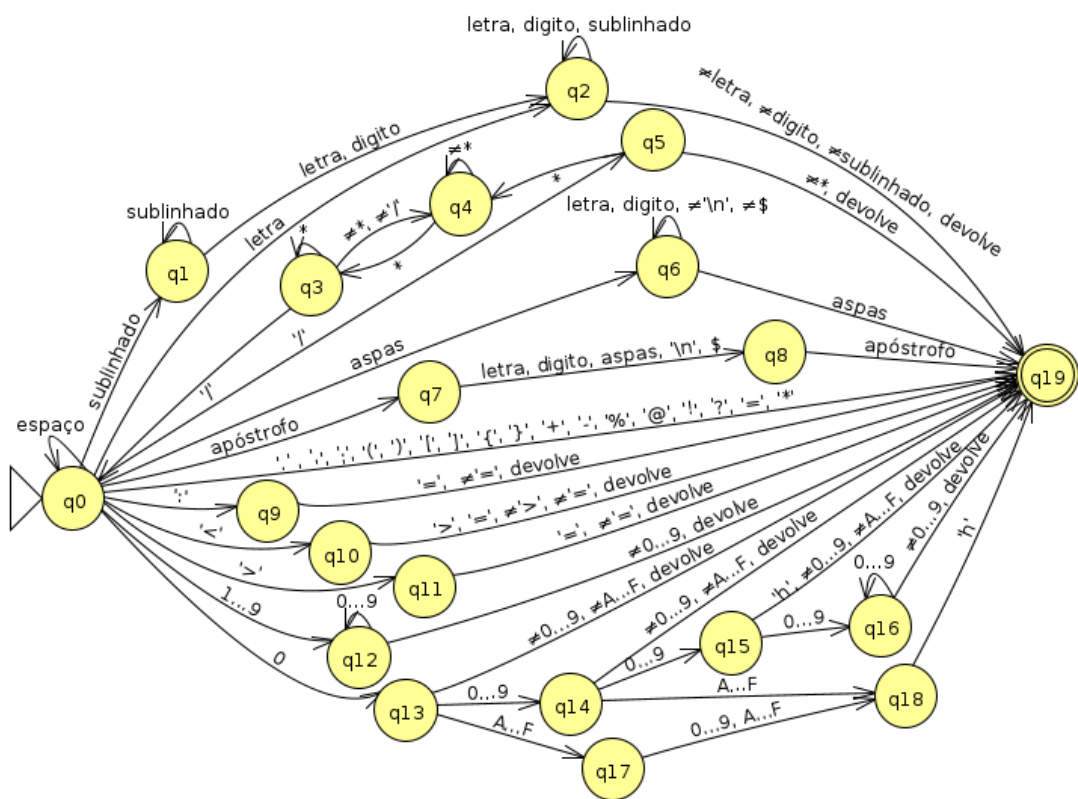


# Trabalho Prático

**Ana Flávia Dias Rodrigues - 561288**  
**Jorge Luiz Mendes Moreira Junior - 553273**  
**Stefany Gaspar Xavier França - 582931**

Instituto de Ciências Exatas e Informática  
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

## 1. Autômato



## 2. Definição do alfabeto

Token	Lexema
final	'final'
int	'int'
char	'char'
for	'for'
if	'if'
TRUE	'TRUE'
else	'else'
and	'and'
or	'or'

not	'not'
:=	':='
=	'='
(	'('
)	')'
<	'<'
>	'>'
<>	'<>'
<=	'<='
>=	'>='
,	','
+	'+'
-	'-'
*	'*'
/	'/'
;	';'
{	'{'
}	'}'
then	'then'
readln	'readln'
FALSE	'FALSE'
write	'write'
writeln	'writeln'
%	'%'
[	'['
]	']'
main	'main'

### 3. Gramática e Esquema de tradução

#### 3.1. Gramática

**S** -> {D} main "{ {C} }"

#### Declaração

**D** -> T (1) X ; | final (6) id = (7) V;

**X** -> { (2) id ([ = V (4) ] | "[" (5) constante "]" ) [, ] } +

**T** -> int | boolean | char

**V** -> [ + | - ] (3) constante

#### Comando

**C** -> (8) (11) id [ "[" (9) Exp "]" ] := (10) Exp; |  
for "(" {J}; Exp; {J} ")" ( C | "{" {C} + " " ) |  
if "(" Exp ")" then ( C | "{" {C} "}" ) [else ( C | "{" {C} "}" )] |  
readln "(" (12) id [ "[" (13) Exp "]" ] "; |  
write "(" A+ " " ; |  
writeln "(" {A} + " " ; |  
;  
**J** -> C [,]

#### Expressão

**A** -> Exp [,]

**Exp** -> ExpS [ = | <> | < | > | <= | >= ExpS ]

**ExpS** -> [ + | - ] Termo ( + | - | or ) Termo

**Termo** -> F ( \* | / | % | and ) F

**F** -> not F | "(" Exp ")" | (14) constante | (15) id [ "[" (16) Exp "]" ]

#### 3.2. Esquema de tradução

(1) SE boolean -> novoSimbolo.tipo = 1

SENÃO SE int -> novoSimbolo.tipo = 2

SENÃO -> novoSimbolo.tipo = 3

(2) SE simbolo.tipo == 0 -> **ERRO** (Identificador já declarado)

SENÃO simbolo.tipo = novoSimbolo.tipo, novoSimbolo.classe = 1 (É uma variável)

(3) SE boolean -> simboloAnterior.tipo = 1

SENÃO SE int -> simboloAnterior.tipo = 2

SENÃO SE char -> simboloAnterior.tipo = 3

(4) SE novoSimbolo.tipo != simboloAnterior.tipo -> **ERRO** (Classe de identificador incompatível)

(5) SE int(simbolo.lexema) >= 8000 -> **ERRO** (Tamanho do vetor excede o máximo permitido)

(6) SE simbolo.classe != 0 -> **ERRO** (Identificador já declarado)

SENÃO simbolo.classe = 2 (É uma Constante)

(7) simbolo.tipo = simboloAnterior.tipo

(8) SE simbolo.tipo == 0 -> **ERRO** (Identificador não declarado)

(9) SE `int(simbolo.lexema) >= 8000` ->**ERRO** (Tamanho do vetor excede o máximo permitido)

(10) SE `boolean` ->`simboloAnterior.tipo = 1`  
SE `inteiro` ->`simboloAnterior.tipo = 2`  
SE `char` ->`simboloAnterior.tipo = 3`

(11) SE `simbolo.tipo != simboloAnterior.tipo` ->**ERRO** (Classe de identificador incompatível)

(12) SE `simbolo.tipo == 0` ->**ERRO** (Identificador não declarado)

(13) SE `int(simbolo.lexema) >= 8000` ->**ERRO** (Tamanho do vetor excede o máximo permitido)

(14) SE `boolean` ->`simboloAnterior.tipo = 1`  
SENÃO SE `int` ->`simboloAnterior.tipo = 2`  
SENÃO SE `char` ->`simboloAnterior.tipo = 3`

(15) SE `simbolo.tipo == 0` ->**ERRO** (Identificador não declarado)

(16) SE `int(simbolo.lexema) >= 8000` ->**ERRO** (Tamanho do vetor excede o máximo permitido)