

## Capítulo 7

# Prolog (Corte e negação)

### 7.11 Corte

7.11.1. Considere o seguinte programa:

```
p(X,Y) :- q(X),r(Y).  
p(5,z).  
q(1).  
q(2).  
r(a).  
r(b).
```

- (a) Indique todas as respostas do PROLOG ao objetivo  $p(X,Y)$ .
- (b) Suponha que a primeira cláusula é substituída por  $p(X,Y) :- !,q(X),r(Y)$ . Indique todas as respostas do PROLOG ao objetivo  $p(X,Y)$ .
- (c) Suponha que a primeira cláusula é substituída por  $p(X,Y) :- q(X),!,r(Y)$ . Indique todas as respostas do PROLOG ao objetivo  $p(X,Y)$ .
- (d) Suponha que a primeira cláusula é substituída por  $p(X,Y) :- q(X),r(Y),!$ . Indique todas as respostas do PROLOG ao objetivo  $p(X,Y)$ .

7.11.2. Considere o seguinte programa:

```
p(2, 3).  
p(X,Y) :- q(X), r(Y).
```

```
q(1).
q(2).
q(3).
r(1).
r(3).
```

- (a) Indique todas as respostas do PROLOG ao objetivo  $p(X,Y)$ .
- (b) Introduza um operador de corte, de modo a só obter a primeira resposta.
- (c) Introduza um operador de corte, de modo a só obter as 3 primeiras respostas.
- (d) Introduza um operador de corte, de modo a só obter as 2 primeiras respostas.
- (e) Introduza um operador de corte, de modo a só obter as 5 primeiras respostas.

7.11.3. Defina o predicado `classe/2`, tal que `classe(N, C)`, em que  $N$  é um inteiro, significa que a classe de  $N$  é  $C$ , sendo o domínio de  $C$  o conjunto  $\{\text{zero}, \text{positivo}, \text{negativo}\}$ . Use o operador de corte sempre que tal melhorar o seu programa. Por exemplo,

```
?- classe(10, C).
C = positivo.
```

```
?- classe(0, C).
C = zero.
```

```
?- classe(-20, C).
C = negativo.
```

## 7.13 Negação

7.13.4. Defina o predicado `intersecao/3`, tal que `intersecao(L1, L2, I)`, em que  $L1$  e  $L2$  são listas de inteiros, significa que  $I$  é a lista com os elementos comuns a  $L1$  e  $L2$ . Use o operador de corte sempre que tal melhorar o seu programa. O seu programa não deve permitir que o pedido de múltiplas respostas origine respostas erradas. Por exemplo,

```
?- intersecao([1,2,3,4,5], [1,3,5], I).
I = [1, 3, 5].
```

7.13.5. Defina o predicado `disjuntas/2`, tal que `disjuntas(L1,L2)`, em que `L1` e `L2` são listas de inteiros, significa que `L1` e `L2` são disjuntas, isto é, não têm elementos em comum. Por exemplo,

```
?- disjuntas([1,2], [3,4,2]).
false.
```

```
?- disjuntas([1,2], [3,4]).
true.
```

Sugestão: use o predicado `membro` definido no livro.

- (a) Usando o corte e o falhanço forçado.
- (b) Usando a negação.

7.13.6. Considere o seguinte programa:

```
trabalha(P, D) :-
    pessoa(P),
    diaUtil(D),
    \+ temJust(P, D).

temJust(P, D) :-
    doente(P, D),
    \+ constipado(P, D).

diaUtil(D) :- membro(D, [seg, ter, qua, qui, sex]).
pessoa(rui).
pessoa(jaime).
pessoa(joana).
doente(rui, seg).
doente(jaime, qua).
constipado(jaime, qua).
```

- (a) Quais as respostas do PROLOG ao objetivo `?- trabalha(P, seg).?`
- (b) Qual o objetivo que origina a seguinte resposta do PROLOG ?

```
trabalha(rui,qua)
trabalha(jaime,qua)
trabalha(joana,qua)
false.
```

- (c) Quais as respostas do PROLOG ao objetivo  
?- \+ temJust(P, seg).?
- (d) Qual o objetivo que deveria ser dado ao PROLOG para responder  
à questão “Quem não tem justificação na segunda-feira?”?