

Ficha Prática 6

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

2002/03

Objectivos

1. Cut e Fail. Exercícios.
2. Árvores de Prova (continuação).

Exercícios

Cut e Fail

1. Considere a seguinte definição:

$$f(X) = \begin{cases} 0 & \Leftarrow X < 3 \\ 2 & \Leftarrow X \geq 3 \wedge X < 6 \\ 4 & \Leftarrow X \geq 6 \end{cases}$$

Crie um predicado Prolog que faça o mesmo que esta definição.

2. Para a definição:

```
p(1).  
p(2):- !.  
p(3).
```

Diga qual o resultado das seguintes queries:

- `p(X)`.
- `p(X),p(Y)`.

- $p(X), !, p(Y)$.

Para cada uma das queries faça a respectiva árvore de prova.

3. Para o programa Prolog que se apresenta:

```

q(a).
q(b).
q(c).
r(b,b1).
r(c,c1).
r(a,a1).
r(a,a2).
r(a,a3).
p(X,Y):- q(X),r(X,Y).
p(d,d1).
p1(X,Y):- q(X),r(X,Y),!.
p2(X,Y):- q(X),!,r(X,Y).
p3(X,Y):- !,q(X),r(X,Y).
p3(d,d1).

```

efectue as árvores de prova para as queries seguintes:

- $p(X,Y)$.
 - $p1(X,Y)$.
 - $p2(X,Y)$.
 - $p3(X,Y)$.
4. (a) Crie um predicado `classe/2`, que determine se um determinado número é **positivo**, **negativo** ou **zero**.
- (b) Utilize o predicado definido anteriormente por forma a criar um outro, denominado `split/3`, que dada uma lista de números inteiros, origine duas outras listas, uma com os números positivos e zero e uma outra com os números negativos.
- (c) Faça a árvore de prova para a invocação `split([1,-1,3,0,-5,-2], LPositivosZero, LNegativos)`.
5. Crie um predicado `ifthenelse/3`, que implemente a estrutura condicional `if...then...else`.

6. Recorrendo ao uso de `cuts`, formule o predicado `interseccao/3` que implementa a intersecção de listas (vistas como conjuntos). Utilize o `cut` para melhorar a eficiência da procura de resultado.
7. Crie um predicado que dada uma lista de inteiros, devolva a lista dos inteiros que ocorrem mais do que duas vezes nessa lista. Mais uma vez, utilize o `cut` como mecanismo de eliminação de ramos da árvore que não sejam interessantes.
8. Crie um predicado `diferentes/2`, que dê verdadeiro se os dois parâmetros do predicados forem de facto diferentes.
9. Crie um predicado `diferenca/3`, que efectue a diferença de listas, isto é, cria uma lista com os elementos da primeira lista que não se encontram na segunda. Utilize o predicado `negação` definido na aula.