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

Exame Segunda Chamada **18 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. *Unfolding* é uma técnica de transformação de programas baseado em resolução de cláusulas em tempo de compilação. Resumidamente, seleccionando-se uma cláusula do programa a transformar, por exemplo:

$$A \leftarrow B_1 \ \& \ C,$$

e um literal dentro do corpo dessa cláusula, por exemplo B_1, procura-se todas as outras cláusulas do programa em que a cabeça é um literal que unifica com o literal seleccionado (B_1). Supondo que esta última é unicamente a cláusula:

$$B_2 \leftarrow R \ \& \ S.$$

Então, a cláusula gerada por *unfolding* é:

$$(A \leftarrow R \ \& \ S \ \& \ C)\theta,$$

em que θ é o unificador mais geral entre B_1 e B_2. Assim do programa original:

$$P = \{A \leftarrow B_1 \ \& \ C \ , \ B_2 \leftarrow R \ \& \ S\}$$

geramos o programa

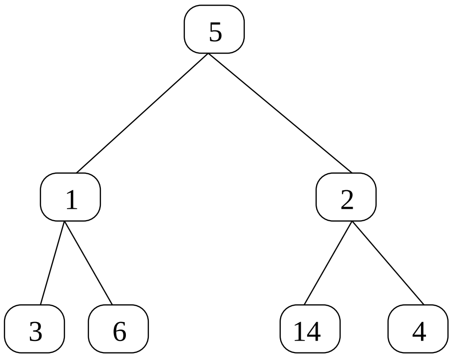
$$U(P) = \{(A \leftarrow R \ \& \ S \ \& \ C)\theta\}.$$

i. Mostre que usando *unfolding* consegue do programa original que define member obter o programa U(member)=

```
member(X,[Y,X|L]).
member(X,[Y1,Y2|L]) :- member(X,L).
```

ii. Mostre, usando árvores de prova, que para a query ?member(2, [1, 2, 3]) o programa transformado U(member) é efectivamente mais eficiente que o programa original member. Justifique a sua resposta.

2. Uma árvore binária pode ser representada pelo termo $\mathsf{tree}(\mathsf{Nodo}, \mathsf{Left}, \mathsf{Right})$ onde $\mathsf{Left}, \mathsf{Right}$ 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 para a árvore acima descrita a query

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

dá $\mathsf{L} = [3, 6, 14, 4]$

3. Considere o seguinte meta-interpretador que simula o modelo computacional do Prolog:

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

É bem conhecido os problemas levantados pelos goals que contém referencias ao operador not. Existe a possibilidade do sistema de inferência do Prolog ser incompleto se forem processados goals com not de queries com variáveis não instanciadas e.g. ?not p(X). Para evitar este problema alguns sistemas Prolog são capazes de atrasar a resolução destes goals até que as variáveis contidas neles fiquem totalmente instanciadas. Assim privilegiam (escolhem) outros goals até que os goals negados estejam num estado "pacifico" de serem calculados.

Adapte este meta-interpretador por forma a que implemente esta ideia de delay de goals com o operador not. No caso em que a lista de goals seja composta unicamente por goals com referencia a not com variáveis não instanciadas a computação deve parar e uma mensagem de erro deve ser escrita.

4. Recorde o seu trabalho prático de geração de horários. Assume que existe um predicado restricoes que define as restrições a impor sobre os horários gerados. Cada aula gerada (conjuntamente com todas as outras já geradas) deve satisfazer as várias restrições. Assim que for gerada uma aula, a interrogação

?not restricoes(Disciplina, Dia, Hora, Sala, Turma)

é processada na esperança de que nenhum clausula restricoes resolva com a query positiva, (onde Disciplina é a disciplina da aula gerada, Dia e Hora é o horário da aula gerada, Sala e Turma são a sala e a turma que recebe esta aula). Ou seja, se alguma restrição suceder então significa violação da restrição. Os horários (e consequentemente as aulas já geradas) são representados nas clausulas

horario(Disciplina, Dia, Hora, Turma, Sala, Entidade).

onde Entidade é a entidade a quem pertence o horário. As disciplinas estão descritas nas clausulas:

disciplina(Disciplina, Tipo, Carga_semanal).

Implemente a restrição, definindo uma ou mais novas clausulas restricoes, que impõe que para cada horário de cada turma, não existam aulas teóricas consecutivas.

FIM

Boa Sorte