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

Exame Primeira Chamada **3 de Fevereiro 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. Considere o seguinte programa:

```
q(a).
q(b).

p(a,b)
p(a,c).
p(b,c).
```

Apresente duas árvores de prova para a query

$$?\mathsf{q}(\mathsf{X}), \mathsf{not}\ \mathsf{p}(\mathsf{X}, \mathsf{Y}), \mathsf{q}(\mathsf{Y}).$$

Uma para representar a computação Prolog da query. A outra para representar a computação de query usando o modelo computacional SLDNF. Não esquecer de apresentar convenientemente os unificadores mais gerais em cada ramo das árvores.

2. Escreva um predicado somabin(Tree, S) que dado uma árvore binária Tree de inteiros calcula a soma S dos respectivos inteiros.

3. Considere uma relação pai(X, Y) que representa que X é pai de Y. Escreva um predicado que constrói uma lista com todos os pares de irmãos existentes.

4. Dado uma lista de pares representativas de jogos entre duas equipas, por exemplo

$$[\mathsf{r}(\mathsf{porto}, \mathsf{benfica}), \mathsf{r}(\mathsf{sporting}, \mathsf{porto}), \mathsf{r}(\mathsf{benfica}, \mathsf{porto}), \mathsf{r}(\mathsf{porto}, \mathsf{braga}), \ldots],$$

escreva o predicado mais_jogos(Lista, Nome) que determina o Nome da equipa com mais jogos.

5. Considere o predicado quicksort/2 com diferença de listas. Explique sucintamente porque é que a computação Prolog da subquery

$$?\mathsf{quicksort}([], [3|\mathsf{S}] - \mathsf{S})$$

não termina.

6. Apresente um predicado que dada uma lista de listas de inteiros produza uma lista de somas dos inteiros. A lista resultado deve ser uma diferença de listas. Por exemplo para a query

$$?\mathsf{somalista}([[2,[1,3,1],3],[1,[5]],[5,7,1]],\mathsf{Soma}-[])$$

obtém-se como resultado $\mathsf{Soma} = [10, 6, 13]$.

7. Recorde o seu trabalho prático sobre o jogo *quatro-em-linha*. Assuma que o tabuleiro de jogo é representado por uma lista de cinco lista onde cada lista representa uma linha do tabuleiro. A ordem das listas está invertida em relação ao tabuleiro o que facilita a simulação da *lei da gravidade*. Isto é, a primeira lista representa a ultima linha (a que está mais perto do solo). O tabuleiro inicial é representado pela lista:

```
[
 [-,-,-,-,-,-,-],
 [-,-,-,-,-,-,-],
 [-,-,-,-,-,-,-],
 [-,-,-,-,-,-,-],
 [-,-,-,-,-,-,-]
]
```

Escreva um predicado $\mathsf{insere}(\mathsf{Tabuleiro}, \mathsf{Coluna}, \mathsf{Ficha}, \mathsf{Novo})$ que dado o tabuleiro $\mathsf{Tabuleiro}$ insere a ficha Ficha na coluna Coluna gerando o tabuleiro Novo.

8. Dado o seguinte meta-interpretador:

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

Escreva uma versão deste meta-interpretador onde se implementa a regra de computação que escolhe o subgoal da lista de sub-interrogações que resolve com o menor número de clausulas. Por exemplo da lista $[\mathsf{p}(\mathsf{X}), \mathsf{q}(\mathsf{Y})]$ e do programa

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

a regra de computação escolheria o goal $?\mathsf{q}(\mathsf{Y})$.

FIM

Boa Sorte