

Paradigmas da Programação I

MiECom (2º ano)

3º Teste

Data: 17 de Janeiro de 2008
Hora: 13:00H – 14:00H

Dispõe de 1:30 hora para realizar este exame

Questão 1: Autómatos, Bases de Conhecimento e Listas

Para simular o funcionamento de uma máquina de transição de estados (MTE) e reconhecer se uma sequência de símbolos terminais (uma *Frase*) é aceite pelo autómato determinista que modela a dita MTE, apresentou-se na aulas um programa Prolog, que se lista abaixo e que tem a importante característica de ser absolutamente genérico, isto é, sempre o mesmo para todos os casos (simula qualquer MTE).

```
reconhece( Frase ) :- automato( _, _, S, _ ), transita( S, Frase ).

transita( erro, _ ) :- !, fail.
transita( Q, _ )    :- automato( _, _, _, Z ), member( Q, Z ), !.
transita( _, [] )   :- !, fail.
transita( Q, [H|T] ):- delta( Q, H, P ), transita( P, T ).
```

Este programa genérico depende de 2 predicados, `automato/4` e `delta/3`, que descrevem cada MTE específica, como se exemplifica abaixo para um autómato concreto:

```
automato( [a,b,c,d], [1,2,3,4,5], 1, [3,5] ).
delta( 1,a,2 ).
delta( 1,d,4 ).
delta( 1,_,erro ).
delta( 2,b,2 ).
delta( 2,c,3 ).
delta( 2,_,erro ).
delta( 4,a,2 ).
delta( 4,c,3 ).
delta( 4,b,5 ).
delta( 4,_,erro ).
```

Tomando em consideração a Base de Conhecimento (BC) constituída pelas regras e factos acima, responda às alíneas seguintes:

- Explique por palavras suas a lógica do predicado `reconhece/1`.
- Explique por palavras suas a lógica do predicado `transita/2`.
- Represente graficamente o autómato descrito em Prolog acima, através dos predicados `automato/4` e `delta/3`.
- Diga que questão devia colocar ao seu interpretador Prolog para saber todos os estados que vão dar ao estado 2, isto é, a partir dos quais há um transição válida para 2.

- e) Diga que questão devia colocar ao seu interpretador Prolog para saber todas as transições válidas pelo símbolo terminal T.
- f) Escreva uma predicado `novaTrans/3` que recebe como argumentos—1 estado, 1 terminal e 1 estado—e acrescenta à base de factos um novo facto `delta/3`, caso ainda não haja nenhuma transição a partir desse estado e desse terminal; se já existe, não faz nada.
- g) Diga o que faz o predicado `enigma/1` abaixo

```
enigma( C ) :- automato( _,Q,_,_ ) apura( Q,0,C ).
apura( [],C,C ).
apura( [_|T],CA,CF ) :- CN is CA+1, apura( T,CN,CF ).
```

- h) Escreva uma predicado `valida/0` que percorre a base de factos e valida todas as cláusulas do predicado `delta/3` garantindo que o 2º argumento é sempre um elemento membro do conjunto T dos símbolos terminais do autómato.

Questão 2—Autómatos Deterministas: Despertador de um Telemóvel Nokia

Pretende-se desenvolver um autómato finito determinista (desenhe o grafo que representa a sua função de transição `delta`) que modele o funcionamento, se descreve a seguir, de *Despertador* de um telemóvel Nokia de 1ª geração.

Após receber o sinal `Menu`, a opção `Despertador` selecciona-se pressionando 3 vezes a seta para baixo (`↓`), ou 2 vezes a seta para cima (`↑`); alternativamente, pode introduzir-se o número da opção (4).

Confirmada a opção, com o sinal `OK`, é gerado um sinal interno, com o valor `activo` ou `não-activo`.

Se o despertador ainda não estava activado, o utilizador só pode escolher a hora de despertar, para o que deve introduzir os 4 dígitos correspondentes às Horas e Minutos; a meio ou no fim, confirma a hora de despertar com o sinal `OK`. Depois do `OK`, conclui a operação com o sinal `Sair`.

Se o *Despertador* já estava activado, 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, `↓`, seguido de `OK`), terminando logo a operação.

Implemente, então, em Prolog esse autómato determinista, escrevendo apenas os predicados `automato/4` e `delta/3`.