

# Roteiro 8

Alexsandro Santos Soares

`prof.asoares@gmail.com`

Programação Lógica

Faculdade de Computação

Universidade Federal de Uberlândia

16 de julho de 2022

Este roteiro tem por finalidades:

- Familiarizá-lo com DCGs simples e com aquelas que utilizam argumentos e testes adicionais.

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

## 1 Gramáticas de cláusulas definidas com argumentos e testes extras

Primeiro alguns exercícios de fixação:

- Ex. 1** Rastreie alguns exemplos de DCG que utilizem argumentos extras para tratar a distinção sujeito/objeto, de DCG que produza análise sintática e de DCG que utilize testes extras para separar o léxico das regras. Certifique-se de que você compreenda totalmente o modo com o qual todas as três DCGs funcionam. Use os slides da aula teórica dessa semana.
- Ex. 2** Realize alguns rastreamentos para a DCG para a linguagem  $a^n b^n c^n$ . Experimente casos onde os três blocos de *as*, *bs* e *cs* sejam de fato do mesmo tamanho, assim como casos onde isto não ocorre.

Agora alguns exercícios para treinar a técnica.

- Ex. 3** A linguagem formal  $a^n b^{2m} c^{2m} d^n$  consiste de todas as strings da seguinte forma: um bloco contíguo de *as* seguido por um bloco contíguo de *bs*, seguido por um bloco contíguo de *cs*, seguido por um bloco contíguo de *ds*, tal que os blocos *a* e *d* são exatamente do mesmo tamanho, e os blocos *b* e *c* são exatamente do mesmo tamanho e, além disto, consistem de um número par de *bs* e *cs*. Por exemplo,  $\epsilon$ , *abbccd* e *aaabbbbccccddd* pertencem a  $a^n b^{2m} c^{2m} d^n$ . Escreva uma DCG que gere esta linguagem.
- Ex. 4** Abaixo encontra-se a nossa DCG básica.

```

s --> sn, sv.

sn --> det, n.

sv --> v, sn.
sv --> v.

det --> [o].
det --> [a].

n --> [mulher].
n --> [homem].

v --> [bate].

```

Suponha que adicionemos o nome “homens”, que é plural, e o verbo “batem”. Então, gostaríamos de uma DCG que diga que “O homem bate” está correto, “Os homens batem” está correto, “O homem batem” não está correto e que “Os homens bate” também não está correto. Altere a DCG tal que ela corretamente trate estas sentenças. Use um argumento extra para lidar com a distinção singular/plural.

**Ex. 5** Traduza a seguinte regra DCG em um formato padrão de regras do Prolog:

```
cangu(V,R,Q) --> ru(V,R), salta(Q,Q), {marsupial(V,R,Q)}.
```

Finalmente, alguns exercícios de programação.

**Ex. 6** Primeiro, reúna todas as coisas que aprendeu sobre DCGs para Português em uma única DCG. Em particular, nessa semana vimos como usar argumentos extras para lidar com a distinção sujeito/objeto, e nos exercícios anteriores você usou argumentos adicionais para lidar com a distinção singular/plural. Escreva uma DCG que trate ambos. Além disto, escreva a DCG de tal forma que ela produza árvores sintáticas e faça uso de um léxico separado.

**Ex. 7** Escreva uma DCG que reconheça numerais cardinais escritos por extenso em Português, entre zero e mil.

```

?- cardinal([zero], []).
true

?- cardinal([vinte,e,um], []).
true

?- cardinal([novecentos,e,trinta,e,sete], []).
true

?- cardinal([setecentos, e, setenta, e, sete], []).
true.

```

```
?- cardinal([mil], []).
true.

?- cardinal([cem,onze], []).
false.
```

**Ex. 8** Modifique o reconhecedor do exercício anterior para que ele também produza os dígitos do número reconhecido em uma lista.

```
?- cardinal(N, [zero], []).
N = [0] .

?- cardinal(N, [vinte,e,um], []).
N = [2, 1] .

?- cardinal(N, [novecentos,e,trinta,e,sete], []).
N = [9, 3, 7] .

?- cardinal(N, [novecentos,e,trinta], []).
N = [9, 3, 0] .

?- cardinal([7,7,7], Extenso, []).
Extenso = [setecentos, e, setenta, e, sete] .

?- cardinal([1,0,0,0], Extenso, []).
Extenso = [mil].
```

**Ex. 9** Escreva uma DCG que reconheça todos os numerais ordinais em português entre 1 e 1000. Para recordar sobre os numerais ordinais use:

<https://www.normaculta.com.br/numerais-ordinais/>

Teste o reconhecedor com:

```
?- phrase(ordinal, [nono]).
true.

?- phrase(ordinal, [nove]).
false.

?- phrase(ordinal, [vigésimo]).
true.

?- phrase(ordinal, [vigésimo, quinto]).
true.

?- phrase(ordinal, [trecentésimo, sexagésimo, terceiro]).
true.

?- phrase(ordinal, [nongentésimo, septuagésimo, sétimo]).
```

```
true.  
  
?- phrase(ordinal, [setuagésimo, primeiro]).  
true.
```

**Ex. 10** Modifique a DCG do exercício anterior para que ela retorne uma lista com os Algarismos do numeral ordinal reconhecido.

```
?- phrase(ordinal(Algarismos), [nono]).  
Algarismos = [9]  
  
?- phrase(ordinal(Algarismos), [nove]).  
false.  
  
?- phrase(ordinal(Algarismos), [vigésimo]).  
Algarismos = [2, 0].  
  
?- phrase(ordinal(Algarismos), [vigésimo, quinto]).  
Algarismos = [2, 5].  
  
?- phrase(ordinal(Algarismos), [trecentésimo, sexagésimo, terceiro]).  
Algarismos = [3, 6, 3].  
  
?- phrase(ordinal(Algarismos), [nongentésimo, septuagésimo, sétimo]).  
Algarismos = [9, 7, 7].  
  
?- phrase(ordinal(Algarismos), [setuagésimo, primeiro]).  
Algarismos = [7, 1].
```

**Ex. 11** Escreva um predicado que receba um número natural positivo e devolva em seu segundo argumento a sua lista de dígitos:

```
?- dígitos(777, Ds).  
Ds = [7, 7, 7].  
  
?- dígitos(584, Ds).  
Ds = [5, 8, 4].
```

**Ex. 12** Use os dois exercícios anteriores para criar um predicado que escreva por extenso um numeral ordinal entre 1 e 1000:

```
?- ordinal_por_extenso(777, Extenso).  
Extenso = [ septingentésimo, septuagésimo, sétimo ].  
  
?- ordinal_por_extenso(584, Extenso).  
Extenso = [ quingentésimo, octogésimo, quarto ].  
  
?- ordinal_por_extenso(363, Extenso).  
Extenso = [ trecentésimo, sexagésimo, terceiro ].
```

**Ex. 13** Considere o arquivo HTML5 abaixo:

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <title>Título da Página Básica</title>
  </head>
  <body>
  </body>
</html>
```

Escreva uma gramática DCG que analise o arquivo acima e devolva o título da página. Para facilitar a testagem digite um predicado de teste junto com a DCG:

```
teste([
  <, !, 'DOCTYPE', html, >,
  <, html, lang, =, 'pt-BR', >,
    <, head, >,
      <, meta, charset, =, 'UTF-8', >,
      <, title, >, 'Título', da, 'Página', 'Básica', <, /, title, >,
    <, /, head, >,
    <, body, >,
    <, /, body, >,
  <, /, html, >
]).
```

Teste a sua gramática com:

```
?- teste(Página), phrase(html(Título), Página).
Título = ['Título', da, 'Página', 'Básica' ]
```

## 2 Sugestões de leitura

- Luiz Arthur Pagani. (2004). *Analizador gramatical em Prolog para gramáticas de estrutura sintagmática*. Revista Virtual de Estudos da Linguagem -ReVEL. Ano 2, n. 3. Disponível em:

[http://www.revel.inf.br/files/artigos/revel\\_3-analisador\\_gramatical.pdf](http://www.revel.inf.br/files/artigos/revel_3-analisador_gramatical.pdf)

- 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>