

Trabalho de Compiladores - Linguagem L

Gabriel G. Luciano¹, Geovane F S Santos¹, Luigi D C Soares¹

¹Pontifícia Universidade Católica de Minas Gerais (PUC-MG)
30535-901 – Belo Horizonte – MG – Brasil

{geovane.fonseca,luigi.soares,glgomes}@sga.pucminas.br

1. Descrição Geral da Linguagem L

A linguagem “L” é uma linguagem imperativa simplificada, com características herdadas das linguagens C e Pascal. Ela oferece tratamento para 2 tipos básicos explícitos: char e integer, além do tipo lógico que é implícito. O tipo char pode ser escrito em formato alfanumérico ou hexadecimal. No primeiro caso, deve seguir o padrão ‘*c*’, onde *c* é um caractere imprimível. No segundo, deve ser escrito da forma 0x*HH*, onde *H* é um número hexadecimal.

Além dos tipos básicos, a linguagem permite a definição de vetores unidimensionais de caracteres e inteiros. Um string é um vetor de caracteres que, quando armazenado em memória, tem seu conteúdo útil finalizado pelo caracter ‘\$’. Constantes que representam strings são delimitadas por aspas e não devem conter aspas, quebra de linha ou \$. Entretanto, esses elementos são permitidos nos vetores de caracteres. Dessa forma, vetores e strings são compatíveis entre si, ficando a cargo do programador o controle dos seus conteúdos e tamanhos. Tipo lógico assume valores 0 (falso) ou 1 (verdadeiro). Comentários são delimitados por /* */. A quebra de linha e o espaço podem ser usados livremente como delimitadores de lexemas. Caracteres maiúsculos e minúsculos não são diferenciados. A estrutura básica de um programa-fonte é da forma:

Declarações Bloco_de_Comandos fim_arquivo

2. Alfabeto e Padrão de Formação dos Lexemas

Para a definição dos padrões de formação, considere os elementos (em itálico) como correspondentes aos meta-símbolos definidos a seguir:

- *l* (letra) := [a-zA-F];
- *d* (dígito) := [0-9];
- *s* (caracteres aceitos em uma string) := ≠ {", \$, quebra de linha};
- *α* (caracteres alfanuméricos) := qualquer caractere válido no arquivo fonte;
- *h* (hexadecimal) := [0-9a-fA-F].

Os demais caracteres (aqueles que não estão em itálico) correspondem ao caractere propriamente dito.

Σ	Lexema
Id	$(l \cup (_ \cup \cdot)^+(l \cup d))(l \cup d \cup _ \cup \cdot)^*$
Const	$d^* \cup "s^*" \cup '\alpha' \cup 0(x \cup X)hh$
ConstKW	$(C \cup c)(O \cup o)(N \cup n)(S \cup s)(T \cup t)$
Var	$(V \cup v)(A \cup _)(R \cup r)$
Integer	$(I \cup i)(N \cup n)(T \cup t)(E \cup e)(G \cup g)(E \cup e)(R \cup r)$
Char	$(C \cup c)(H \cup h)(A \cup a)(R \cup r)$
For	$(F \cup f)(O \cup o)(R \cup r)$
If	$(I \cup i)(F \cup f)$
Else	$(E \cup e)(L \cup l)(S \cup s)(E \cup e)$
And	$(A \cup a)(N \cup n)(D \cup d)$
Or	$(O \cup o)(R \cup r)$
Not	$(N \cup n)(O \cup o)(T \cup t)$
EQ	$=$
To	$(T \cup t)(O \cup o)$
LParen	$($
RParen	$)$
LT	$<$
GT	$>$
NE	$<>$
GE	$>=$
LE	$<=$
Comma	$,$
Add	$+$
Sub	$-$
Mult	$*$
Div	$/$
Semicolon	$;$
LBrace	$\}$
RBrace	$\{$
Then	$(T \cup t)(H \cup h)(E \cup e)(N \cup n)$
Readln	$(R \cup r)(E \cup e)(A \cup a)(D \cup d)(L \cup l)(N \cup n)$
Step	$(S \cup s)(T \cup t)(E \cup e)(P \cup p)$
Do	$(D \cup d)(O \cup o)$
Write	$(W \cup w)(R \cup r)(I \cup i)(T \cup t)(E \cup e)$
Writeln	$(W \cup w)(R \cup r)(I \cup i)(T \cup t)(E \cup e)(L \cup l)(N \cup n)$
Mod	$\%$
LBracket	$[$
RBracket	$]$

Table 1. Alfabeto e Padrão de Formação dos Lexemas

3. Autômato Finito Determinístico

Nos automâtos a seguir, alguns caracteres foram delimitados por apóstrofos, a fim de evitar confusões relacionadas ao uso destes em outros contextos (e.g. vírgula, usada para separar os terminais lidos em cada transição). Estes apóstrofos não devem

ser confundidas com o caractere propriamente dito, que é utilizado para delimitar uma constante do tipo *char*. Por convenção, *outro* representa qualquer caractere que não apareça em outras transições do estado que a origina. Em transições para o estado final, a presença de colchetes (e.g. [outro]) refere-se a ação de devolver o caractere ao arquivo de entrada. Por fim, definem-se os seguintes meta-símbolos a serem utilizados em certas transições:

- $\alpha :=$ caracteres alfanuméricos (qualquer caractere válido);
- *branco* := espaço em branco (incluindo tabulações e quebras de linhas);
- *letra* := [a-zA-F];
- *dígito* := [0-9];
- *nãoNulo* := [1-9];
- *hexa* := *dígito* | *letra*.

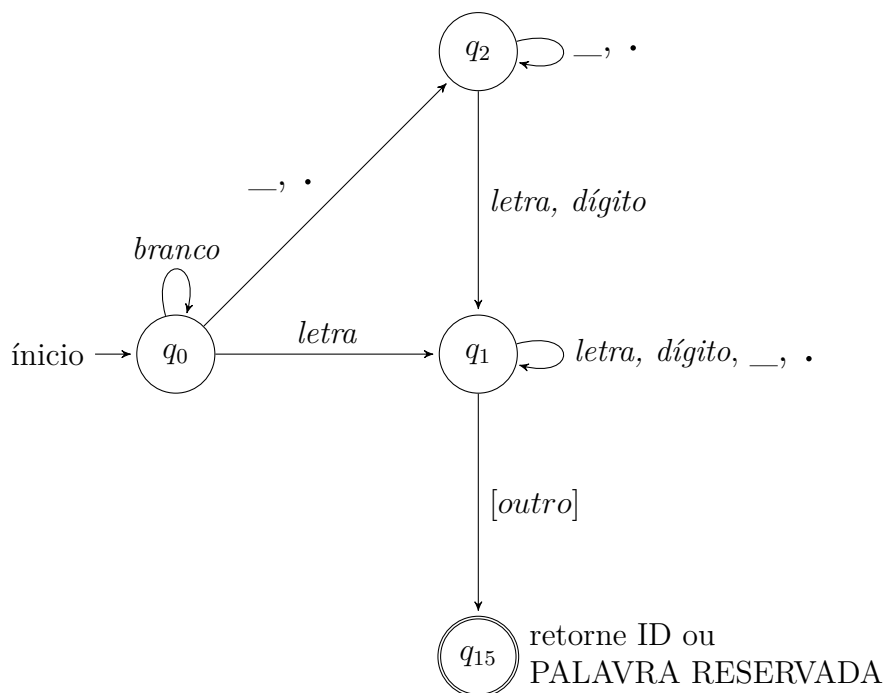


Figure 1. Autômato: identificadores, palavras reservadas e espaço em branco.

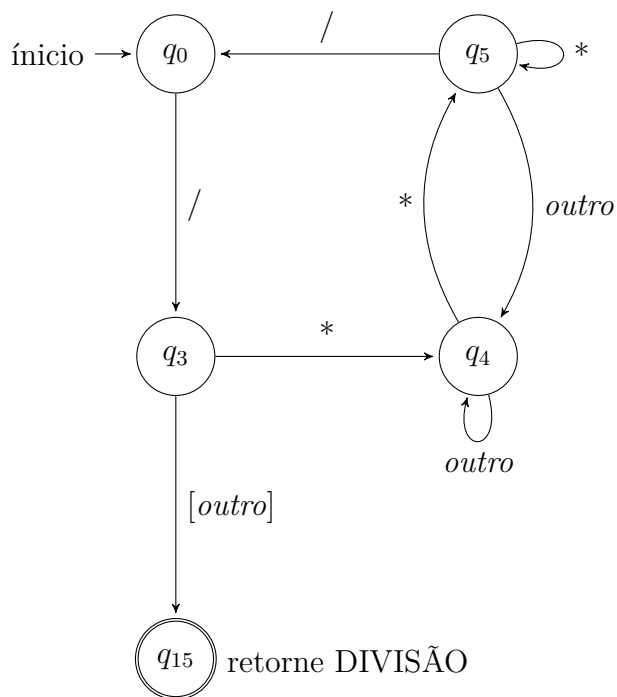


Figure 2. Autômato: Operador de divisão e comentário.

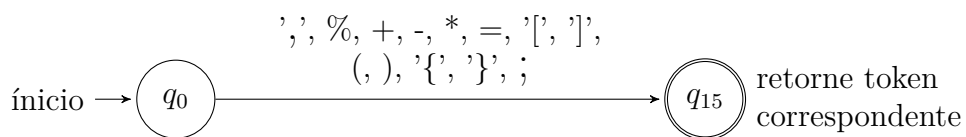


Figure 3. Autômato: Operadores aritméticos, comparadores, delimitadores, etc.

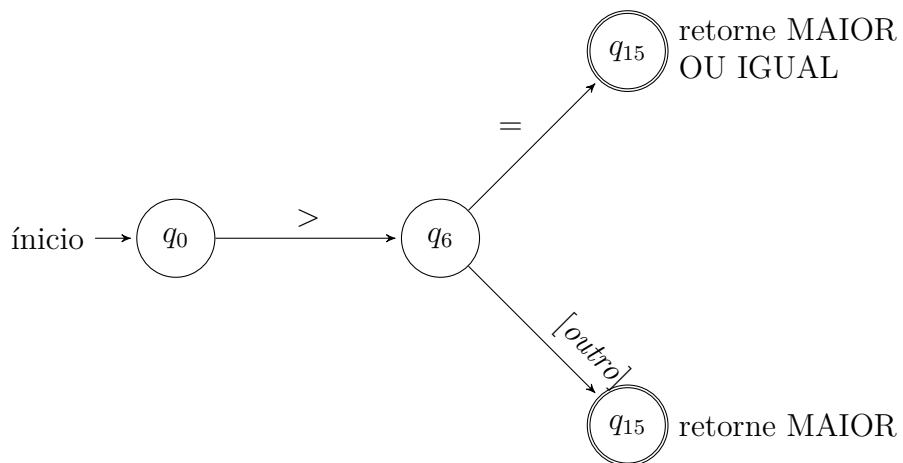


Figure 4. Autômato: $>$, $>=$.

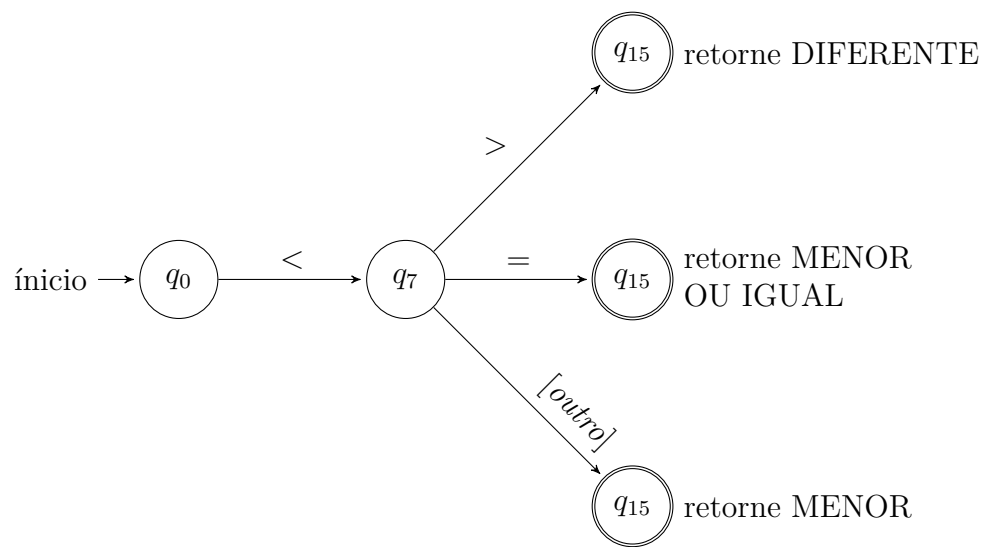


Figure 5. Autômato: $<$, $<=$, $<.>$.

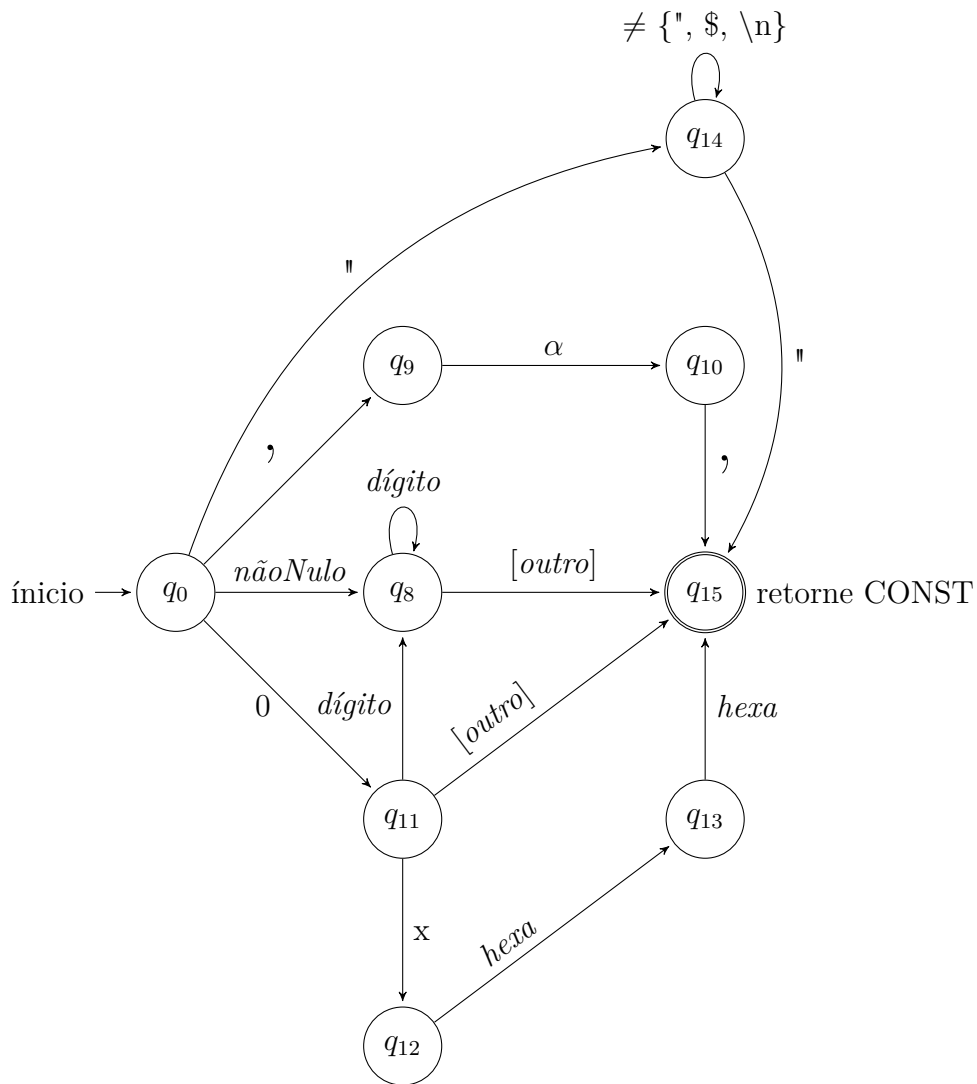


Figure 6. Autômato: Constantes.

4. Gramática

A seguir, define-se a gramática LL(1) da linguagem:

$$\begin{aligned}
S &\longrightarrow \{D\} \{C\} \\
D &\longrightarrow \text{var } \{(\text{integer} \mid \text{char}) \textit{DV} ;\}^+ \mid \text{constkw} = [+ \mid -] \text{const} ; \\
DV &\longrightarrow \text{id } DVO \{, \text{id } DVO\} \\
DVO &\longrightarrow [= [+ \mid -] \text{const} \mid '[' \text{const} ']'] \\
C &\longrightarrow \\
&\quad \text{id } '[' \textit{Exp} ']' = \textit{Exp} ; \\
&\quad \mid \text{if } \textit{Exp} \text{ then } LC \text{ [else } LC] \\
&\quad \mid \text{for } i = \textit{Exp} \text{ to } \textit{Exp} \text{ [step const] do } LC \\
&\quad \mid ; \\
&\quad \mid \text{readln } '(' \text{ id } ')' ; \\
&\quad \mid (\text{write} \mid \text{writeln}) '(' \textit{Exp} \{, \textit{Exp} \} ')' ; \\
LC &\longrightarrow '(' \{C\} ')' \mid C \\
\textit{Exp} &\longrightarrow \textit{ExpS} [(= \mid <> \mid < \mid > \mid <= \mid >=) \textit{ExpS}] \\
\textit{ExpS} &\longrightarrow [+ \mid -] T \{ (+ \mid - \mid \text{or}) T \} \\
T &\longrightarrow F \{ (* \mid / \mid \text{and} \mid \%) F \} \\
F &\longrightarrow \text{not } F \mid '(' \textit{Exp} ')' \mid \text{const} \mid \text{id } '[' \textit{Exp} ']'
\end{aligned}$$

Onde os símbolos representados por caracteres maiúsculos e em itálico correspondem aos não-terminais, podendo ser entendidos como:

- S := Símbolo inicial;
- D := Declarações;
- DV := Declaração de variável;
- DVO := Parte opcional da declaração de variável (atribuição ou vetor);
- C := Comandos;
- LC := Lista de comandos;
- Exp e ExpS := Expressão;
- T := Termo;
- F := Fator.