( 6, [4,2,0,-1,0,-7,14], L ).
   L = [24,10,0,-3,0,-7]
```

Questão 3: Representação de Conhecimento

Para apoio à Unidade de Arqueologia, vai ter de desenvolver um sistema integrado (HW/SW), SIAA, que os arqueólogos pretendem usar no trabalho de campo (quando vão escavar uma zona cuja prospecção indicou ser um sítio arqueológico interessante). O SIAA é suportado por um computador de média dimensão, onde reside a base de dados central, e por vários PDA's que os arqueólogos levam quando se deslocam para o campo e para os quais descarregam vistas dessa BD central. No contexto deste exame, estamos apenas preocupados com a informação que tem de ser armazenada e manipulada pelo sistema.

Assim pretende-se ter informação sobre: cada *sítio arqueológico* (identificação e descrição, tipo, localização, estado de exploração, responsável, etc.); cada *escavação* (sítio onde foi realizada, período, quem chefiou, equipa que nela trabalhou, objectivos, resultados); os achados levantados em cada escavação (tipo de objecto, local onde foi recuperado, estado, datação, etc.); os arqueólogos (identificação, instituição onde trabalham, equipas que integram, etc.).

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 SIAA.

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

Questão 4: Autómatos Deterministas Reactivos

Pretende-se desenvolver um autómato reactivo que modele o comportamento de uma *calculadora* muito simples, que efectua as 5 operações aritméticas elementares sobre números inteiros.

A *calculadora* aceita dígitos (de 0 a 9) que formam um *operando*, ou então um *operador* (+, -, *, /, ↑)¹ ou um sinal (= ou C). A sequência começará, normalmente, por um *operando*, mas aceita também no início o *operador unário* -. *Operadores* a seguir a um *operador* serão ignorados.

A *calculadora* é capaz de realizar várias operações, antes de apresentar o resultado, sem respeitar prioridades, isto é, realiza-as pela ordem em que aparecem. Após mostrar o resultado (quando lê o *senal* =), permite aproveitar esse resultado como *operando* para a próxima operação, o que acontece quando após o *senal* = recebe um *operador*; normalmente, após o *senal* =, lê o *senal* C e então apaga o resultado e volta ao início. O sinal C, lido após um dígito, tem como efeito apagar, apenas, esse último dígito.

Implemente, então, em Prolog esse autómato determinista reactivo que será identificado pelo nome 'cs' (*calculadora simples*); escreva apenas os predicados `automato/7` e `deltalfa/5` (não precisa de escrever o predicado `acciaoSem/1` para definir as acções semânticas).

Mostre, no fim, como podia usar o predicado genérico `processa/2` para verificar se uma determinada sequência de operandos e operadores (escolha uma concreta) é válida na linguagem da dita *calculadora simples*.

¹O quinto operador, ↑, representa a potência de expoente inteiro positivo.