

Inteligência Artificial 2014-2015

Teste de Prolog. Duração: 1:30H

2014/12/18

Nota: Nos exercícios que se seguem, assume-se a utilização do Prolog standard. Sempre que nada for dito, podem usar-se todos os recursos existentes no Prolog (e.g., member/2).

As perguntas 1, 2 e 3 devem ser respondidas numa folha, e as 4, 5 e 6, noutra folha

I – Prolog Declarativo (quase, quase)

Os exercícios deste grupo não podem recorrer a mecanismos de controlo, por exemplo, o cut, o repeat, o fail, o assert e o retract. Pode, se for necessário, usar-se a negação por falha.

(3,5 Valores) 1 – Escreve o predicado *list_count/2* que recebe uma lista e computa o seu comprimento. Exemplo:

```
?- list_count([], X).  
X = 0  
  
?- list_count([a, b, [1, 2], c], X).  
X = 4
```

Não se pode usar o predicado *length/2*. A solução deve garantir que o primeiro argumento é uma lista e o segundo é um inteiro não negativo ou uma variável.

R:

```
list_count(L, N) :-  
    is_list(L),  
    (var(N); integer(N), N >= 0),  
    lcount(L, N).  
  
lcount([_|Rest], N) :-  
    lcount(Rest, N1),  
    N is N1 + 1.  
lcount([], 0).
```

(3,5 Valores) 2 – Escreve o predicado *oAttrib/4* que recebe informação sobre valores de vários atributos de vários objetos e devolve o atributo especificado de um objeto.

oAttrib(Information, Object, Attribute, X) significa que, de acordo com a informação em *Information*, *X* é o valor do atributo *Attribute* do objeto *Object*. O argumento *Information* é uma lista com informação sobre vários objetos. Cada elemento da lista diz respeito aos valores de um dado atributo de todos os objetos considerados.

Por exemplo, a lista *[cor:[o1:verde, o2:vermelho], formato:[o2:triângulo, o1:retângulo], cheiro:[o1:relva]]* representa informação sobre os objetos *o1* e *o2*. *o1* é um retângulo verde com cheiro a relva. *o2* é um triângulo vermelho sem informação sobre o cheiro.

Exemplo:

```
?- oAttrib([peso:[o2:5, o1:1], cheiro:[o1:relva]], o2, peso, X).  
X = 5
```

Não faça validações dos argumentos.

Nota: pensa-se que a informação sensorial do nosso cérebro está organizada de uma forma semelhante a esta.

R:

```
oAttrib(Information, Object, Attribute, Value):-  
    member(Attribute:AttribInfo, Information),  
    member(Object:Value, AttribInfo).
```

(2,5 Valores) 3 – Escreve o predicado *homogeneous_list_with_exception/5* para gerar listas de elementos todos iguais exceto um deles. *homogeneous_list_with_exception(N, E, I, X, List)* significa que *List* é uma lista com *N* elementos, os quais são todos iguais a *E*, exceto o elemento de índice *I*, que é igual a *X*. Exemplo:

```
?- homogeneous_list_with_exception(4, a, 1, b, L).  
L = [b, a, a, a]
```

Não faça validações dos argumentos.

R:

```
homogeneous_list_with_exception(N, E, I, X, [E|List]):-  
    N >= 1, I \= 1,  
    N1 is N - 1, I1 is I - 1,  
    homogeneous_list_with_exception(N1, E, I1, X, List).  
homogeneous_list_with_exception(N, E, 1, X, [X|List]):-  
    N >= 1,  
    N1 is N - 1,  
    homogeneous_list_with_exception(N1, E, 0, X, List).  
homogeneous_list_with_exception(0, _, _, _, []).
```

```
/* ALTERNATIVA  
homogeneous_list_with_exception(N, E, I, X, [E|List]):-  
    N > 1, I \= 1,  
    N1 is N - 1, I1 is I - 1,  
    homogeneous_list_with_exception(N1, E, I1, X, List).  
homogeneous_list_with_exception(N, E, 1, X, [X|List]):-  
    N > 1,  
    N1 is N - 1,  
    homogeneous_list_with_exception(N1, E, 0, X, List).  
homogeneous_list_with_exception(1, E, I, _, [E]):-  
    I \= 1.  
homogeneous_list_with_exception(1, _, 1, X, [X]).  
*/
```

(3,5 Valores) 4 – Escreve o predicado *identity_matrix/2* para gerar a matriz identidade com a dimensão especificada. Uma matriz identidade é uma matriz quadrada com todos os elementos iguais a 0, exceto os da diagonal principal, os quais são iguais a 1. Uma matriz é representada por uma lista de listas. Cada uma das listas da matriz representa uma linha.

Exemplo:

```
?- identity_natrix(3, M).
M = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]
```

Não faças validações dos argumentos.

Sugestão: usa o predicado *homogeneous_list_with_exception/5* para gerar cada uma das linhas da matriz identidade.

R:

```
identity_natrix(N, Matrix):-
    identity_natrix(N, N, 1, Matrix).
identity_natrix(N, Len, I, [Line|Rest]):-
    N > 0,
    homogeneous_list_with_exception(Len, 0, I, 1, Line),
    N1 is N - 1, I1 is I + 1,
    identity_natrix(N1, Len, I1, Rest).
identity_natrix(0, _, _, []).
```

II – Prolog Procedimental

Os exercícios deste grupo podem recorrer a todo e qualquer mecanismo de controlo, por exemplo, o cut, o repeat, o fail, o assert e o retract.

(3,5 Valores) 5 – Escreve o procedimento *print_object_description/1* que recebe o identificador de um objeto e imprime, no ecrã do computador, a informação disponível sobre os seus atributos.

Exemplo:

```
?- print_object_description(o3).
cor(o3) = vermelho
peso(o3) = 2.5
formato(o3)= retângulo
true
```

A informação disponível sobre os objetos é representada através de uma lista como a da pergunta 2, armazenada no facto do predicado *object_information/1*, por exemplo

```
object_information([peso:[o2:5, o1:1], cor:[o1:vermelho])).
```

Cada atributo possível está especificado num facto do predicado *attribute/1*, por exemplo

```
attribute(cor).
attribute(peso).
attribute(formato).
```

...

Não faças validações.

Sugestão: usa o predicado *oAttrib/4* para obter o valor de cada atributo do objeto.

Sugestão: se te der jeito, podes usar o procedimento *writelist/1* para imprimir os vários elementos de uma lista.

R:

```

print_object_description(Object):-
    object_information(Information),
    print_object_description(Information, Object).

print_object_description(Information, Object):-
    attribute(Attribute),
    oAttrib(Information, Object, Attribute, Value),
    print_object_attribute(Object, Attribute, Value),
    nl,
    fail.

print_object_description(_, _).

print_object_attribute(Object, Attribute, Value):-
    writelist([Attribute, '(', Object, ') = ', Value]).

```

(3,5 Valores) 6 – Escreve o procedimento *describe_objects/0* que descreve, no ecrã do computador, os objetos indicados pelo utilizador. O procedimento pede o nome de um objeto ao utilizador e descreve esse objeto. O procedimento repete este processo enquanto o utilizador não introduzir a palavra *fim*, em vez do nome de um objeto.

A impressão da descrição do objeto recorre ao predicado *print_object_description/*.

Exemplo:

```

?- describe_objects.
Objeto (ou fim, para terminar)> o3.
cor(o3) = vermelho
peso(o3) = 2.5
formato(o3)= retângulo

Objeto (ou fim, para terminar)> fim.
true

?-

```

Não faças validações.

R:

```

describe_objects:-
    repeat,
        read_object_name(Obj),
        process_object(Obj), !.
process_object(fim):-!.
process_object(Obj):-
    print_object_description(Obj), nl,
    fail.

read_object_name(Obj):-
    write('Objeto (ou fim, para terminar)> '),
    read(Obj).

```