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

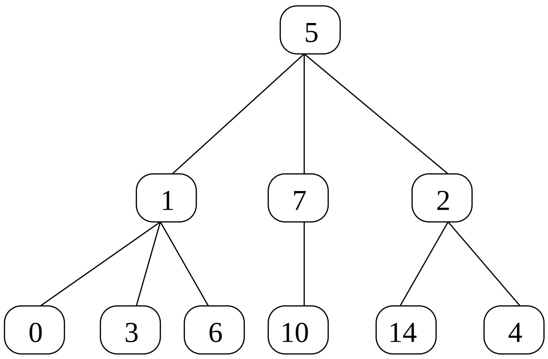
Exame Primeira Chamada **6 de Fevereiro de 1999**

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. Assuma uma relação R que é assimétrica e transitiva. Prove, usando resolução, que R é também irreflexiva.

2. Escreva um programa em Prolog que dado um árvore não binária produza uma árvore binária <u>ordenada</u>. As árvores não binárias podem ser representadas pelo termo $\mathsf{t}(\mathsf{nodo}, [\mathsf{lista\_de\_nodos\_filhos}])$. A árvore da figura



é representada pelo termo

$$\mathsf{t}(5, [\mathsf{t}(1, [\mathsf{t}(0, []), \mathsf{t}(3, []), \mathsf{t}(6, [])]), \mathsf{t}(7, [\mathsf{t}(10, [])]), \mathsf{t}(2, [\mathsf{t}(14, []), \mathsf{t}(4, [])])]).$$

Para a interrogação

$$?\mathsf{treeconv}(\mathsf{t}(5, [\mathsf{t}(1, []), \mathsf{t}(7, []), \mathsf{t}(2, [])]), \mathsf{A})$$

o programa deve produzir o resultado

$$\mathsf{A} = \mathsf{tree}(5, \mathsf{t}(1, \mathsf{nil}, \mathsf{t}(2, \mathsf{nil}, \mathsf{nil})), \mathsf{t}(7, \mathsf{nil}, \mathsf{nil}))$$

2.1 Apresente a árvore de prova (árvore SLD) para a query

$$?\mathsf{treeconv}(\mathsf{t}(5, [\mathsf{t}(1, []), \mathsf{t}(7, []), \mathsf{t}(2, [])]), \mathsf{A}).$$

Não esquecer de apresentar convenientemente os respectivos unificadores mais gerais em cada ramo da árvore.

3. Dado o seguinte meta-interpretador:

```
solve([]).
solve([G|Goals]) :-
        clausula(G,Body),
        append(Body,Goals,More_Goals),
        solve(More_Goals).
```

i. Escreva uma versão deste meta-interpretador, sem usar append, onde se preserva a política "goal mais à esquerda primeiro" mas em que se contabiliza o número de resoluções efectuadas com clausulas unitárias i.e. clausulas em que o corpo é vazio. Para o programa

```
clausula(p(X),[q(X)]).
clausula(q(a),[]).
```

a query ?solve([p(X)], N) tem com resultado N = 1.

ii. 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.

Adapte este 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

Boa Sorte