

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Universidade do Porto
Faculdade de Engenharia

FEUP

EXERCÍCIOS DE PROGRAMAÇÃO EM LÓGICA

LUÍS PAULO REIS
DANIEL CASTRO SILVA

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E
COMPUTAÇÃO

PROGRAMAÇÃO EM LÓGICA - 3º ANO
SETEMBRO DE 2007



Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação
Programação em Lógica

2003/2004
LEIC
(3º Ano)
1º Sem

Docentes: Luís Paulo Reis e Eugénio da Costa Oliveira

Exercícios – Unificação e Predicados CUT e NOT

Exercício CN 1. Efeito do Cut

1) Considere a seguinte base de dados:

A :- B, C, !, D, E.

A :- F, G.

Supondo que as letras maiúsculas acima representam termos em PROLOG, qual o efeito do "cut" no conjunto de cláusulas acima ?

Exercício CN 2. Efeito do Cut 2

Considere a seguinte base de dados:

p(1) .

p(2) :- !.

p(3) .

Indique todas as respostas dadas pelo interpretador às questões:

a) ?- p(X).

b) ?- p(X), p(Y).

c) ?- p(X), !, p(Y).

Solução:

a) X=1 ; X=2

b) X=1 Y=1 ; X=1 Y=2 ; X=2 Y=1 ; X=2 Y=2

c) X=1 Y=1 ; X=1 Y=2

Exercício CN 3. Efeito do Cut 3

Suponha a seguinte base de factos em Prolog

dados(um) .

dados(dois) .

dados(tres) .

a) Qual o resultado da seguinte pergunta?

```
cut_teste_a(X) :-
```

```
    dados(X) .
```

```
cut_teste_a('ultima_clausula') .
```

```
?- cut_teste_a(X), write(X), nl, fail.
```

a) Qual o resultado do seguinte programa com um Cut no final da primeira clausula?

```
cut_teste_b(X):-  
    dados(X), !.  
cut_teste_b('ultima_clausula').  
?- cut_teste_b(X), write(X), nl, fail.
```

c) Qual o resultado do seguinte programa com um Cut no meio dos dois objectivos?

```
cut_teste_c(X,Y) :-  
    dados(X),  
    !,  
    dados(Y).  
cut_teste_c('ultima_clausula').  
?- cut_teste_c(X,Y), write(X-Y), nl, fail.
```

Solução:

a) um
 dois
 tres
 ultima_clausula
 no

b) um
 no

c) um - um
 um - dois
 um - tres
 no

Exercício CN 4. Maior de Três Números

Estude o seguinte programa que calcula o maior de entre três números:

```
max(X, Y, Z, X) :- X>Y, X>Z, !.  
max(X, Y, Z, Y) :- Y>X, Y>Z, !.  
max(_, _, Z, Z).
```

a) Diga em que situações o programa não funciona correctamente.

b) Corrija o programa.

Solução:

a) Quando X é igual a Y e X e Y são maiores que Z, Z é indicado como sendo o maior número.

b) `max2(X, Y, Z, X) :- X>=Y, X>=Z, !.`

`max2(X, Y, Z, Y) :- Y>=X, Y>=Z, !.`

`max2(_, _, Z, Z).`

Exercício CN 5. Unificação

Defina o predicado *unificavel(L1, Termo, L2)* em que *L2* é uma lista com todos os elementos de *L1* que são unificáveis com *Termo*. Os elementos de *L2* não são no entanto unificados com *Termo*. Exemplo:

```
?- unificavel([X,b,t(Y)],t(a),L).  
L=[X,t(Y)]
```

Note que se *Termo1=Termo2* resulta então *not(Termo1=Termo2)* falha e a instanciação resultante de *Termo1=Termo2* é anulada.

Solução:

```
unificavel([],_,[]).  
unificavel([T|Resto],T1,Resto1):-  
    not T=T1, !,  
    unificavel(Resto,T1,Resto1).  
unificavel([T|Resto],T1,[T|Resto1]):- unificavel(Resto,T1,Resto1).
```

Exercício CN 6. Cuts Verdes e Vermelhos

Explique a função dos ‘cuts’ incluídos no programa abaixo.

```
imaturo(X):- adulto(X), !, fail.  
imaturo(X).  
adulto(X):- pessoa(X), !, idade(X, N), N>=18.  
adulto(X):- tartaruga(X), !, idade(X, N), N>=50.
```

Solução:

O Cut em *Imaturo* funciona como no predicado *Not*. Se *X* for adulto então *(!, fail)* não é imaturo. Senão é imaturo. É um Cut Vermelho pois é essencial para o funcionamento do programa.

O Cut em *adulto* evita explorar espaço de pesquisa em que é impossível estar a solução. Se *X* for uma pessoa, então só será adulto se tiver uma idade maior ou igual a 18. Só se *X* não for uma pessoa é que se vai verificar se *X* é uma Tartaruga, ...

É um Cut verde pois não altera as soluções obtidas mas sim a eficiência do programa.