



Universidade do Minho

Recebido por \_\_\_\_\_

Nº \_\_\_\_\_

Nome \_\_\_\_\_

Licenciatura em Engenharia Informática  
Sistemas de Representação de Conhecimento e Raciocínio  
3º Ano, 2º Semestre  
Ano letivo 2012/2013

Prova escrita, em exame  
10 de julho de 2013

### GRUPO 1

(6 valores)

Responda às questões deste grupo em folha separada.

#### QUESTÃO 1

Considere os dados constantes na Tabela 1, onde se apresenta um conjunto de elementos respeitantes a um grupo de indivíduos, tais como a formação académica e a atividade profissional e respetivo vencimento.

Tabela 1  
Formação académica,  
atividade profissional e  
vencimento.

PESSOA	FORMAÇÃO	FUNÇÃO	VENCIMENTO
Andreia	Licenciatura		
Belmiro		Professor	1500
Carlos		Tesoureiro	750
Carlos		Contabilista	1000
Davide	Bacharel	Varredor	500
Eduardo	{ Licenciatura, Mestrado }		
Fernando	Doutoramento	Presidente	<u>variável</u>
Guilherme		Administrador	#
Hélder	Básica	{ Calceteiro, Pedreiro, Alvanel }	990
Ivo	<u>formado</u>		

Sabe-se, ainda, que a formação do Belmiro não é doutoramento.

Atenda a que ' { ' e ' } ' é a notação usada para denotar conjuntos e ' [ ' e ' ] ' é a notação para representar intervalos, para valores nulos do tipo impreciso. Os átomos 'variável' e 'formado' denotam valores nulos do tipo incerto. O símbolo ' # ' representa um valor nulo do tipo interdito.

- Descreva o sistema a desenvolver em termos da representação do conhecimento;
- Represente o conhecimento descrito na Tabela 1 em termos das extensões dos predicados descritos em a);
- Apresente o invariante que garanta não ser possível remover informação respeitante à formação de um indivíduo enquanto existir informação sobre o vencimento e a profissão.

#### QUESTÃO 2

Num sistema de Representação de Conhecimento Imperfeito, a problemática da evolução do sistema deve considerar a assimilação de conhecimento perfeito e imperfeito.

Explique que procedimentos deverão ser tomados em consideração para implementar estes mecanismos de evolução de conhecimento.

**GRUPO 2**  
(6 valores)

Responda às questões deste grupo neste enunciado, assinalando a veracidade (V) ou falsidade (F) das afirmações produzidas, justificando a resposta EXCLUSIVAMENTE no espaço disponibilizado.

NÃO SERÃO CONSIDERADAS respostas para as quais não exista uma justificação expressa.

## QUESTÃO 1

☐ O predicado  $is( \text{Termo}, \text{Expressão} )$  realiza a operação aritmética dada em **Expressão**, unificando o respetivo resultado com **Termo**.

Justificação: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

## QUESTÃO 2

☐ Na linguagem de programação em lógica PROLOG, a adoção do pressuposto do mundo fechado é dependente da extensão de cada predicado.

Justificação: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

## QUESTÃO 3

☐ Em PROLOG, é possível construir mecanismos de raciocínio que recorram a invariantes para a representação de conhecimento negativo.

Justificação: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

## QUESTÃO 4

☐ No contexto da programação em lógica estendida, a negação por falha na prova tem como objetivo a representação de conhecimento negativo.

Justificação: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

## QUESTÃO 5

☐ Em termos da representação de conhecimento imperfeito, é possível que um valor nulo do tipo impreciso identifique um conjunto infinito de valores.

Justificação: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

## QUESTÃO 6

☐ Num sistema hierárquico de representação de conhecimento, é possível construir mecanismos de controlo sobre a herança que considerem a autorização para herdar conhecimento.

Justificação: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

**GRUPO 3**  
(6 valores)

Responda às questões deste grupo neste enunciado, assinalando a veracidade (V) ou falsidade (F) das afirmações produzidas.

EM CADA QUESTÃO, uma afirmação incorretamente assinalada ANULA outra afirmação assinalada corretamente.

**QUESTÃO 1**

Assuma que os predicados `é_um( Agente, Superior )` e `agente( Agente, Teoria )` são os adequados para a representação de conhecimento num sistema de raciocínio de base hierárquica.

Considere, ainda, os predicados `demo1( Agente, Questão )` e `demo2( Agente, Questão )`, cujas extensões corporizam sistemas de inferência, dados no que se segue:

`demo1( Agente, Questão ) :-`  
     `agente( Agente, Teoria ),`  
     `prova( Questão, Teoria ).`

`demo1( Agente, Questão ) :-`  
     `é_um( Agente, Superior ),`  
     `demo1( Superior, Questão ).`

`demo2( Agente, Questão ) :-`  
     `agente( Agente, Teoria ),`  
     `prova( Questão, Teoria ).`

`demo2( Agente, Questão ) :-`  
     `agente( Agente, Teoria ),`  
     `não( pertence( Questão, Teoria ) ),`  
     `é_um( Agente, Superior ),`  
     `demo2( Superior, Questão ).`

- ☐ O predicado `demo1` não implementa qualquer mecanismo de controlo sobre a herança.
- ☐ A extensão do predicado `demo1` impossibilita encontrar todas as soluções para uma dada questão.
- ☐ No predicado `demo2`, a extensão dos predicados `prova` e `pertence` deverá ser igual.
- ☐ O mecanismo de controlo sobre a herança apresentado no predicado `demo2` não é eficaz se a `Teoria` do agente for composta exclusivamente por cláusulas que denotem factos.
- ☐ Quando a `Teoria` do agente é vazia, o predicado `demo2` não apresenta capacidade de herdar conhecimento.

**QUESTÃO 2**

Considere as bibliotecas LINDA do SICStus PROLOG.

- ☐ Os predicados `in( X )` e `rd( X )` bloqueiam enquanto não é possível unificar o termo dado em `X`.
- ☐ O predicado `close_client( Máquina: Porta )` encerra a ligação estabelecida ao quadro negro identificado pelo endereço `Máquina: Porta`.
- ☐ O predicado `in( Lista, X )` sucede quando todos os elementos da `Lista` estiverem disponíveis para leitura.
- ☐ O predicado `out( X )` regista o termo dado em `X` no espaço de memória partilhada (quadro negro).
- ☐ O predicado `out( X )` nunca falha, exceto se o termo dado em `X` for uma variável não unificada.

## QUESTÃO 3

Considere o seguinte excerto de PROLOG, que mostra a extensão de predicados usados na implementação de um agente inteligente que realiza cálculos sobre sequências de números, recorrendo ao modelo de distribuição de computação baseado em quadros negros, disponibilizado pelas bibliotecas LINDA do SICStus PROLOG:

```
run( Agente ) :-
    in( [op( max ), op( min )], op( OP ) ),
    bagof_rd_noblock( X, n( X ), Li sta ),
    Predi cado=. . [OP, Li sta, Resul tado],
    call( Predi cado ),
    out( resul tado( OP, Resul tado ) ),
    run( Agente ).

max( [X], X ).
max( [X|R], X ) :-
    max( R, M ), X > M
max( [X|R], M ) :-
    max( R, M ), X <= M

min( [X], X ).
min( [X|R], X ) :-
    min( R, M ), X < M
min( [X|R], M ) :-
    min( R, M ), X >= M
```

- ☐ A expressão `op( Operação )` é um termo da linguagem de comunicação que identifica a operação a ser aplicada à sequência de números.
- ☐ A expressão `resul tado( Operação, Sol ução )` é um termo da linguagem de comunicação que identifica uma e uma só solução para cada questão colocada a um agente.
- ☐ Um agente corporizado por estes procedimentos tem capacidade para calcular o máximo e mínimo de uma sequência de números.
- ☐ O predicado `run( Agente )` está errado porque tem capacidade para realizar duas operações diferentes, mas apenas permite identificar um resultado.
- ☐ Um agente inteligente que englobe estes procedimentos não funciona porque os predicados `max(L, M)` e `mi n(L, M)` não usam primitivas das bibliotecas LINDA.

**GRUPO 4**  
(2 valores)

Responda às questões deste grupo em folha separada.

Considere a extensão do predicado filho:  $X, Y \rightarrow \{V, F\}$ , que denota que X é filho de Y, dada na forma:

```
...
filho(joao, carlos).
filho(carlos, pedro).
...
```

- a) Apresente a árvore de prova, que lhe permita responder à questão:  
Quem é filho de quem?
- b) Justifique a sua resposta, i.e., apresente a extensão da relação de derivabilidade.