Universidade Federal de Juiz de Fora
Departamento de Ciência da Computação
DCC045 - Teoria dos Compiladores Semestre ERE 2020-1

Analisador Sintático

Edson Lopes da Silva Junior 201635023

Vinicius Alberto Alves da Silva 201665558C

Professor: Leonardo Vieira dos Santos Reis

Relatório do trabalho prático Analisador Sintático, parte integrante da avaliação da disciplina.

Juiz de Fora

Outubro de 2020

1 Introdução

Este relatório é do trabalho prático 2 da disciplina de Teoria dos Compiladores do Ensino Remoto Emergencial (ERE) 2020.1 e tem como objetivo descrever o processo da criação de um Analisador Sintático (AS) para a Linguagem *Lang*.

2 Metodologia Utilizada

Para o desenvolvimento do Analisador Sintático requerido na especificação do trabalho, a linguagem Java foi utilizada. Optou-se pelo uso ta estratégia top-down left-left (LL) com a utilização da ferramenta *Another Tool for Language Recognition (ANTLR)*. Além disso a responsabilidade de reconhecedor léxico também foi transferida para o ANTLR para facilitar o uso da ferramenta.

2.1 Organização do Código

Para a organização do código, todos os fontes pertencem ao *package* nomeado parser. Os arquivos LangLexer.tokens, LangLexer.interp, Lang.interp, LangParser.java, LangBaseListener.java, LangLexer.java, LangListener.java são gerados pela ferramenta ANTLR e estão responsáveis pela análise léxica e sintática. No *package* ast encontra-se a classe Node responsáveis por gerar a arvore abstrata de dados (AST) que implementa a interface SuperNode. Por ultimo, a classe LangAdaptor implementa a interface ParseAdaptor. A classe LangAdaptor é responsável por chamar o método ParseFile, este faz a leitura do arquivo a partir da classe LangLexer e é feito o parser com a classe LangParser.

2.1.1 Como executar

Para executar basta executar o arquivo bash run.sh, ele contem o seguinte conteúdo:

```
#!/bin/bash

if [ -z "$1"]; then antlr_jar="antlr-4.8-complete.jar"; else antlr_jar=$1; fi

java -jar ./$antlr_jar -o ./parser/ Lang.g4
javac -cp .:$antlr_jar ast/*.java parser/*.java LangCompiler.java -d .
java -classpath .:$antlr_jar lang.LangCompiler -bs
```

É possível trocar o endereço do jar do antlr 4.8 por outro de sua preferência, basta fornecer o caminho do .jar na chamada de execução do script, exemplo:
bash run.sh pasta1/pasta2/antlr-4.8-complete.jar

2.1.2 Detalhes de implementação

Na implementação do método *parseFile* são feitas duas checagem de erro. A primeira é a partir de erros encontrados no textitLexer, foi adicionado um *ErrorListener* assim é possível reportar erros léxicos. Em sequência, erros do *parser* são reportados a partir do método *getNumberOfSyntaxErrors()*. Tanto erros léxicos quantos sintáticos resultam no método *parseFile* retornar *null*. Conforme especificação do trabalho.

2.2 Análise Sintática

O Antlr é uma ferramenta de geração de reconhecedores LL(*) essa técnica permite que algumas recursividades recursividades à esquerda não sejam tratadas. Na implementação deste trabalho não foi retirada a recursividade a esquerda da linguagem.

O arquivo Lang.g4 é utilizado na geração do analisador sintático pelo ANTLR, contendo a gramática que descreve a linguagem e as regras léxicas para identificar os tokens. No geral, a escrita da gramática definida no trabalho foi transposta para o arquivo Lang.g4 sem modificações.

As regras léxicas foram implementadas conforme descritas relatório do Analisador Léxico. Entretanto, duas mudanças foram feitas. A primeira é com relação ao identificador "nome de tipo", este padrão estava sendo reconhecido no nosso Analisador Léxico. Entretanto, para facilitar a implementação do parser de acordo com a gramática, esse casamento léxico foi retirado. Deixando a desambiguação do ID para o "nome de tipo"para as próximas etapas.

Além disso, foi feita uma mudança no LITERAL_CHAR pois no antlr é possível escrever [ntbr] ao invés de ter que separar cada caractere como no exemplo (n|t|b|r), seguindo a mesma lógica descrita no relatório anterior:

'\" ('\\' [btnr"'\\] | ~[\r\n\\"]) '\"

3 Conclusão

Este trabalho apresentou o processo de desenvolvimento de um analisador sintático para a linguagem lang. A ferramenta ANTLR foi selecionada para auxiliar o desenvolvimento.

Foram realizados testes de sintaxe com os arquivos disponibilizados, o AS aceitou 36/36 dos casos certos e rejeitou 35/36 dos casos errados sendo attrFALSE.cmd único arquivo aceito. Com isso, pode-se observar a corretude da implementação apresentada. O desenvolvimento deste projeto permitiu ampliar os conhecimentos e enfrentar de perto as dificuldades no desenvolvimento de um analisador sintático, enquanto as interfaces e o sistema de testes fornecidos foram grandes facilitadores durante o desenvolvimento.