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

Exame Primeira Chamada **24 de Janeiro de 2002**

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 a seguinte fórmula lógica

$$T = \{ \quad p(X) \ \wedge \ (q(X) \rightarrow t(X)) \quad \}.$$

Mostre usando resolução que $T \vdash p(a)$.

2. Escreva um predicado em Prolog que coleccione numa diferença de listas os elementos de um determinado nível de uma dada árvore binária. Por exemplo:

$$?\mathsf{nivel}(2, \mathsf{t}(1, \mathsf{t}(2, \mathsf{nil}, \mathsf{nil}), \mathsf{t}(3, \mathsf{nil}, \mathsf{nil})), \mathsf{L} - [])$$

o resultado é:

$$\mathsf{L} = [2, 3].$$

3. Apresente uma árvore SLD (de prova) para a query

$$?\mathsf{append}(\mathsf{A}, \mathsf{B}, [1, 2])$$

assumindo o programa append standard.

4. Dada uma lista de termos Prolog, construa um predicado que identifique quais os termos que contém um dado functor. O predicado devolve como resultado uma diferença de listas com os termos em questão. Por exemplo:

$$?\mathsf{termos}(\mathsf{f}, [\mathsf{p}(1, \mathsf{f}(\mathsf{X})), \mathsf{q}(\mathsf{a}), \mathsf{member}(1, \mathsf{f}(\mathsf{X})), \mathsf{f}, \mathsf{f}(\mathsf{f}(1)), \mathsf{a}], \mathsf{L} - [])$$

o resultado é:

$$\mathsf{L} = [\mathsf{p}(1, \mathsf{f}(\mathsf{X})), \mathsf{member}(1, \mathsf{f}(\mathsf{X})), \mathsf{f}, \mathsf{f}(\mathsf{f}(1))]$$

5. Considere uma relação web/2 que descreve os acessos a páginas web. Por exemplo,

$$\mathsf{web}(\mathsf{pag1}, \mathsf{user1})$$

descreve que o utilizador **user1** consultou a página pag1. Escreva um predicado que determine o nome da página com mais acessos.

6. Explique sucintamente e usando exemplos porque é que o procedimento SLD com negação por falhanço é um sistema potencialmente incompleto.

7. Uma empresa de informática pretende gerar um plano de afectação de tarefas pelos seus colaboradores. Para isso, considera-se a informação sobre cada colaborador que consta do nome, idade, sexo e lista de funções capaz de exercer. Durante a geração do plano de tarefas temos de ter em conta as seguintes restrições:

- Não é possível atribuir mais do que duas tarefas ao mesmo colaborador,
- A interface deve ser desenhado por uma colaboradora,
- Quem desenvolver a base de dados não pode escrever os manuais,
- A pessoa que fizer o debugging também tem de fazer a tarefa de teste,
- A tarefa de análise tem de ser atribuda a um colaborador com mais do que 35 anos.

Escreva um predicado funcoes(L, X) que calcule a lista X de afectação de tarefas. A informação dos colaboradores é representada na cláusula info/4. Considere como exemplo as seguintes cláusulas:

```
info(ana,f,27,[programar,bases_de_dados,interfaces]).
info(joao,m,20,[marketing,debugging,programar,testar]).
info(rui,m,22,[bases_de_dados,testar,escrever_manuais]).
info(rita,f,40,[escrever_manuais,analise,programar]).
info(julia,f,25,[debugging,bases_de_dados,interfaces]).
```

Perante estes candidatos a query

?funcoes([analise, interfaces, programar, bases_de_dados, debugging, testar, escrever_manuais], X)

tem como possível resultado

```
X = [analise/rita, interfaces/julia, programar/ana, bases_de_dados/rui,
     debugging/joao, testar/joao, escrever_manuais/rita ]
```

FIM