Análise Léxica

Mateus Tomoo Yonemoto Peixoto

DACOM – Universidade Tecnológica Federal do Paraná (UTFPR)
Caixa Postal 271 – 87301-899 – Campo Mourão – PR – Brazil

{mateustomoo}@gmail.com

Abstract. *This paper describes the development of the lexical analysis for a compiler being designed for the T++ programming language. In the future, it will also approach the development of the syntatic and semantic analysis.*

Resumo. *Este artigo descreve o desenvolvimento da análise léxica para um compilador projetado para a linguagem T++. No futuro, também abordará o desenvolvimento da análise sintática e semântica.*

1. Introdução

A Análise Léxica é a fase do compilador que lê o programa-fonte como um arquivo de caracteres e o separa em um conjunto de marcas (tokens), onde cada marca representa uma unidade. Podemos reconhecer como marca identificadores e símbolos especiais a partir de expressões regulares e autômatos finitos.

Para o desenvolvimento da análise léxica, foi utilizado a linguagem de programação Python (versão 3.6.2), onde possui a biblioteca chamada PLY, que contém ferramentas léxicas e sintáticas.

2. A linguagem

A linguagem T++ possui algumas características, dentre elas são:

- Tipos básicos de dados suportado: inteiro e flutuante;

- Suporte a arranjos uni e bidimensionais (arrays);

- Variáveis globais e locais devem ter um dos tipos especificados;

- Linguagem quase fortemente tipificada: nem todos os erros são especificados, mas sempre deve ocorrer avisos.

3. Análise Léxica

As palavras reservadas da linguagem T++ que foram utilizadas são:

Palavras reservadas
se
então
senão
fim
repita
flutuante
retorna
até
leia
escreve
inteiro

Tabela 1. Palavras reservadas

Todas essas palavras reservadas são tratadas como ‘PR’ no programa.

Os símbolos da linguagem T++ que foram utilizados são:

Símbolos
+
-
*
/
=
,
:=
<
>
<=
>=
(
)
:
[
]
&&
!

Tabela 2. Símbolos

Todos esses símbolos tratados como ‘SB’ no programa.

Foram utilizadas algumas expressões regulares para se fazer as análises, que são elas:

- NEWLINE: \n+ - função que trata a quebra de linha;
- COMENTÁRIO: \{[^}]*[^{]*\} – função que trata os comentários;
- PR: se\b|então\b|senão\b|fim\b|repita\b|flutuante\b|retorna\b|até\b|leia\b|escreva\b|inteiro\b – função que trata as palavras reservadas da linguagem;
- ID: [a-zA-Zà-ú][a-zA-Zà-ú_0-9]* - função que trata qualquer ID que aparecer;
- CIENTÍFICA: ([+-]?(\d+)(.\d+)([eE][+\-]?(\d+))) – função que trata números em notação científica;
- FLUTUANTE: ((\+|\-)?\d+)(\.(\d)+) – função que trata números em ponto flutuante;
- INTEIRO: ((\+|\-)?\d+) – função que trata números inteiros;
- SB: \+|\-|*|/|\=|,|(\?<\!<|>|:|=)=(?!=)|\<|\>|\<=\b|\>=\b|\(|\)|\:|\[|\]|&&|\! – função que trata todos os símbolos;
- ESPAÇO EM BRANCO: t_ignore = ' \t' – expressão que trata os espaços em brancos.

3.1. Autômatos

-NÚMERO INTEIRO - NÚMERO FLUTUANTE - ID

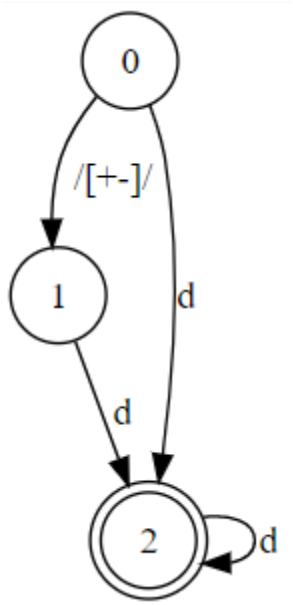

Figura 1. Autômato Número Inteiro

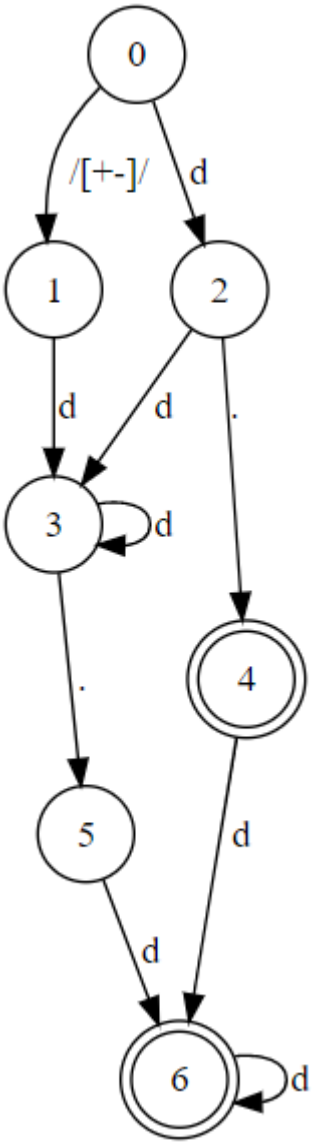

Figura 2. Autômato Número Flutuante

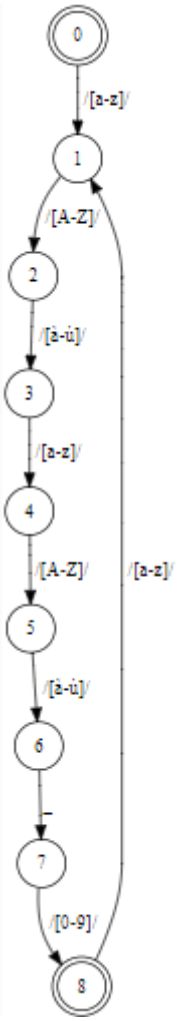

Figura 3. Autômato ID

- NOTAÇÃO CIENTÍFICA

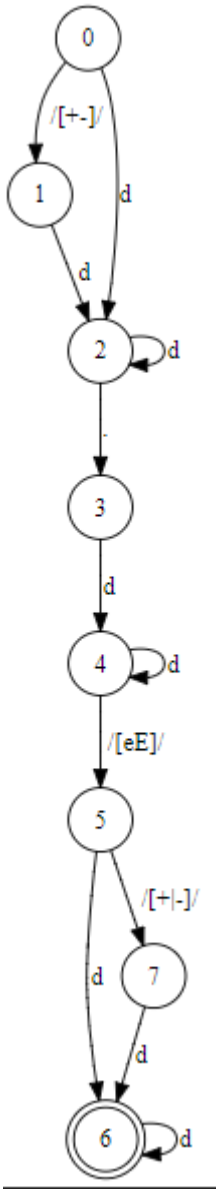

Figura 4. Autômato
Notação Científica

- COMENTÁRIO

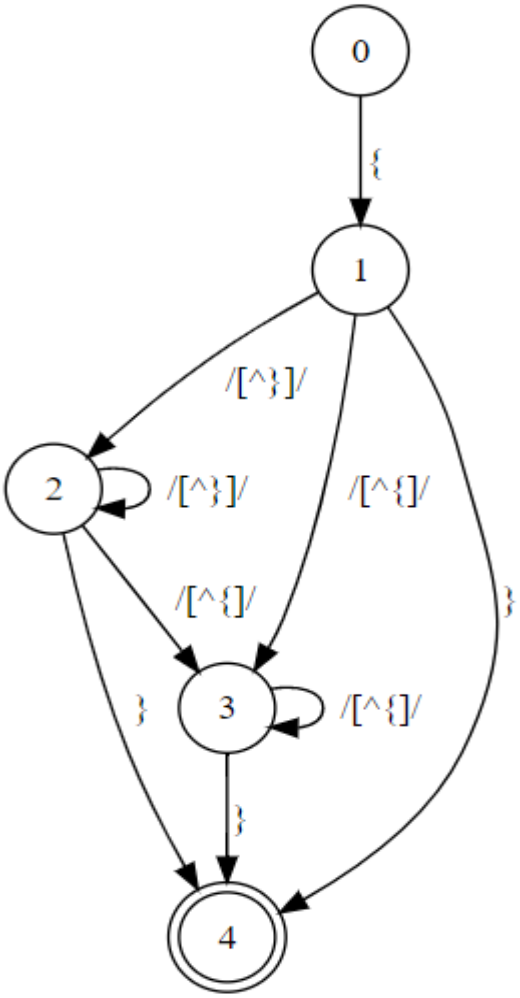

Figura 5. Autômato Comentário

- PALAVRAS RESERVADAS

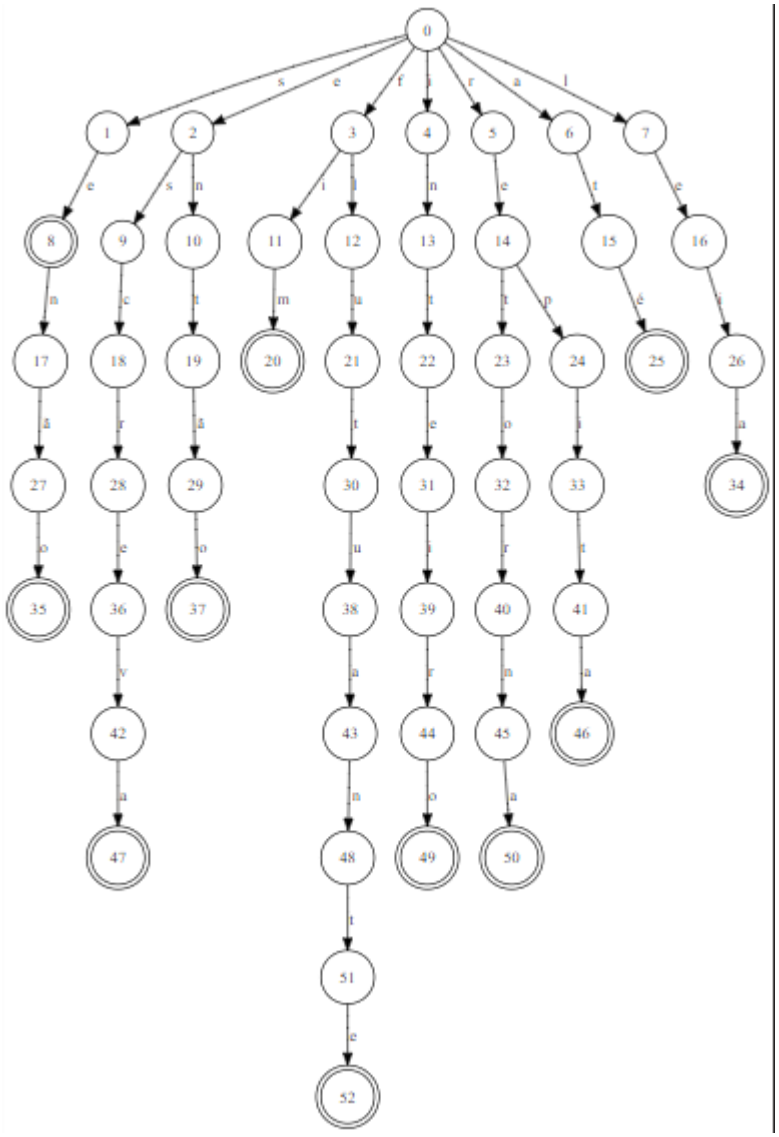

Figura 6. Autômato Palavras Reservadas

3.2. Exemplo de saída

Dado essa entrada:

```
1   inteiro: n, g
2   flutuante: d,a
3   inteiro fatorial(inteiro: n)
4       d := 5.6
5       inteiro: fat
6       fat := 2e+89 + 67*12
7       se a > 10 então
```

Será gerado a seguinte saída:

```
<PR, inteiro>
<SB, :>
<ID, n>
<SB, ,>
<ID, g>
<PR, flutuante>
<SB, :>
<ID, d>
<SB, ,>
<ID, a>
<PR, inteiro>
<ID, fatorial>
<SB, (>
<PR, inteiro>
<SB, :>
<ID, n>
<SB, )>
<ID, d>
<SB, :=>
<FLUTUANTE, 5.6>
<PR, inteiro>
<SB, :>
<ID, fat>
<ID, fat>
<SB, :=>
<INTEIRO, 2>
<ID, e>
<INTEIRO, +89>
<SB, +>
<INTEIRO, 67>
<SB, *>
<INTEIRO, 12>
<PR, se>
```

4. Referências

LOUDEN, Kenneth C. Compiladores: princípios e práticas. São Paulo, SP: Thomson, c2004. xiv, 569 p. ISBN 8522104220.

http://hackingoff.com/compilers/regular-expression-to-nfa-dfa

PLY (Python Lex-Yacc). Disponível em: <http://www.dabeaz.com/ply/>.