

Paradigmas da Programação I

LECom (2º ano)

Exame de 1ª Época – 2ª Chamada

Data: 30 de Janeiro de 2006
Hora: 09:30 (Sala B1.17)

Dispõe de 2:30 horas para realizar este exame

Questão 1: Bases de Conhecimento

Um *mapa de conceitos*, ou *rede semântica*, é uma excelente forma de modelar o conhecimento relativo a determinada área do saber e é usado, com frequência, pelos professores para planificarem uma disciplina ou para preparar as aulas; os *conceitos* colocam-se nos nodos (ou vértices) do grafo e as *relações* (ou *associações*) colocam-se como etiquetas nos ramos que ligam os nodos.

Considere a base de conhecimento (BC) formada, apenas por factos, que modela o *mapa de conceitos* referente a uma disciplina de Introdução à Informática (apenas se mostra um extracto) — como é óbvio, as *relações* foram representadas por *predicados* e os *conceitos* são *valores* que se comportam como *argumentos dos predicados*; um *facto* na BC representa uma *associação* no grafo.

```
inclui(hardware, disco-duro).
inclui(hardware, memória-central).
inclui(hardware, processador-central).
composto-por(software, instruções).
composto-por(software, variáveis).
e-parte-de(monitor, hardware).
e-parte-de(impressora, hardware).
e-parte-de(sistema-operativo, software).
e-parte-de(utilitário, software).
e-um(access, utilitário).
e-um(power-point, utilitário).
e-um(word, utilitário).
e-um(linux, sistema-operativo).
e-um('GestCli3', aplicação-final).
serve-para(internet-explorer, navegação-site).
serve-para(internet-explorer, construção-site).
serve-para(power-point, edição-poster).
serve-para(power-point, edição-diapositivos).
serve-para(word, edição-texto).
```

Responda, então, às alíneas seguintes:

- a) qual a questão (*nota*: pode precisar de uma regra com mais de um átomo lógico, ou mesmo de mais de uma regra) que devia colocar ao Interpretador de Prolog para saber:
1. tudo o que um dado *conceito* inclui ?
 2. tudo o que serve para realizar qualquer tarefa ?

3. quais são os 'utilitarios' conhecidos ?
4. as componentes do computador que pertencem ao 'hardware' ?

b) observe a seguinte cláusula

```
xxx(X) :- e-um( X, P ), e-parte-de( P, software).
```

e diga:

1. o que infere, na variável X, o predicado xxx ? Esclareça a sua resposta indicando o primeiro e último valor calculado para X.
 2. caso se substituísse, na cláusula acima, o valor `software` pela variável S , o que seria diferente na sua resposta à alínea anterior? Justifique.
 3. poderíamos interrogar o Interpretador de Prolog com a questão `xxx(monitor)` ? se sim, diga qual a resposta calculada pelo Interpretador de Prolog; se não, diga o que era inválido (em qualquer caso, justifique a resposta).
 4. poderíamos interrogar o Interpretador de Prolog com a questão `xxx('GestClie3')` ? se sim, diga qual a resposta calculada pelo Interpretador de Prolog; se não, diga o que era inválido (em qualquer caso, justifique a resposta).
 5. qual seria o efeito prático de inserir o operador `cut` ("!") entre o primeiro e segundo átomos do corpo da cláusula acima ? Justifique.
- c) caso fosse mandatório garantir a validade da BC, em termos de assegurar que todos os conceitos usados nos argumentos dos factos são válidos (isto é, fazem parte de um conjunto de termos considerados os conceitos base da área de conhecimento em causa), diga que informação precisava de acrescentar à BC listada para poder fazer essa verificação.
Assumindo que a BC foi aumentada com o conhecimento que referiu, escreva um predicado `verifica1/0` que realiza a validação pretendida para o caso do predicado `inclui/2`.
- d) O critério usado acima para modelar em Prolog o *mapa de conceitos*, embora pareça muito natural, não era a melhor forma de descrever esse conhecimento!
1. explique porque é que a BC apresentada acima não permitia facilmente responder à questão: *todos os conceitos que estão ligados a um dado conceito*.
 2. justifique a afirmação seguinte “era preferível ter usando apenas o predicado `ligados(conceito1, relac, conceito2)` para modelar o referido mapa de conceitos”. Esclareça a sua resposta, formulando então a questão mencionada na alínea anterior.
 3. mostre como ficaria a BC no caso de optar pelo esquema de representação descrita na alínea anterior (use apenas alguns exemplos variados da BC original (listada acima)).

Questão 2: Manuseamento de Listas

Sobre operações com Listas em Prolog, responda às alíneas seguintes:

- a) Pretende-se que implemente um predicado `eliminaRep/2` que, dada uma lista, devolva como segundo argumento a mesma lista sem elementos repetidos. Exemplo:

```
?- eliminaRep([1,2,3,4,2,3,4,4], C).
   C = [1,2,3,4]
```

- b) Pretende-se que implemente um predicado `procura/3` que, dada uma palavra em português e um dicionário português-inglês que pode conter mais de uma entrada para cada palavra, —implementado como uma lista de pares (palavra-português, tradução-inglês)— retorne no terceiro argumento uma lista com todas as traduções alternativas para inglês. Exemplo:

```
?- procura(bom, [(bom,good), (mau,bad), (quente,hot), (bom,nice)], T).
   T = [good,nice]
```

- c) Pretende-se que implemente um predicado **emparelha/3** que, dadas duas listas, devolva no terceiro argumento uma lista de pares em que o primeiro elemento pertence à primeira lista e o segundo à segunda lista e tal que cada elemento da primeira é combinado com todos os elementos da segunda. Exemplo:

```
?- emparelha([1,2], [a,b,c], L).
   L = [(1,a), (1,b), (1,c), (2,a), (2,b), (2,c)]
```

Questão 3: Representação de Conhecimento

Na sua função de Organizador da Jornadas da LECom precisa de criar um sistema de informação (dito SIJ06) para poder gerir o **programa**, os **oradores convidados** (que vem fazer as palestras do programa) e as **inscrições dos participantes**. O programa está organizado em sessões; a cada **sessão** (de 1h30) corresponde um tema (da lista de temas escolhidos para as Jornadas de 2006), um dia/hora e sala. Por sua vez, cada sessão comporta 3 **palestras** (30min cada), cada uma com seu título e orador. Para cada orador é preciso saber, além dos dados de identificação pessoal, a instituição onde trabalha e o país de onde vem, bem como o hotel onde vai ficar alojado e as datas de chegada e partida. Por fim, cada inscrição deve ter os dados do respectivo participante e a classe (aluno, docente, ou empresa).

Usando a abordagem seguida nas aulas para modelar sistemas de informação em lógica, recorrendo à linguagem de programação Prolog para poder interrogar de seguida o seu universo de discurso, construa uma Base de Conhecimento (BC) que descreva o SIJ06.

Após identificar o tipo de cláusulas (factos ou regras) que deve usar para modelar o sistema, indique algum tipo de perguntas que poderiam ser respondidas com o seu programa lógico.

Questão 4: Autómatos Deterministas Reactivos

Pretende-se desenvolver um autómato reactivo que modele a função *Despertador* de um telemóvel Nokia de 1ª geração, como se descreve a seguir.

Após receber o sinal **Menu**, a opção **Despertador** selecciona-se pressionando 3 vezes a seta para baixo (\downarrow), ou 2 vezes a seta para cima (\uparrow); alternativamente, pode introduzir-se o número da opção (4). Confirmada a opção, com o sinal **OK**, o utilizador deve introduzir os 4 dígitos correspondentes às Horas e Minutos; a meio ou no fim, confirma a hora de despertar com o sinal **OK**. A seguir deve indicar se a definição é apenas para um dia (sinal **OK**) ou se se aplica a vários dias consecutivos (sinal seta para baixo, \downarrow , seguido de **OK**). Por fim, conclui a operação com o sinal **C**.

Note porém que se o *Despertador* já estava activado, após a confirmação da opção 4, o utilizador tem de escolher se quer *ajustar a hora* (sinal **OK**), seguindo-se a introdução dos dígitos respectivos e o processo normal acima descrito; ou se quer *desactivá-lo* (sinal seta para baixo, \downarrow , seguido de **OK**), terminando logo a operação.

Implemente, então, em Prolog esse autómato determinista reactivo que será identificado pelo nome 'dp' (*despertador nokia*); escreva apenas os predicados **automato/7** e **deltalfa/5** (não precisa de escrever o predicado **acciaoSem/1** para definir as acções semânticas).

Mostre, no fim, como podia usar o predicado genérico **processa/2** para verificar se uma determinada sequência de comandos (escolha uma concreta) é válida para marcar uma hora no dito *despertador nokia*.