

1ª Aula Prática de Compiladores: Análise Léxica e Sintática com ANTLR 4

Roteiro

- ANTLR
- Instalação
- Estrutura da Gramática
- Exercício Prático

ANTLR

- ANTLR (ANother Tool for Language Recognition)
- Gerador de lexers/parsers escrito em Java
- Sintaxe simples e poderosa
- Open-source (BSD)
- Utilizado por: Oracle (SQL Developer), Twitter (query parsing), Hadoop (Hive/Pig)...

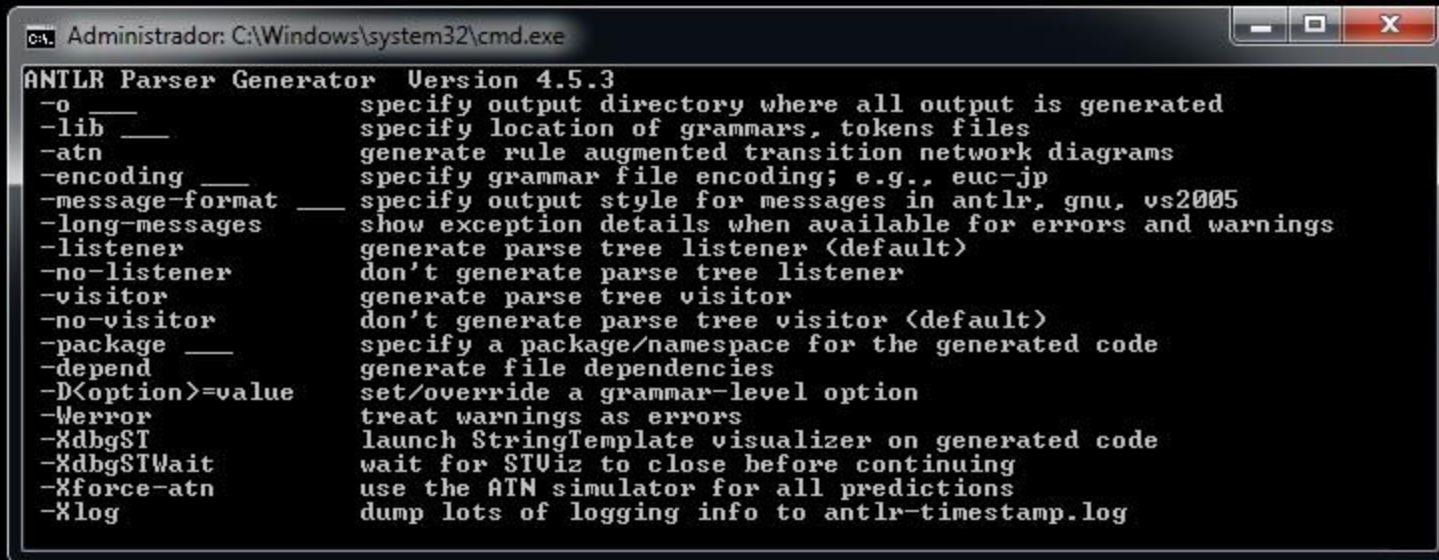
ANTLR

- Página Inicial do ANTLR:
<http://www.antlr.org/>
- Repositório do Projeto do ANTLR:
<https://github.com/antlr>
- Documentação do ANTLR 4:
<https://github.com/antlr/antlr4/blob/master/doc/index.md>

ANTLR

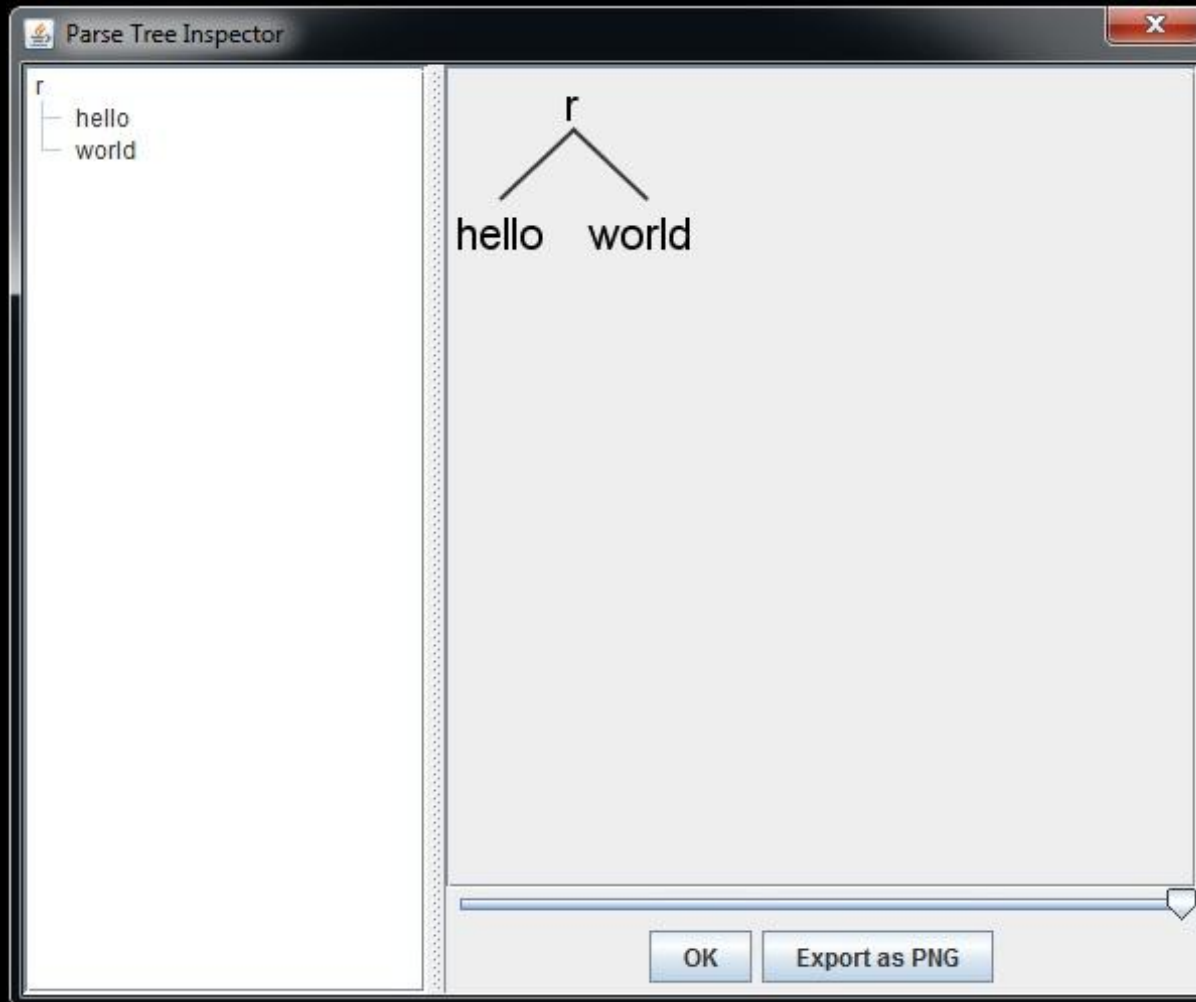
- Seções Importantes da Documentação:
 - [Getting Started with ANTLR v4](#)
 - [Grammar Lexicon](#)
 - [Grammar Structure](#)
 - [Parser Rules](#)
 - [Lexer Rules](#)
 - [Wildcard Operator and Nongreedy Subrules](#)

ANTLR (Versão 4.5.3)



```
Administrator: C:\Windows\system32\cmd.exe
ANTLR Parser Generator  Version 4.5.3
-o _____ specify output directory where all output is generated
-lib _____ specify location of grammars, tokens files
-atn            generate rule augmented transition network diagrams
-encoding _____ specify grammar file encoding; e.g., euc-jp
-message-format _____ specify output style for messages in antlr, gnu, vs2005
-long-messages show exception details when available for errors and warnings
-listener       generate parse tree listener (default)
-no-listener    don't generate parse tree listener
-visitor        generate parse tree visitor
-no-visitor     don't generate parse tree visitor (default)
-package _____ specify a package/namespace for the generated code
-depend         generate file dependencies
-D<option>=value set/override a grammar-level option
-Werror        treat warnings as errors
-XdbgST        launch StringTemplate visualizer on generated code
-XdbgSTWait    wait for STViz to close before continuing
-Xforce-atn    use the ATN simulator for all predictions
-Xlog          dump lots of logging info to antlr-timestamp.log
```

ANTLR (Parse Tree Inspector)



Instalação

- Façam o download do ANTLR Versão 4.5.3 a partir do link abaixo:
<http://www.antlr.org/download/antlr-4.5.3-complete.jar>
- Confiram as instruções do arquivo README.txt na pasta compilers-cin\aulas-praticas\ap1\ do repositório da disciplina

Estrutura da Gramática

```
/** javadoc opcional */
```

```
grammar Test;
```

```
/* Regras do lexer/parser, possivelmente  
mescladas */
```

```
rule1 : ... ;
```

```
rule2 : ... | ... ;
```

```
...
```

```
ruleN : ...
```

Estrutura da Gramática

- Operadores:

. //Operador coringa
~ //Negação
EOF //Token de End Of File

- Quantificadores:

* e *? //Nenhuma ou mais instâncias
+ e +? //Apenas uma ou mais instâncias
? e ?? //Nenhuma ou apenas uma instância

Estrutura da Gramática

- Quantificadores Gulosos:

*

+

?

- Quantificadores Não Gulosos:

*?

+?

??

- Loops Gulosos vs Loops Não Gulosos

Estrutura da Gramática

- Um arquivo do ANTLR pode conter apenas definições léxicas, ou apenas definições sintáticas ou ambas
- Definições léxicas devem começar obrigatoriamente com letra maiúscula

```
INT    :  [0-9]+;
```

```
CHAR   :  '\'' [a-zA-Z] '\'';
```

```
WS     :  [ \t\r\n]+ -> skip;
```

Estrutura da Gramática

- Definições sintáticas começam com letra minúscula
- O operador '|' pode ser utilizado para separar múltiplas produções da mesma regra

```
retstat : 'return' expr ';' ;
stat    : retstat
        | 'break' ';'
        | 'continue' ';'
        ;
```

Estrutura da Gramática

- Tanto as definições léxicas como as sintáticas têm precedência uma sobre as outras de acordo com a ordem de suas declarações
- Dada as definições léxicas abaixo, a palavra de entrada 'if' seria tokenizada como um token do tipo IF, uma vez que a definição de IF precede a definição de ID

IF: 'if';

ID: [a-z]+;

Estrutura da Gramática

- Subregras

```
returnType : (type | 'void');
```

- Fragmentos

```
//Não constitui um token por si só
```

```
fragment DIGIT : [0-9];
```

```
//Referencia o fragmento DIGIT
```

```
INT : DIGIT+;
```

Estrutura da Gramática

- Ações e atributos

```
decl : type ID {println("found a decl");} ;
```

- Comandos Léxicos

```
WS : [ \t\r\n]+ -> skip;
```

- Geração de código

- Apenas Java no ANTLR 4!

Estrutura da Gramática (Exemplo)

```
grammar Exp;
```

```
s : e;
```

```
e : e '*' e
```

```
   | e '+' e
```

```
   | INT
```

```
;
```

```
INT : [0-9]+;
```

```
WS  : [ \t\r\n]+ -> skip;
```

Estrutura da Gramática (Exemplo)

grammar Exp;

s : e;

e : e '*' e
| e '+' e
| INT
;



'*' tem prioridade sobre '+', pois sua produção foi declarada primeiro.

INT : [0-9]+;

WS : [\t\r\n]+ -> skip;

Estrutura da Gramática (Exemplo)

```
grammar Exp;
```

```
s : e;
```

```
e : e '*' e
```

```
  | e '+' e
```

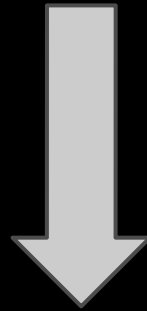
```
  | INT
```

```
;
```

```
INT : [0-9]+;
```

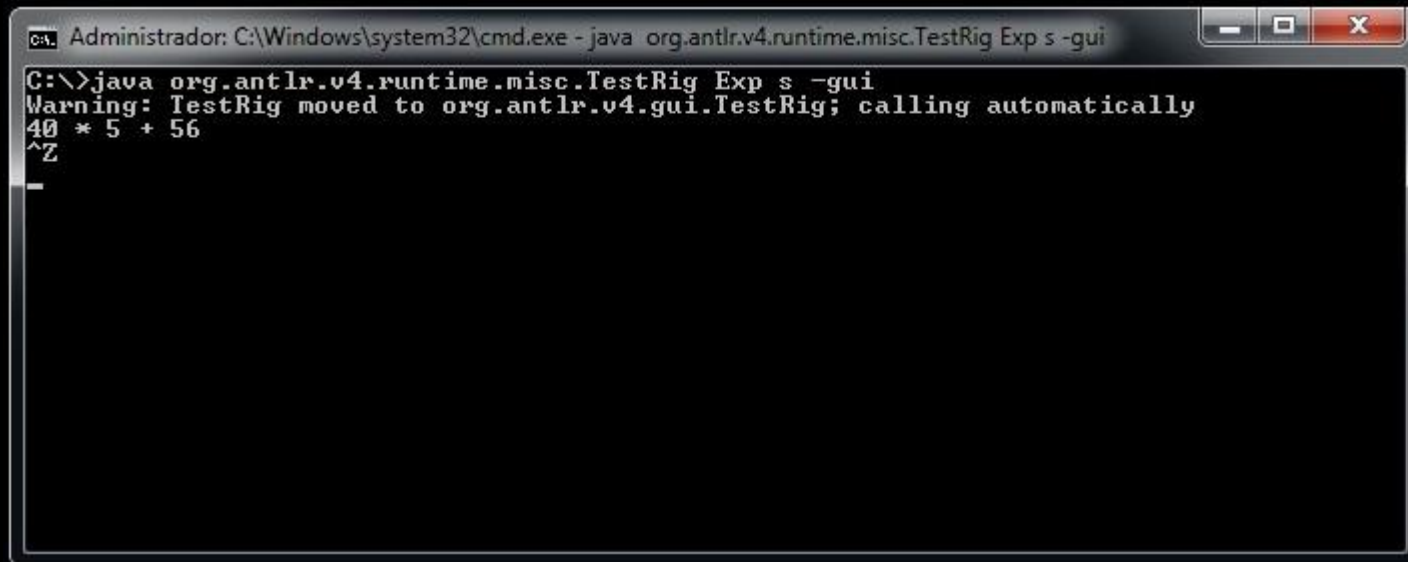
```
WS  : [ \t\r\n]+ -> skip;
```

Descarta espaços,
tabulações, quebras
de linha, etc..



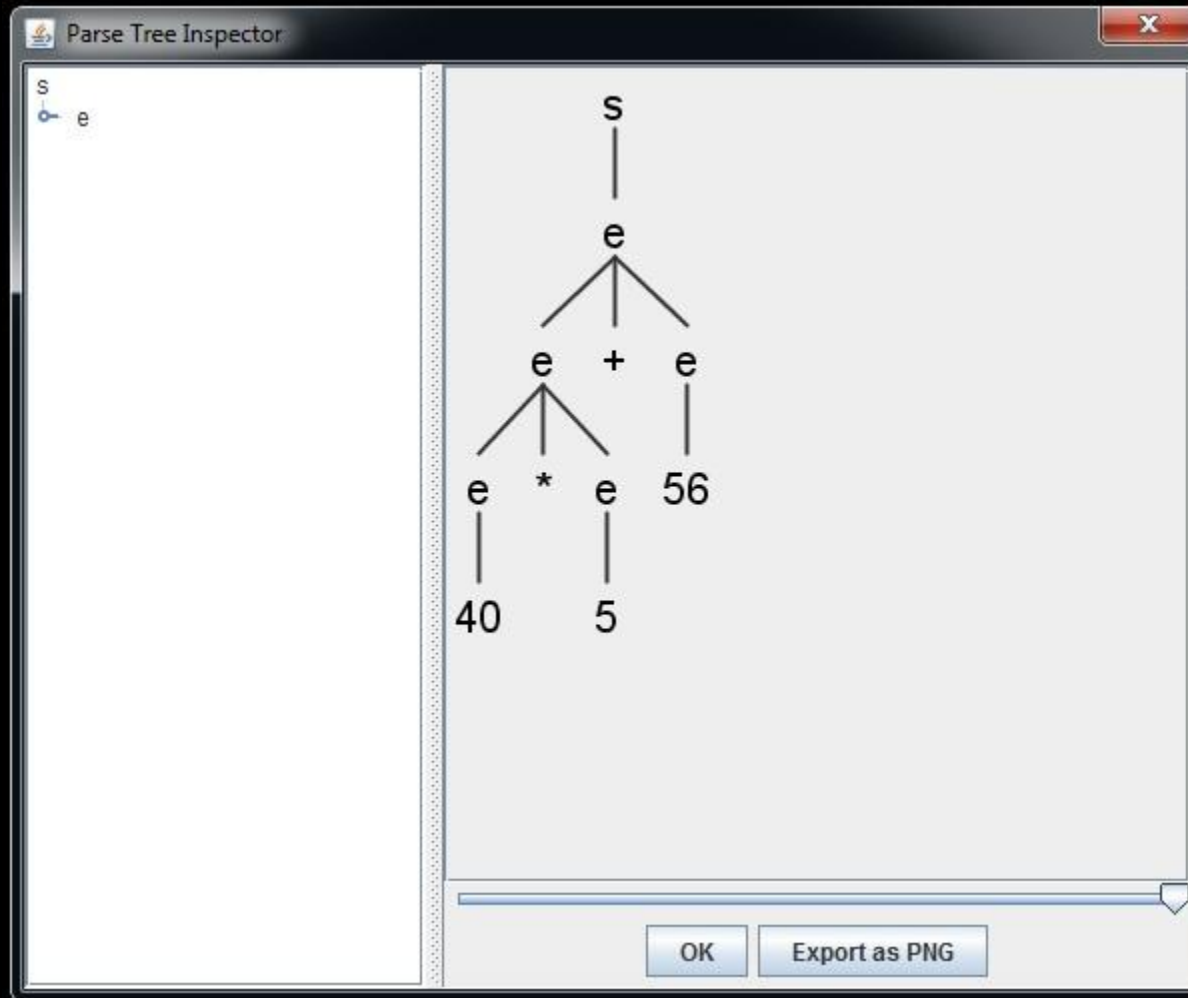
Estrutura da Gramática (Exemplo)

- Testando a gramática Exp com a expressão “40 * 5 + 56”



```
Administrator: C:\Windows\system32\cmd.exe - java org.antlr.v4.runtime.misc.TestRig Exp s -gui
C:\>java org.antlr.v4.runtime.misc.TestRig Exp s -gui
Warning: TestRig moved to org.antlr.v4.gui.TestRig; calling automatically
40 * 5 + 56
^Z
-
```

Estrutura da Gramática (Exemplo)



Exercício Prático

1. Modifique a gramática Exp para aceitar:

- As operações de subtração e divisão

Ex.:

$$9 - 16 / 4;$$

- Expressões com parênteses

Ex.:

$$9 - (8 * 2 / 4);$$

- Mais de uma expressão separada por ';'

Ex.:

$$2 + 3; 4 + 5 + 8;$$

Exercício Prático

- Comentários de linha

Ex.:

```
//blablabla123...
```

- Comentários de bloco

Ex.:

```
/*comentario de  
bloco */
```

- Atribuições de valores a variáveis

Ex.:

```
x = 3 * 4 + 5;  
y = z + 10;
```

Exercício Prático

- O exercício prático deve ser realizado individualmente e enviado por e-mail com o assunto “EXERCÍCIO PRÁTICO 01” para monitoria-if688-l@cin.ufpe.br até às 23:59 de domingo (01.04.2018)
- A resolução do exercício prático deve estar em um arquivo com o nome “AP1.g4”