

Trabalho de Compiladores – Análise Léxica

Yuri Cristian Pedrosa de Oliveira¹

¹Universidade Federal do Ceará – Campus Crateús – Crateús – CE – Brasil

yuripedrosa@alu.ufc.br

Abstract. *This paper describes the lexical analyzer of a C language compiler, showing how language tokens are constructed using regular expressions and automata.*

Resumo. *Este artigo descreve o analisador léxico de um compilador da linguagem C, mostrando como são construídos os tokens da linguagem por meio de expressões regulares e de autômatos.*

1. Expressões Regulares

1.1. Classes Base

- letra: [a-zA-Z]
- dígito: [0-9]
- palavra: (letra | dígito)*

1.2. Identificadores

- id: {letra} ({letra} | {dígito})*
- variável: tipo id pont
- constante: #define id palavra | literal

1.3. Operadores

- oplog: && | || | !
- oprel: < | > | <= | >= | != | ==
- oparit: + | - | * | / | %
- opatri: =

1.4. Delimitadores

- delimitador: { | } | (|) | ; | . | : | # | @ | ,

1.5. Tipos de Dados

- int: dígito+
- float: dígito+.dígito+
- char: '(letra|dígito)'

- void:
- string: “(letra|digito)*”

1.6. Reservadas

- reservado: if | int | void | float | char | string | else | main | return | for | while

2. Autômatos

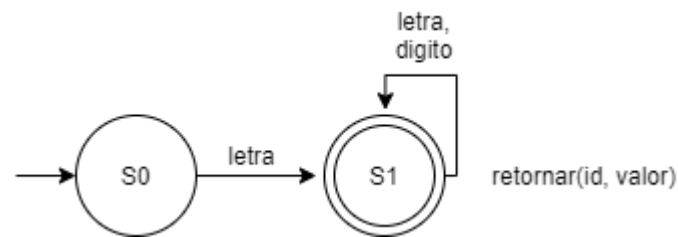


Figura 1. Autômato para identificadores.

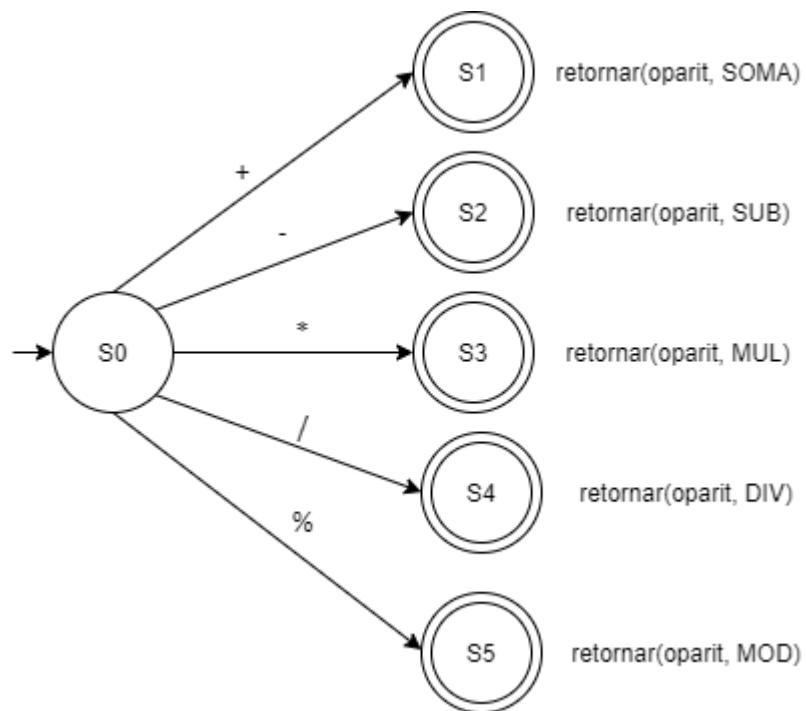


Figura 2. Autômato para operadores aritméticos.

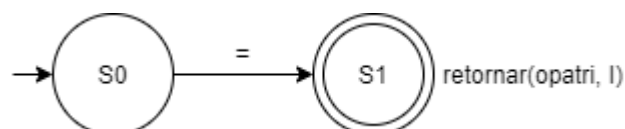


Figura 3. Autômato para operadores de atribuição.

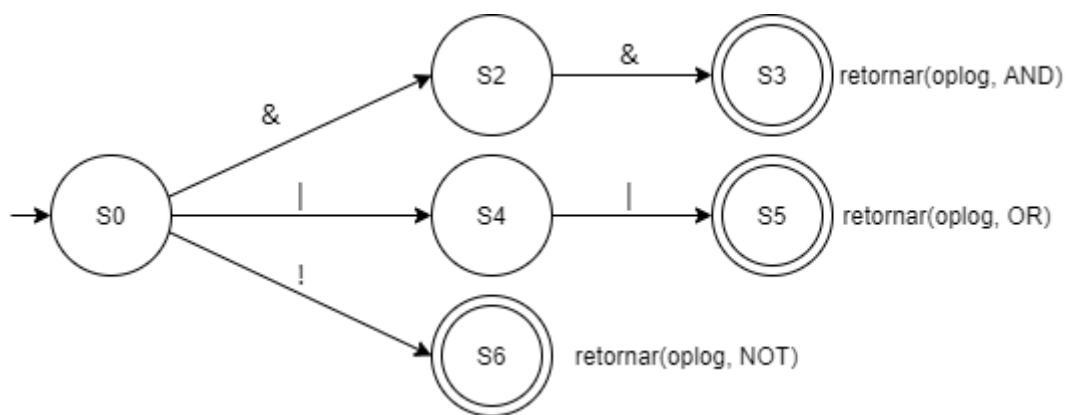


Figura 4. Autômato para operadores lógicos.

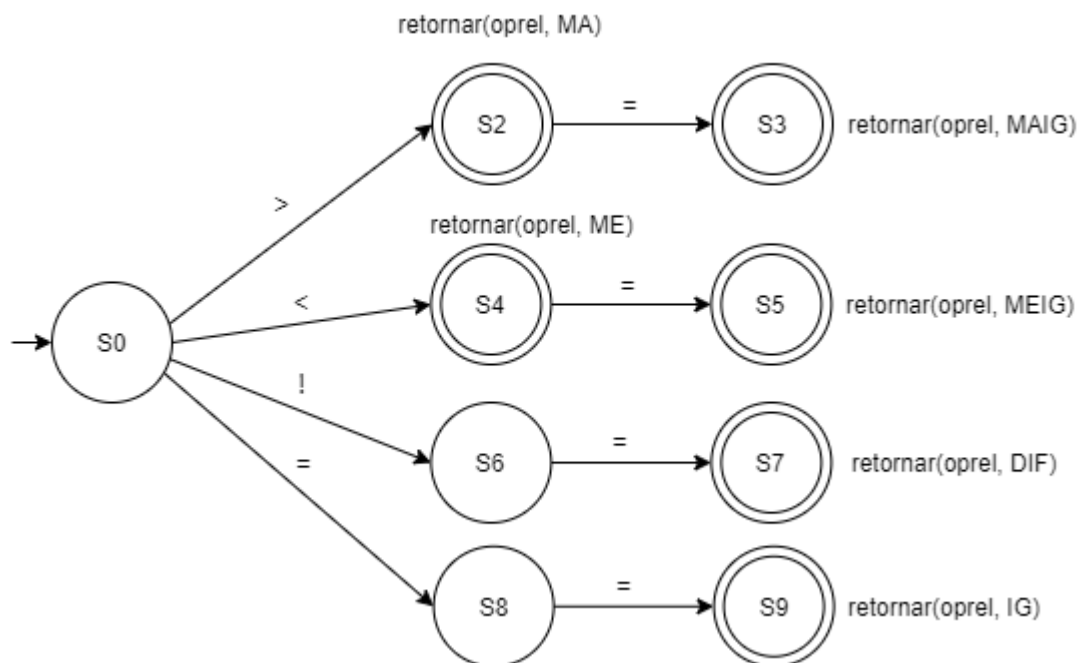


Figura 5. Autômato para operadores relacionais.

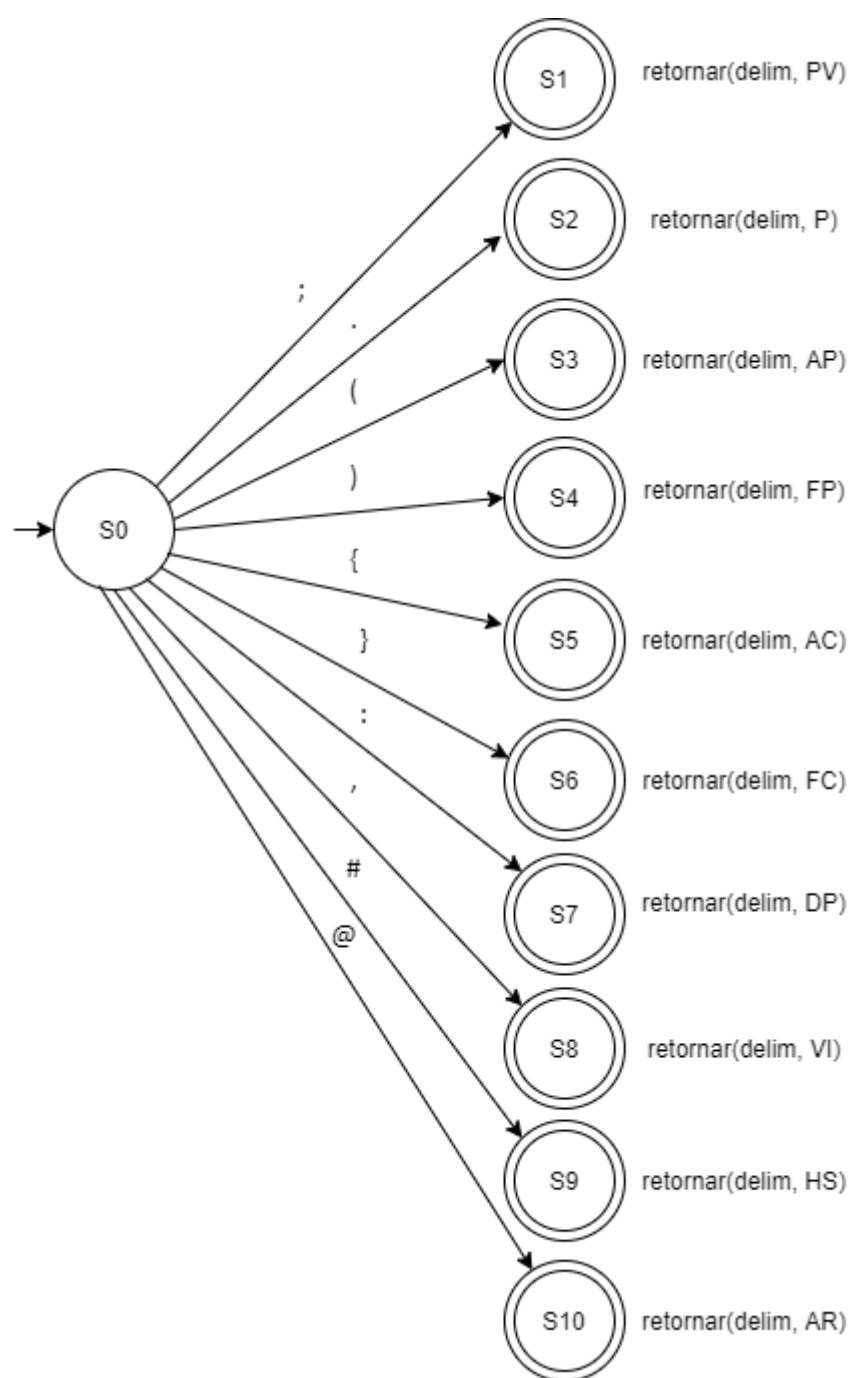


Figura 6. Autômato para delimitadores.

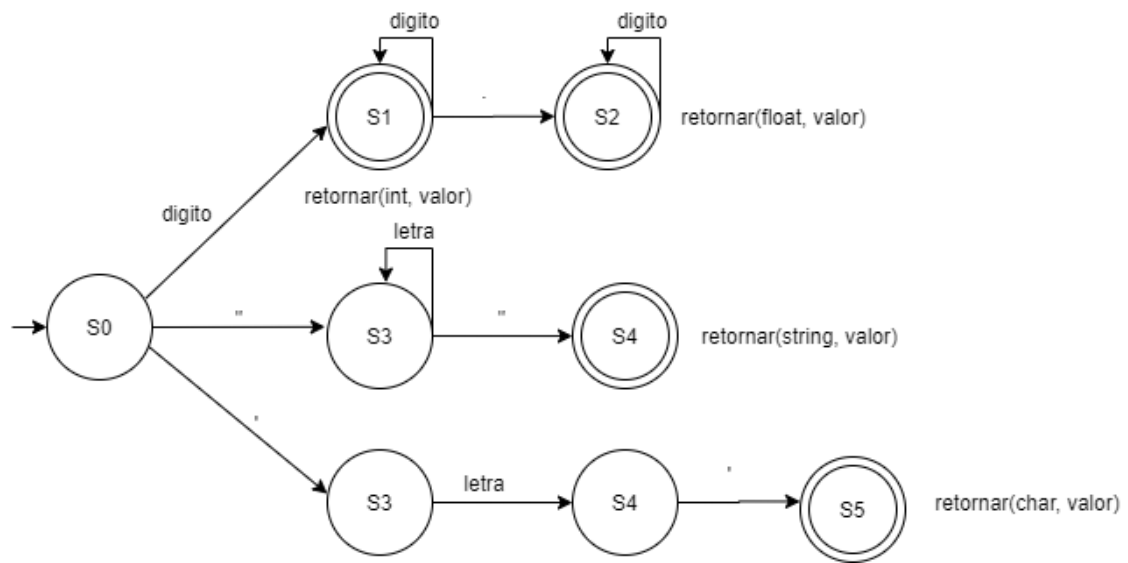


Figura 7. Autômato para tipos de dados.