T2 – Análise Léxica de Compiladores

Descrição: O trabalho 2 (T2) da disciplina consiste em implementar um analisador sintático para a linguagem LA (Linguagem Algorítmica) desenvolvida pelo prof. Jander, no âmbito do DC/UFSCar. O analisador sintático deve ler um programa-fonte e apontar onde existe erro sintático, indicando a linha e o lexema que causou a detecção do erro.

Professor: **Daniel Lucrédio**

Alunos:

- Daniel Watanabe 800697
- Paula Larocca 769705
- Victor Bonometo 800232

Requerimentos:

- Sistema operacional utilizado Linux (Ubuntu)
- Ferramenta ANTLR
- Manifesto para o arquivo Jar
- Java JDK

Execução:

1. Instalando ANTLR

   Vá no site: https://www.antlr.org/download.html

   E encontre a versão ANTLR-JAVA, podendo baixar tanto manualmente no site ou então no terminal com o comando wget:
   `wget https://github.com/antlr/antlr4/blob/master/doc/java-target.md`

   Agora é necessário mostrar o path do ANTLR que o sistema entenda:

   ```
   vim ~/.bashrc
   ```

   Vá para a última linha e adicione os seguintes caminhos:

   ```
   export CLASSPATH=".:/local/baixado/antlr-4.10.1-complete.jar:$CLASSPATH"
   alias antlr4='java -jar /local/baixado/antlr-4.10.1-complete.jar'
   alias grun='java org.antlr.v4.gui.TestRig'
   ```

   Agora o comando antlr4 deve funcionar.

2. Instalando Java JDK

   Basta rodar o comando:

   ```
   sudo apt install openjdk-11-jdk
   ```

   Da mesma forma, é necessário mostrar o path:

   ```
   vim ~/.bashrc
   ```

   Vá para a última linha e adicione os seguintes caminhos:

```
export JAVA_HOME=/usr/lib/jvm/jdk-11
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

Agora os comandos ***jar***, ***java*** e ***javac*** devem funcionar.

3. Compilar a gramática fornecida

Dessa forma basta gerar os arquivos fonte do ANTLR a partir da gramática:

```
antlr4 /path/onde/foi/baixado/Grama.g4
```

Ele criará diversos arquivos dentro da mesma pasta do arquivo Grama.g4, e então basta rodar o seguinte comando para compilar todos os arquivos Java gerados:

```
javac Grama*.java
```

Será gerado o arquivo GramaLexer que no arquivo T1.java poderá ser chamado como gramática de referência.

4. Compilar o arquivo java

Compila todos os arquivos java sendo tanto da gramática quanto o t1 colocando o resultado na basta bin

```
javac -classpath .:/pasta/do/antlr /t2/antlr-4.13.1-complete.jar -d bin /pasta /do/t2/*.java
```

Agora é necessário criar um arquivo manifesto:

```
vim manifest.txt
```

Coloque esses dados dentro:

```
manifest-Version: 1.0
```

```
Class-Path: . /pasta/da /ANTLR /t2/antlr-4.13.1-complete.jar
```

```
Main-Class: T2
```

E por fim criar um arquivo jar executável:

```
jar cvfm t2.jar manifest.txt -C bin/ .
```

Criando assim um arquivo chamado ***t1.jar*** executável, para executá-lo basta rodar o seguinte comando:

```
java -jar t2.jar /local/arquivo/entrada/entrada.txt /local/arquivo/saida/saida.txt
```

5. Rodar os casos de testes

Para conseguir rodar o corretor automático deve-se baixar os arquivos necessários dos casos de teste, criar uma pasta temporária para as próprias saídas geradas

```
java -jar compiladores-corretor-automatico-1.0-SNAPSHOT-jar-with-dependencies.jar "java -jar /local/do/executavel/t1.jar" gcc /pasta/temporaria/temp /local/pasta/testes/casostestes "ra3, ra2, ra1" t2
```

Gerando a seguinte saída: