

FACULDADE: CENTRO UNIVERSITÁRIO DE BRASÍLIA – UniCEUB
CURSO: CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA: CONSTRUÇÃO DE COMPILADORES
CARGA HORÁRIA: 75 H. A. **ANO/SEMESTRE:** 2016/02
PROFESSOR: EDUARDO FERREIRA DOS SANTOS
HORÁRIOS: Quartas e Sextas às 07h40

LABORATÓRIO 01 – FERRAMENTA ANTLR

RESUMO

A construção de compiladores é uma disciplina da computação que tem como objetivo transformar linguagem de programação de alto nível em linguagem de máquina. O ANTLR é uma ferramenta de *parsing* construída em Java que pode ser utilizada para ler, processar, executar ou traduzir texto estruturado em arquivos binários. A utilização da ferramenta ANTLR vai facilitar a implementação de uma gramática de forma a construir um compilador.

OBJETIVOS

Objetivo Geral

Conhecer a ferramenta ANTLR e suas possibilidades de utilização.

Objetivos Específicos

1. Instalar a ferramenta ANTLR;
2. Construir uma ferramenta de parsing simples para uma gramática definida

EXERCÍCIO 01 – EXEMPLO DE UTILIZAÇÃO

- Todos os exemplos abordados no guia podem ser encontrados na página oficial do livro: https://pragprog.com/titles/tpantlr2/source_code
 - Vamos aprender a utilizar a ferramenta ANTLR utilizando o tutorial básico disponível no seguinte endereço: <https://github.com/antlr/antlr4/blob/master/doc/getting-started.md>
1. Construa a árvore de parsing apresentada no exercício do primeiro tutorial.

EXERCÍCIO 02 – A FERRAMENTA ANTLR

A ferramenta ANTLR é baseada em dois componentes principais: o ANTLR em si e a API da ferramenta de *parsing* em tempo de execução. Ao dizer “rodar o ANTLR em uma gramática”, estamos nos referindo a utilizar uma das API’s da ferramenta para reconhecer sentenças descritas na linguagem.

EXERCÍCIO 02 – A FERRAMENTA ANTLR

Considere a seguinte gramática para descrever os elementos de um array:

```
/** Grammars always start with a grammar header. This grammar is called
 * ArrayInit and must match the filename: ArrayInit.g4
 */
grammar ArrayInit;

/** A rule called init that matches comma-separated values between {...}. */
init : '{' value (',' value)* '}' ; // must match at least one value

/** A value can be either a nested array/struct or a simple integer (INT) */
value : init
      | INT
      ;

// parser rules start with lowercase letters, lexer rules with uppercase
INT :
[0-9]+ ;

// Define token INT as one or more digits
WS :
[ \t\r\n]+ -> skip ; // Define whitespace rule, toss it out
```

Ao realizar rodar o ANTLR na gramática ArrayInit uma série de arquivos é gerado, conforme exemplificado na Figura 1.

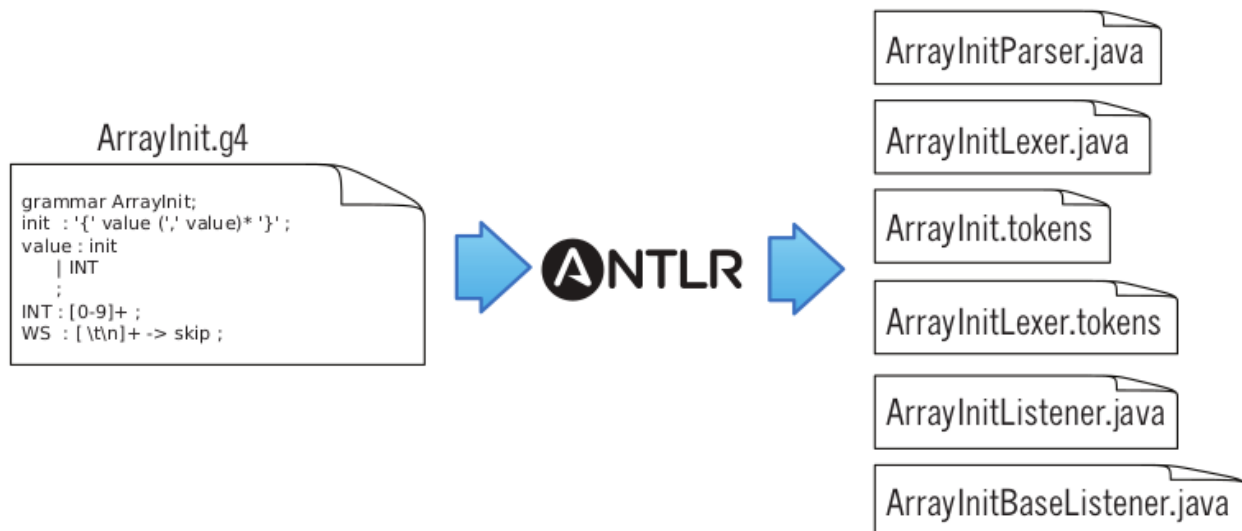


Figura 1: Parsing de um array no ANTLR

2. Analise a saída do ANTLR para o parsing do seguinte Array: {99, 3, 451}
 - a) Descreva a função de cada um dos arquivos produzidos e qual seu papel na compilação da sentença da gramática;
 - b) Descreva o significado de cada uma das linhas da saída.
3. Construa uma árvore de parsing para o ANTLR utilizando o mesmo Array.

EXERCÍCIO 03 – PARSING

Considere a seguinte gramática no formato simplificado:

```
<SENTENCE>    → <NOUN-PHRASE><VERB-PHRASE>
<NOUN-PHRASE> → <CMLPX-NOUN> | <CMLPX-NOUN> <PREP-PHRASE>
<VERB-PHRASE> → <CMLPX-VERB> | <CMLPX-VERB> <PREP-PHRASE>
<PREP-PHRASE> → <PRE> <CMLPX-NOUN>
<CMLPX-NOUN>  → <ARTICLE> <NOUN>
<CMLPX-VERB> → <VERB> | <VERB> <NOUN-PHRASE>
<ARTICLE>    → a | the
<NOUN>       → boy | girl | flower
<VERB>       → touches | likes | sees
<PREP>       → with
```

Utilize o ANTLR para construir a árvore de parsing da gramática fornecida.

BIBLIOGRAFIA

PARR, Terence. **The definitive ANTLR 4 reference**. Pragmatic Bookshelf, 2013. Disponível em <http://www4.di.uminho.pt/~gepl/GQE/documents/books/Pragmatic.The.Definitive.ANTLR.4.Reference.Jan.2013.pdf>

AHO, Alfred V. E Outros. **Compiladores: princípios, técnicas e ferramentas**. PEARSON, 2007.

PRICE, Ana Maria de Alencar. **Implementação de linguagens de programação: compiladores**. SAGRA-LUZZATTO, 2005.