

Universidade Federal de Juiz de Fora
Departamento de Ciência da Computação
Relatório Trabalho Prático
Teoria dos Compiladores

Relatório Trabalho Prático

André Luiz dos Reis - 201965004AC
Lucca Oliveira Schröder - 201765205C

Juiz de Fora - MG
2022

Sumário

Analizador léxico	3
Expressões regulares	3
Autômato para os tokens da linguagem	3
Estrutura de Dados usadas	4
Estratégia de implementação do analisador léxico e de “bufferização” da entrada	4
Compilação e Execução	4

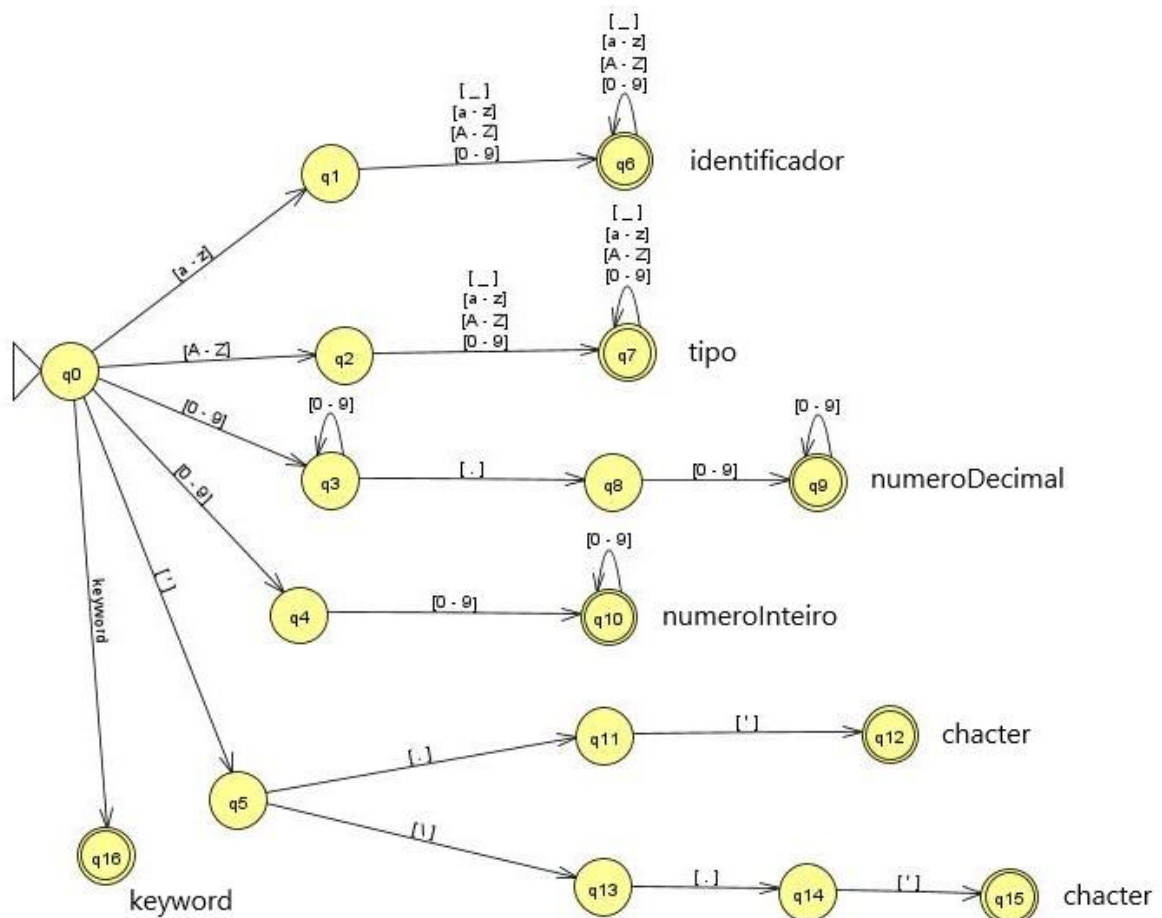
1. Analisador léxico

1.1. Expressões regulares

Token	Expressão regular
identificador	[a - z] [a- Z-0-9- _]*
tipo	[A - Z] ([a - z] [A - Z] [0 - 9] [_])*
numeroDecimal	[0 - 9]* [.] [0 - 9]+
numeroInteiro	[0 - 9]+
chacter	['] + [.] + ['] ['] + [\] + [.] + [']

P.S.: [.] faz referência a qualquer tipo de caractere que possa ser usado

1.2. Autômato para os tokens da linguagem



A keyword faz referência a todas palavras reservadas (true, false, null, bool, etc), a todos os símbolos (“==”, “<”, “]”, etc) e também a todas operações matemáticas (“+”, “-”, etc).

1.3. Estrutura de Dados usadas

Utilizamos a classe “public enum TOKEN_TYPE” para poder fazer a especificação de tokens existentes na linguagem por meio de uma enumeração.

Além disso, temos a classe “Token” que guarda as informações do token. Ela contém linha e coluna, o lexema e em alguns casos temos a informação também.

E para finalizar, existe uma função utilizada para printar os tokens na tela.

1.4. Estratégia de implementação do analisador léxico e de “bufferização” da entrada

Nós utilizamos a ferramenta Jflex por padrão, dessa forma foi gerado o código.

<https://www.jflex.de/docu.html>

1.5. Compilação e Execução

Para a execução poderão ser utilizado os scripts **run.sh** e **run.cmd** que executam o arquivo .jar presente na pasta dist e roda o arquivo **exemplo1.txt** presente na pasta Teste.