Ficha Prática 8

José Creissac Campos, António Nestor Ribeiro
{jose.campos, anr}@di.uminho.pt
PPIII (LESI)

2002/03

Objectivos

1. Leitura de informação a partir de ficheiros.
2. Criação de programas interactivos.
3. Praticar a utilização de `see` e `tell`.
4. Praticar a utilização de `assert` e `retract`.

Leitura de informação a partir de ficheiros

A leitura de informação a partir de um ficheiro e o seu carregamento para a base de conhecimento do Prolog, obedece a um esquema geral, que se baseia na repetição da leitura até que se encontre o fim do ficheiro (representado pelo símbolo `end_of_file`. Simultâneamente à leitura a informação lida é processada (tarefa que é inerente à complexidade do problema) e gravada em memória (através do predicado `assert`).

O código padrão para este comportamento é apresentado de seguida:

```
carrega(F) :- see(F),
              repeat,
                read(T),
                processaTermoLido(T),
                T==end_of_file,
              !,
              seen.
```

```
processaTermoLido(end_of_file).
processaTermoLido(P) :- assert(P).  % P é o resultado da leitura.
                                    % Atenção que pode ser necessário
                                    % fazer algumas operações sobre P,
                                    % dependendo da lógica do problema.
```

Interactividade em programas Prolog

A forma de criação de um predicado que crie a noção de interacção com o utilizador, segue um padrão reutilizável, e utiliza o predicado `repeat` anteriormente utilizado.

```
menu :-
       carrega('ficheiros.dat'),
       repeat,
         write('*** MENU ***'),nl,
         write('1 - Opção 1'),nl,
         write('2 - Opção 2'),nl,
         ...
         write('6 - Opção 6'),nl,
         write('7 - Sair'),nl,
         write('Opção:'),nl,
         read(X),
         processa(X),
       X=7.                % se falhar volta ao repeat


processa(1) :-
  .....,
  !.

...
...
...

processa(7) :- !.
processa(_) :- write('Opção Inválida! Tente Novamente!').
```

Se quisermos efectuar validações ao tipo de entrada que queremos obter, pode ser utilizado o predicado `repeat` por forma a forçar leitura de valores que estejam dentro de uma determinada gama.

Para criar um predicado que valide a entrada de números inteiros, podemos escrever o código que se apresenta:

```
lerInteiro(I) :-
   repeat,
     read(I),
   integer(I),
   !.
```

Exercícios[1]

1. Num ficheiro encontram-se por ordem aleatória predicados que representam definições de figuras geométricas sob a forma: `quadrado(p, tam)`, `triangulo(p1, p2, p3)`, `rectangulo(p1, p2)` e `circunf(p, r)`:

   (a) Escreva o predicado `lerFiguras/1` que carrega para a Base de Conhecimento todas as figuras presentes no ficheiro.

   ```
   carrega(F) :-
         see(F),
         repeat,
           read(X),
           processa(X),
         X=end_of_file,
         !,
         seen.

   processa(end_of_file).
   processa(X):- assert(X).
   ```

   (b) Escreva o predicado `lerQuadrados/1` que carrega para a Base de Conhecimento todos os quadrados presentes no ficheiro.

   ```
   processaQuadrados(end_of_file).
   processaQuadrados(quadrado(ponto(X,Y),T)):-
           assert(quadrado(ponto(X,Y),T)).
   ```

[1] Exercícios baseados no Exame de 18 de Novembro de 1999.

(c) Escreva o predicado `maiorCircunf/2` que define qual a maior circunferência presente no ficheiro.

(d) Escreva ainda o predicado `listaQuad/2` que define a lista dos quadrados presentes na Base de Conhecimento que possuem como lado um valor dado.

2. Termos da forma `telefonou(pessoa1, pessoa2)` são utilizados numa Base de Conhecimento para indicar que uma dada pessoa contactou outra. Escreva um predicado que lhe permita de, forma interactiva, executar as seguintes operações:

   (a) Carregar para a BC tais factos a partir de um ficheiro F.

   (b) Determinar qual a pessoa que mais vezes ligou para a pessoa X.

   (c) Determinar qual o conjunto de pessoas que a pessoa Y contactou.

   (d) Determinar a lista de pares `(pessoa, lista_de_pessoas_contactadas)`.

   (e) Sabendo que cada chamada custa P, gravar num ficheiro, para cada pessoa, o custo das suas chamadas.