
PARADIGMAS DE PROGRAMAÇÃO III

2004/05 - Janeiro (1ª chamada)

LESI e LMCC - 2º Ano - 19 de Janeiro de 2005

Duração Máxima: 2 horas

Obs: Responda a cada uma das partes em folhas separadas e bem identificadas

PARTE I

1: LISTAS e RECURSIVIDADE (5 pontos)

1.1.- Dada uma lista L de números inteiros escreva:

- a) Um predicado *comp/2* que dê como resultado o comprimento da lista parâmetro.
- b) Um predicado *posMenor/2* que dê como resultado a posição do menor elemento da lista (head = posição 1).
- c) Um predicado *sublista/4*, que dê como resultado a sublista da lista parâmetro entre os índices I1 e I2 inclusivé (se qualquer dos índices não for válido o resultado deverá ser uma lista vazia).

1.2.- Defina um predicado *criaLista(N1, N2, L)* que construa, usando backtracking, uma lista L contendo todos os inteiros entre N1 e N2, inclusivé.

2: Semântica do Prolog (1.5 pontos)

Qual a semântica lógica do seguinte programa Prolog:

P :- A, B, !, C.

P :- !, fail, D.

P :- E.

PARTE II

Queries, Múltiplas Soluções e Ficheiros (7.5 pontos)

Num sistema de controlo da produção cada componente necessário à linha de fabrico é descrito por uma estrutura *componente(id, tipo, listadesubcomponentes, quantStock, qtalerta)* onde cada parâmetro tem o seguinte significado:

id: código interno do componente;

tipo: componente simples ou composto (a definir simples/composto, 0/1, s/c, etc...);

listadesubcomponentes: se o componente é simples esta lista é [] senão tem os ids dos respectivos subcomponentes que constituem o componente;

quantStock: quantidade deste componente existente em stock;

qtalerta: quantidade abaixo da qual se devem encomendar mais;

Admitindo que numa BC se encontram armazenados predicados deste tipo, que representam o conjunto de componentes disponíveis para um dada produção, defina predicados que permitam responder às seguintes questões:

- a) *Qual o número total de diferentes componentes?*
- b) *Qual o número total de componentes compostos?*
- c) *Dada a identificação de um componente, existente ou não, qual o número total de diferentes subcomponentes, simples ou compostos, que o constituem ?*
- d) *Determinar a lista com os identificadores de todos os componentes simples da BC.*
- e) *Dado um componente simples, determinar a lista de todos os identificadores de componentes de que este é subcomponente (transitividade deve ser tida em atenção);*
- f) *Gravar num ficheiro F todos os componentes com quantidade em stock abaixo da quantidade de alerta.*
- g) *Eliminar da BC todos os componentes compostos que tenham um dado componente, cuja identificação é dada como parâmetro, na sua lista de subcomponentes.*

PARTE III

Grafos (6 pontos)

Numa BC foram introduzidos predicados **workflow(cod_Doc, lista_cod_Doc)**, onde *cod_Doc* representa um código de um documento e *lista_cod_Doc* é uma lista de códigos de outros documentos, representando os documentos que é necessário resolver para que o documento *cod_Doc* esteja resolvido. Cada facto *workflow/2* representa todos os documentos necessários para validar o documento cujo código é o 1º parâmetro. Um documento independente de outros é representado por **workflow(cod_Doc, [])**, ou seja, a sua validação/resolução não depende de qualquer outro documento.

- a) *A partir de uma BC contendo apenas factos **workflow/2**, crie um grafo de dependências entre documentos, no qual cada nodo contém o código de um dado documento e cada arco orientado (que liga este nodo a outro) representa a dependência entre este documento e o documento do nodo destino. Cada ligação deste grafo deverá ser representada como **doc_Depende_De(cod_Doc1, cod_Doc2)**.*
- b) *Considere que para a instalação de um **workflow** completo para uma dada instituição necessita de uma lista L contendo códigos válidos de documentos. Usando o grafo anteriormente criado (ou em alternativa os factos **workflow/2**), determine a lista de todos os documentos independentes.*
- c) *Determine o código do documento que para ser validado/resolvido depende de mais documentos.*