

# Roteiro 6

Alexsandro Santos Soares

prof.asoares@gmail.com

Programação Lógica  
Faculdade de Computação  
Universidade Federal de Uberlândia

2 de julho de 2022

Este roteiro tem por finalidades:

- Familiarizá-lo com os predicados do Prolog que colecionam todas as soluções de um problema em uma única lista.
- Praticar o uso de predicados para manipulação de bases de conhecimento.

Ao fazer os exercícios **não** use qualquer predicado pré definido ou de alguma biblioteca que resolva diretamente o problema pedido.

## 1 Predicados com coleta de todas as soluções

Suponha que tenhamos um predicado Prolog descrevendo informação sobre países. Os fatos têm o formato *país(Nome, Continente, População, Fronteiras)*, onde *Nome* é o nome do País, *Continente* é o continente a que o país pertence (África, América, Ásia, Europa ou Oceania), *População* é um inteiro que representa o número de habitantes (em milhões) do país e *Fronteiras* é uma lista contendo os nomes dos países com os quais o país faz fronteira. Exemplo

```
país(alemanha, europa, 83, [frança, Bélgica, holanda, suíça]).
país(austrália, oceania, 25, []).
país(Bélgica, europa, 11, [frança, holanda, alemanha]).
país(espanha, europa, 47, [portugal, França]).
país(França, europa, 67, [espanha, suíça, Bélgica, alemanha,
    Itália]).
país(holanda, europa, 17, [Bélgica, alemanha]).
país(indonésia, oceania, 268, []).
país(Itália, europa, 60, [frança, suíça]).
país(madagascar, África, 26, []).
país(portugal, europa, 10, [espanha]).
país(suíça, europa, 8, [frança, alemanha, Itália]).
```

**Ex. 1** Escreva um predicado `pop_elevada(Continente, Lista)` que calcule a lista de todos os países com mais de 15 milhões de habitantes de um dado continente, ordenada por ordem crescente de população, no formato indicado.

Exemplos de consulta:

```
?- pop_elevada(europa, Lista).  
Lista = [47-espanha, 60-itália, 67-frança, 83-alemanha]  
  
?- pop_elevada(áfrica, Lista).  
Lista = [26-madagascar]
```

**Ex. 2** Escreva um predicado `isolados_grandes(Lista)` que calcule a lista, ordenada por ordem alfabética, de todos os continentes que possuem pelo menos dois países que tenham simultaneamente uma população superior a 15 milhões e duas ou menos fronteiras terrestres (com países conhecidos).

Exemplo:

```
?- isolados_grandes(Lista).  
Lista = [europa, oceania]  
% europa pois possui a espanha e a itália;  
% oceania devido à austrália e indonésia.
```

**Ex. 3** Assuma que se tenha a seguinte base de dados:

```
q(blob,blug).  
q(blob,blag).  
q(blob,blig).  
q(blaf,blag).  
q(dang,dong).  
q(dang,blug).  
q(flaf,blob).
```

Qual é a resposta do Prolog às seguintes consultas:

- (a) `findall(X, q(blob,X), Lista).`
- (b) `findall(X, q(X,blug), Lista).`
- (c) `findall(X, q(X,Y), Lista).`
- (d) `bagof(X, q(X,Y), Lista).`
- (e) `setof(X, Y~q(X,Y), Lista).`

**Ex. 4** Dada uma base de dados com vários fatos sobre pessoas e suas idades. Por exemplo:

```
idade(pedro, 8).  
idade(ana, 5).  
idade(paula, 8).  
idade(ricardo,10).
```

```
idade(carla,5).
idade(amanda,12).
idade(sara,15).
idade(débora,14).
idade(joaquim,12).
idade(gabriel,5).
```

Escreva uma consulta que devolva:

- (a) uma lista com todas as pessoas com 5 anos de idade.
- (b) uma lista com as idades de todas as pessoas em ordem crescente e sem duplicações.
- (c) uma lista com os nomes de todas as pessoas.
- (d) uma lista com os nomes de todas as pessoas cuja idades estejam no intervalo fechado de 8 a 12 anos.

## 2 Exercícios sobre conjuntos

**Ex. 5** Conjuntos podem ser pensados como listas que não contenham elementos repetidos. Por exemplo,  $[a,4,6]$  é um conjunto, mas  $[a,4,6,a]$  não é, pois ele contém duas ocorrências de  $a$ .

Escreva um programa Prolog `subconjunto/2` que é satisfeito quando o primeiro argumento é um subconjunto do segundo argumento, isto é, quando qualquer elemento do primeiro argumento é um membro do segundo argumento. Por exemplo:

```
?- subconjunto([a,b],[a,b,c]).
true
```

```
?- subconjunto([c,b],[a,b,c])
true
```

```
?- subconjunto([], [a,b,c])
true
```

Seu programa deveria ser capaz de gerar todos os subconjuntos de um conjunto dado como entrada via retrocesso. Por exemplo, se você der como entrada

```
?- subconjunto(S,[a,b,c]).
```

ele deveria gerar sucessivamente todos os oitos subconjuntos de  $[a,b,c]$ .

**Ex. 6** Usando o predicado `subconjunto` que acabou de escrever, e `findall`, escreva um predicado `conj_potência/2` que recebe um conjunto como seu primeiro argumento e retorna o conjunto potência deste conjunto como o segundo argumento. O conjunto potência de um conjunto é o conjunto de todos os seus subconjuntos. Por exemplo:

```
?- conj_potência([a,b,c],P).
P = [[], [a], [b], [c], [a,b], [a,c], [b,c], [a,b,c]]
```

Não importa se os conjuntos são retornados em uma ordem diferente da anterior. Por exemplo,

```
P = [[a],[b],[c],[a,b,c],[],[a,b],[a,c],[b,c]]
```

também é válido.

### 3 Exercícios sobre manipulação de bases de fatos

**Ex. 7** Assuma que se inicie com uma base de dados vazia. Então, dado o comando:

```
?- assert(q(a,b)), assertz(q(1,2)), asserta(q(foo,blug)).
```

O que estará na base de dados agora?

Na sequência é dado o comando:

```
?- retract(q(1,2)), assertz( (p(X) :- h(X)) ).
```

O que estará na base de dados agora?

Por fim, entre com o comando:

```
?- retract(q(_,_)),fail.
```

O que estará na base de dados agora?

**Ex. 8** Escreva um predicado `somatório/2` que recebe um inteiro  $n > 0$  e calcula a soma de todos os inteiros entre 1 e  $n$ . Exemplo:

```
?- somatório(3,X).
```

```
X = 6
```

```
?- somatório(5,X).
```

```
X = 15
```

Escreva o predicado tal que os resultados sejam guardados na base de dados (é claro que não deveria haver mais que uma entrada por resultado na base de dados para cada valor) e reutilizados sempre que possível. Assim, por exemplo:

```
?- somatório(2,X).
```

```
X = 3
```

```
?- listing.
```

```
res_somatório(2,3).
```

Depois disto, quando realizarmos a consulta

```
?- somatório(3,X).
```

O Prolog não calculará tudo de novo, mas obterá o resultado `somatório(2,3)` da base de dados e somente somará 3 a este resultado. O Prolog então responderá:

```
X = 6
```

```
?- listing.
```

```
res_somatório(2,3).
```

```
res_somatório(3,6).
```

## 4 Sugestões de leitura

- Luiz A. M. Palazzo. *Introdução à programação Prolog*  
<http://puig.pro.br/Logica/palazzo.pdf>
- Eloi L. Favero. *Programação em Prolog: uma abordagem prática*  
<http://www3.ufpa.br/favero>
- Wikilivro sobre Prolog em  
<http://pt.wikibooks.org/wiki/Prolog>
- Patrick Blackburn, Johan Bos and Kristina Striegnitz. *Learn Prolog Now!*  
<http://www.learnprolognow.org>