

# Trabalho Prático de Linguagens Formais e Autômatos

## Implementação de um analisador léxico simplificado

João Victor Domingos e Souza; Lorenzo Jordani Bertozzi Luz; Marcela Gomes Pinheiro

18 de junho de 2025

## 1 Introdução

Um Analisador Léxico, é um componente presente no compilador cuja função é estar diretamente em contato com códigos fonte. Ou seja, um analisador léxico lê e identifica tokens presentes no código fonte, a fim de agrupá-los de forma que faça sentido para outros componentes presentes no compilador [1].

O objetivo do presente trabalho, é desenvolver parte de um **Analisador Léxico Simplificado**, a parte a ser desenvolvida é o **reconhecedor de tokens**. Além disso, a entrada do programa deverá ser um arquivo de texto e sua sintaxe limitada aos caracteres apresentados na Figura 1. Cabe ressaltar que, além dos tokens solicitados, também foi incluído o símbolo ‘//’ para permitir comentários de linha no código de entrada, caso necessário.

Token	Texto Lido	Token	Texto Lido
LParenteses	(	SUM	+
RParenteses	)	SUB	-
LChave	{	MULT	*
RChave	}	DIV	/
Lcolchete	[	RESTO	%
RColchete	]	INTDEF	int
EQ	==	FLOATDEF	float
Atribuicao	=	AND	&&
GEQ	>=	OR	
LEQ	<=	NUM_INT	(digitos) <sup>+</sup>
GT	>	NUM_FLOAT	(digitos)*.(digitos) <sup>+</sup>
LT	<	DIF	!=
NEG	!	CHAR_TYPE	char
Virgula	,	BOLL_TYPE	bool
PVirgula	;	RETURN	return
IF	If	VAR	VAR(letras,digitos) <sup>+</sup>

Figura 1: Tabela de tokens da linguagem C simplificada.

### 1.1 AFD: Autômato Finito Determinístico

Um Autômato Finito Determinístico (AFD) - também conhecido como Máquina de Estados Finitos, pode ser definido como uma máquina que possui como principal função aceitar ou rejeitar certas **cadeias de códigos finitas** a fim de gerar apenas um ramo computacional de resposta para cada entrada possível [4].

Dessa maneira, um AFD pode ser definido formalmente como uma quintupla  $M = (S, Q, d, q_0, F)$ , onde [4]:

- S: alfabeto de códigos/símbolos finitos de entrada;

- A Figura 2 apresenta um exemplo de um AFD.



Durante o desenvolvimento do AFD, foi preciso analisar os textos lidos compostos e realizar o tratamento adequado para retorno do *token* correspondente, ou seja, verificar se o segundo caractere lido era um espaço vazio ou não. Como resultado, obtêm-se o AFD apresentado na Figura 3:



2

## 1.2 Implementação do Analisador Léxico em Python

O Python é uma linguagem de programação amplamente utilizada nas mais diversas áreas da computação, dessa forma, os desenvolvedores optam pela utilização do Python devido a sua eficiência e facilidade de aprendizado. Além de ser uma linguagem executável em diversas plataformas e possuir software gratuito e de fácil integração com os sistemas operacionais [2].

Todo trabalho se encontra em um repositório do *GitHub* e pode ser acessado através do link: [https://github.com/John5626/LFA\\_TPI\\_2025\\_1](https://github.com/John5626/LFA_TPI_2025_1).

## 1.3 Arquivo de entrada - execução do programa

```
1  int VARa = VARb - 4 ;
2  int x = 123 ;
3  float y = 0.5 ;
4  char z = 65 ;
5  bool flag = 1 ;
6
7  if ( x == y ) { return x ; }
8
9  a >= b <= c < d > e != f
10
11  a + b - c * d / e % f && g || h
12
13  [ a , b ]
14
15  { a ; b ; }
```

Listing 1: Entrada

## 1.4 Arquivo de saída do programa

```
1  INTDEF VAR Atribuicao VAR SUB NUM_INT PVirgula
2  INTDEF VAR Atribuicao NUM_INT PVirgula
3  FLOATDEF VAR Atribuicao NUM_FLOAT PVirgula
4  CHAR_TYPE VAR Atribuicao NUM_INT PVirgula
5  BOLL_TYPE VAR Atribuicao NUM_INT PVirgula
6
7  IF LParenteses VAR EQ VAR RParenteses LChave RETURN VAR PVirgula RChave
8
9  VAR GEQ VAR LEQ VAR LT VAR GT VAR DIF VAR
10
11  VAR SUM VAR SUB VAR MULT VAR DIV VAR RESTO VAR AND VAR OR VAR
12
13  LColchete VAR Virgula VAR RColchete
14
15  LChave VAR PVirgula VAR PVirgula RChave
```

Listing 2: Saída

## Referências

- [1] Capítulo 2 - analisador léxico. <https://www.dcce.ibilce.unesp.br/~aleardo/cursos/compila/cap02.pdf>. Accessed: 12-06-2025.
- [2] O que é python? <https://aws.amazon.com/pt/what-is/python/>. Accessed: 12-06-2025.
- [3] J. F. Michels, L. Cassanho, B. Burigo, and C. R. Barbosa. Avaliação do jflap como ferramenta de ensino de gramáticas na disciplina de linguagens formais e autômatos. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação*, pages 199–209, Porto Alegre, RS, Brasil, 2024. SBC.
- [4] J. Pessoa. Autômatos finitos não determinísticos (AFN) e determinísticos (AFD). <http://www.dsc.ufcg.edu.br/~pet/jornal/junho2014/materias/recapitulando.html>. Accessed: 12-06-2025.