Universidade do Minho
Departamento de Informática
Lic. Matemática e Ciências da Computação, 2º Ano
Paradigmas da Programação III

Exame de Recurso **18 de Setembro de 2000**

Duração: 3.00 h

Leia atentamente as questões propostas que se seguem. Tente ser claro e objectivo nas suas respostas. Justifique sempre as suas respostas.

1. Prove por resolução que $P \models D$, sendo

$$\begin{aligned} P = \{&D \leftarrow A \wedge C,\\ &(A \wedge C) \leftarrow E,\\ &E\}. \end{aligned}$$

2. Uma árvore binária pode ser representada pelo termo $\mathsf{tree(Nodo, Left, Right)}$ onde $\mathsf{Left, Right}$ são respectivamente as subárvores esquerda, direita. Escreva um programa em Prolog que dada uma árvore binária produza uma lista com os nodos de profundidade N (sendo N dado). Por exemplo a query

$$\mathsf{?profundo(t(5, t(1, t(3, nil, nil), t(6, nil, nil)), t(2, t(14, nil, nil), t(4, nil, nil))), 3, L)}$$

dá $\mathsf{L} = [3, 6, 14, 4]$

3. Em árvores binárias cujos nodos sejam números podemos ter a definição de árvore *equilibrada*:

`Uma árvore binária diz-se` *`equilibrada`* `se para cada nodo, exceptuando os nodos folha, o número do nodo é igual à soma dos números dos dois nodos filhos.`

Escreva um predicado equi que dado uma árvore binária verifica se esta é equilibrada ou não. Considere as seguintes queries como exemplos:

$\mathsf{?equi(t(7, nil, nil))}$
yes.

$\mathsf{?equi(t(5, t(2, nil, nil), t(2, nil, nil)))}$
no.

4. Escreva um predicado $\mathsf{numeros(M, N, Lista)}$ em que dado um extremo inferior (M) e superior (N), produza uma lista de inteiros compreendidos entre estes dois extremos. Por exemplo a query $\mathsf{?numeros(1, 5, L)}$ tem como resultado $\mathsf{L} = [1, 2, 3, 4, 5]$.

5. Apresente a árvore de prova para a query ?member$(X, [1, 2, 3])$. Assuma o programa standard member/2.

6. Implemente o programa separa/4 que dado um padrão e uma lista devolve duas listas. A primeira com os elementos da lista inicial que ocorrem antes da ocorrência do padrão e a segunda com os elementos depois do padrão. Por exemplo a query ?separa(@, [1, 2, 3, @, 4, 5, 6], P, S) tem como resultado

$$P = [1, 2, 3],$$
$$S = [4, 5, 6].$$

7. É conhecido o problema levantado pelas queries que envolvem os predicados da calculadora de expressões aritméticas em Prolog (e.g. predicado is). Mais precisamente, não é possível calcular expressões que incluem variáveis não instanciadas. Por exemplo, a query ?X is $6 + Y + 2$ em Prolog standard não podem ser calculada, originando um erro (com respectiva mensagem). No entanto alguns Prologs são capazes de lidar com este problema, atrasando a resolução deste tipo de queries até que todas as variáveis nas expressões aritméticas estejam instanciadas. Assim, seleccionam outros goals até que o goal em causa tenha a expressão em situação calculável. Quando não pode seleccionar outros goals e a situação se mantém produz-se uma mensagem de erro e a computação pára.

Escreva um meta-interpretador por forma a que incorpore este mecanismo de delay para ultrapassar o descrito problema no calculo de expressões aritméticas com o predicado is. No caso em que a lista de goals seja composta unicamente por goals para o predicado is com variáveis não instanciadas a computação deve parar e uma mensagem de erro deve ser escrita.

FIM