

Paradigmas da Programação I

MiECom (2º ano)

Exame de 1ª Época – 2ª Chamada

Data: 29 de Janeiro de 2007

Hora: 09:30

Dispõe de 2:30 horas para realizar este exame

Leia as questões com toda a atenção
e responda com calma e clareza em folha convencional

Já trabalhámos o quiosque, a ourivesaria e a padaria de Dume. É agora a vez de informatizar a Mercearia de Dume, MDume, para ficar “controlado” todo o comércio dumiense nesse quarteirão elegante de Lamações. O seu exame vai então desenvolver-se em torno dessa tarefa.

Questão 1: Bases de Conhecimento

Considere a BC formada apenas por factos correspondentes aos predicados `fresco/6` (produto perecível que é comercializado no dia, não se armazena), `classe/1`, `unidade/1` e `fornecedor/3` que podia ser um extracto do programa Prolog que alguém desenvolveu para apoio à gestão da MDume.

```
% fresco(Ref, Nome, PrecUnit, Unidade, Classe, Proveniencia).
fresco(a1, cenouras, 1, quilo, legume, f1).
fresco(a2, grelos, 2, molho, legume, f1).
fresco(b11, salsa, '0.5', molho, salada, f1).
fresco(c21, leite, '1.5', litro, lacticinio, f2).
fresco(dda, truta, 6, peca, peixe, f3).

% classe(Nome).
classe(fruta).
classe(salada).
classe(legume).
classe(lacticinio).
classe(peixe).

% unidade(Nome).
unidade(quilo).
unidade(litro).
unidade(molho).
unidade(peca).
unidade(caixa).

% fornecedor(Codigo, Regiao, Contacto).
fornecedor(f1, minho, 919191911).
fornecedor(f2, beira, 933332221).
fornecedor(f3, minho, 966665554).
fornecedor(f4, douro, 918881117).
```

Responda, então, às alíneas seguintes:

a) Porque razão designamos as *cláusulas de Horn* acima por *factos* e não por *regras*?

b) Escreva um predicado que, sem requer a intervenção do utilizador, produz automaticamente as respostas às questões seguintes

1. o código de todos os fornecedores do Minho?
2. o telefone de todos os fornecedores de peixe?
3. o nome de todos os produtos que se vendem ao quilo, ao litro, ou à caixa?

c) Observe as 2 seguintes questões

```
?- unidade( litro ).
?- unidade( Litro ).
```

e diga qual a diferença entre ambas; indique a resposta dada pelo Interpretador de Prolog a cada uma delas. Se a ordem dos 5 factos unidade da BC fosse alterada, isso iria provocar alguma alteração na resposta calculada às 2 questões acima?

d) Diga o que é o *mecanismo de unificação* e explique a sua importância no cálculo da resposta às questões que se colocam ao Interpretador de Prolog.

e) Recorrendo ao predicado standard do Prolog `assert/1`, que acrescenta dinamicamente cláusulas a uma BC, escreva um predicado `novaClasse/1` que recebe o Nome de uma Classe de frescos e junta à BC (no início) um novo facto `classe/1` com esse nome, desde que ainda não exista na BC nenhuma classe com o Nome em causa.

f) Escreva um predicado `valida/0` que verifica se a *classe* e *unidade* de todos os *frescos* está correcta, isto é, se existe definida na BC como uma designação de *classe*, ou *unidade*, válida. Cada anomalia detectada deve ser reportada.

g) Escreva um predicado `limpa/0` que remove da BC todos os *fornecedores* que não estão a ser usados, isto é, que não estão referidos em nenhum *fresco*.

Questão 2: Manuseamento de Listas

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

a) Sendo `vinhos` uma lista de pares (`região, lista de marcas`), implemente um predicado `busca/3` que, dados os `vinhos` da mercearia e uma lista de `regiões` (como 2º argumento), devolva como terceiro argumento a lista de todas as marcas existentes relativas às regiões indicadas no 2º argumento. Exemplo:

```
?- busca([(douro, [v1,v2]), (alentejo, [v3,v4]), (algarve, [v5])], [alentejo, algarve], Ms).
Ms = [v3,v4,v5]
```

b) Supondo que a lista `vinhos` acima, é modificada para passar a ser uma lista de ternos, (`região, tipo, lista de marcas`), pretende-se que implemente um predicado `conta/2` que receba essa lista como 1º argumento e devolva como segundo argumento o número de regiões diferentes que constam da lista (note que a introdução do tipo faz com que a mesma região possa aparecer mais de uma vez na lista). Exemplo:

```
?- conta([(douro, tinto, [v1,v2]), (douro, branco, [v3,v4]), (douro, rose, [v5])], C).
C = 1
```

c) Voltando a considerar a lista `vinhos` da alínea anterior, implemente um predicado `qualRegiao/3` que, recebendo `vinhos` como 1º argumento e uma `marca` como 2º argumento, devolva como 3º argumento a `região` a que essa marca pertence, ou o átomo `nao-se-sabe`, caso a `marca` não exista à venda na mercearia. Exemplo:

```
?- qualRegiao([(douro, tinto, [v1,v2]), (douro, branco, [v3,v4]), (douro, rose, [v5])], v3, R).
R = douro
```

Questão 3: Representação de Conhecimento

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 SGM Dume, Sistema para Gestão da mercearia de Dume, constituído por:

- **produtos comercializados e em stock** (os *frescos* também são comercializados, mas como não são armazenados em stock não fazem parte desta sistema de gestão). Todos esses produtos (identificados univocamente por uma referência) pertencem a uma de três famílias: **bebidas** (vinhos, águas, sumos); **mercearia grossa** (arroz, farinha, açúcar, etc.); **mercearia fina** (chocolates, rebuçados, especiarias, etc.).
- **clientes** relativamente aos quais se guarda, além dos dados de identificação (pessoal e fiscal) e contacto, informação para marketing (a lista de preferências, os produtos mais comprados, etc.).
- **encomendas**, lista de produtos encomendados (e respectiva quantidade) numa determinada data por um cliente
- **guias de remessa**, que inclui a lista dos produtos encomendados (com a quantidade disponível para envio) e a lista de produtos recomendados.

O objectivo do SGM é **controlar a existência** de produtos em stock, a qual deve ser actualizada em cada venda (satisfação de uma encomenda). Cada produto tem um **valor de alerta** associado, de modo a disparar um alarme (produzir uma mensagem) sempre que a existência descer abaixo desse valor. Os produtos também tem uma data de compra e um número de meses de validade, pretendendo-se emitir um alarme sempre esteja atingido esse prazo de validade.

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 pela sua BC.

Questão 4: Autómatos Deterministas Reactivos

Pretende-se desenvolver um autómato reactivo que modele o comportamento, que se descreve abaixo, da balança-calculadora (BiCalc) usada na MDume.

Implemente, então, em Prolog esse autómato determinista reactivo, que vá descrevendo as operações que a BiCalc irá executar à medida que recebe comandos/estímulos; escreva apenas os predicados `automato/7`, `deltalfa/5` e `acciaoSem/1`.

Uma vez ligada a balança, esta aceita apenas impulsos (1 ou mais) correspondentes ao peso do produto colocado no prato; 1 impulso por cada 10gramas. A sequência de impulsos de pesagem termina com um sinal de fim-de-pesagem (FdP). Atingido o fim da pesagem, a BiCalc aguarda o código do produto, o qual é uma sequência de comprimento fixo formada por uma letra e dois dígitos. Após a introdução do código já é possível calcular o preço a pagar graças a uma tabela de preços que a BiCalc tem na sua memória. Nesse estado, a balança pode recomeçar novo processo de pesagem que se vai acrescentar ao anterior, se voltar a receber impulsos de pesagem; ou terminar o processo, caso receba uma ordem de **emitir-talão**. Em qualquer estado, a BiCalc pode voltar ao estado inicial, perdendo tudo que está em memória, caso receba o comando **desligar**.