

Caneca++

Chrystian Guth
Lucas Pereira
Renan Netto

Antlr

- Analisador descendente recursivo com **backtracking**.
- Analisador descendente recursivo preditivo **LL(k)** com o **k** definido pelo usuário.
- Analisador descendente recursivo preditivo **LL(*)** onde não há restrição quanto ao **k**, mas exige que a não fatoração seja uma gramática regular.

Gramática sintática

`fontes/g/CanecaSintatico.g`

```
parser grammar CanecaSintatico;
```

```
options {  
    tokenVocab = CanecaLexico;  
    language = Java;  
    backtrack = false;  
    k = 1;  
}
```

```
@header {  
    package br.ufsc.inf.ine5426.canecaantlr;  
}
```

Gramática sintática

programa : (cabecalho (classe | interfaces))+ ;

cabecalho : pacote (importacao)* ;

pacote : PACOTE IDENTIFICADOR_DE_PACOTE TERMINADOR ;

importacao : IMPORTE (IDENTIFICADOR
IDENTIFICADOR_DE_PACOTE (COMO IDENTIFICADOR)? |
IDENTIFICADOR_DE_PACOTE) TERMINADOR ;

classe : CLASSE modificadorDeAcessoFeminino
definicaoDeTipo (listaDeInterfaces)? CorpoDaClasse ;

Gramática sintática

interfaces : INTERFACE modificadorDeAcessoFeminino
definicaoDeTipo (listaDeInterfaces)? corpoDaInterface ;

corpoDaClasse : INICIO (atributo)* (construtor)+ (destrutor)+
(metodo)* FIM ;

corpoDaInterface : INICIO (assinaturaDeMetodo TERMINADOR)*
FIM ;

metodo : assinaturaDeMetodo blocoDeInstrucoes ;

assinaturaDeMetodo : METODO (ESTATICO)?
modificadorDeAcessoMasculino tipo IDENTIFICADOR
listaDeArgumentos ;

construtor : CONSTRUTOR modificadorDeAcessoMasculino
IDENTIFICADOR listaDeArgumentos blocoDeInstrucoes ;

Gramática sintática

destrutor : DESTRUTOR modificadorDeAcessoMasculino
IDENTIFICADOR listaDeArgumentos blocoDeInstrucoes ;

atributo : ATRIBUTO (ESTATICO)?
modificadorDeAcessoMasculino tipo IDENTIFICADOR
(atribuicao)? TERMINADOR ;

modificadorDeAcessoMasculino : PUBLICO | PRIVADO |
PROTEGIDO ;

modificadorDeAcessoFeminino : PUBLICA | PRIVADA |
PROTEGIDA ;

listaDeParametros : PARENTESE_ESQUERDO (expressao
(SEPARADOR expressao)*)? PARENTESE_DIREITO ;

Gramática sintática

`listaDeArgumentos` : PARENTESE_ESQUERDO (definicaoDeVariavel
(SEPARADOR definicaoDeVariavel)*)? PARENTESE_DIREITO ;

`listaDeInterfaces` : IMPLEMENTA tipo (SEPARADOR tipo)* ;

`definicaoDeTipo` : IDENTIFICADOR (COLCHETE_ESQUERDO
IDENTIFICADOR (listaDeInterfaces)? COLCHETE_DIREITO)* ;

`tipo` : IDENTIFICADOR (COLCHETE_ESQUERDO tipo (SEPARADOR tipo)*
COLCHETE_DIREITO)? ;

`definicaoDeVariavel` : tipo IDENTIFICADOR ;

`declaracao` : definicaoDeVariavel (atribuicao)? ;

Gramática sintática

atribuicao : ATRIBUIDOR expressaoOuLogico ;

expressao : expressaoOuLogico (atribuicao)? ;

expressaoOuLogico : expressaoELogico (OU
expressaoELogico)* ;

expressaoELogico : expressaoComparacaoLogica (E
expressaoComparacaoLogica)* ;

expressaoComparacaoLogica : expressaoAditiva
((IGUAL | DIFERENTE | MAIOR | MAIOR_IGUAL |
MENOR | MENOR_IGUAL) expressaoAditiva)* ;

Gramática sintática

expressaoAditiva : expressaoMultiplicativa ((SOMA | SUBTRACAO) expressaoMultiplicativa)* ;

expressaoMultiplicativa : expressaoUnaria ((MULTIPLICACAO | DIVISAO | RESTO_DA_DIVISAO) expressaoUnaria)* ;

expressaoUnaria : expressaoPrimaria | SUBTRACAO expressaoUnaria | NEGACAO expressaoUnaria ;

expressaoPrimaria

: PARENTESE_ESQUERDO expressao PARENTESE_DIREITO
| VALOR_BOOLEANO | VALOR_NULO
| CONSTANTE_INTEIRA | CONSTANTE_REAL
| LITERAL_CARACTERE | LITERAL_TEXTO
| (ESSE | ESSA | instanciacao | IDENTIFICADOR
(listaDeParametros)?) (selecao)* ;

Gramática sintática

selecao : (CHAMADA_DE_CLASSE | CHAMADA_DE_OBJETO)
IDENTIFICADOR (listaDeParametros)? ;

instanciacao : (NOVO | NOVA) tipo listaDeParametros ;

destruicao : DESTRUA ((ESSE | ESSA) CHAMADA_DE_OBJETO)?
IDENTIFICADOR ((CHAMADA_DE_CLASSE | CHAMADA_DE_OBJETO)
IDENTIFICADOR)* listaDeParametros ;

retorno : RETORNE expressao ;

se : SE PARARENTESE_ESQUERDO expressao PARENTESE_DIREITO
blocoDeInstrucoes (SENAO blocoDeInstrucoes)? ;

Gramática sintática

para : PARA PARENTESE_ESQUERDO expressao TERMINADOR expressao
TERMINADOR expressao PARENTESE_DIREITO blocoDeInstrucoes ;

repita : REPITA PARENTESE_ESQUERDO definicaoDeVariavel
PARENTESE_DIREITO blocoDeInstrucoes ;

enquanto : ENQUANTO PARARENTESE_ESQUERDO expressao
PARENTESE_DIREITO blocoDeInstrucoes ;

tente : TENTE blocoDeInstrucoes (CAPTURE PARENTESE_ESQUERDO
definicaoDeVariavel PARENTESE_DIREITO blocoDeInstrucoes)+ ;

lance : LANCE expressao ;

blocoDeInstrucoes : INICIO (instrucao)* FIM ;

Gramática sintática

instrucao

:

(options {k = 2;}

 : expressao TERMINADOR

 | declaracao TERMINADOR

)

| destruicao TERMINADOR

| retorno TERMINADOR

| se

| para

| repita

| enquanto

| tente

| lance TERMINADOR

;

Analizador sintático

O **Antlr** usará a gramática especificada para gerar os arquivos:

`fontes/java/.../antlr/CanecaSintatico.java`

`fontes/java/.../antlr/CanecaSintatico.tokens`

Analizador sintático

Foi criada uma gramática **LL(2)**. Para transformá-la em uma gramática **LL(1)** é necessário fatorar o token **IDENTIFICADOR** que aparece antes de **declarações** e **expressões**.

Nas **declarações** o **IDENTIFICADOR** é o tipo da variável declarada. Nas **expressões** o **IDENTIFICADOR** pode ser o nome de um método ou de uma variável.

Analizador sintático

Como não foi possível realizar a fatoração, uma opção para reduzir a gramática para **LL(1)** seria adicionar um prefixo às declarações, por exemplo:

```
declare Texto textoQualquer;
```

Entretanto, não parece ser vantajoso ter que utilizar uma palavra extra, sempre que for declarar uma variável, em troca de uma análise ligeiramente mais rápida com uma gramática **LL(1)**.

Portanto, nossa gramática gerada permaneceu **LL(2)**.

Compilador

Nessa segunda etapa o compilador apenas irá utilizar o analisador sintático gerado pelo **Antlr** para imprimir na tela se houve ou não erros sintáticos na análise de um dado arquivo.

`fontes/java/.../Compilador.java`

Uso do compilador:

```
java -classpath  
binarios/class:bibliotecas/jar/antlr.jar  
br.ufsc.inf.ine5426.caneca.Compilador.java  
<arquivo.caneca>
```


Exemplos

Análise sintática dos arquivos:

`fontes/caneca/ClasseDeExemplo.caneca`

`fontes/caneca/InterfaceDeExemplo.caneca`