

TRABALHO DE LINGUAGENS FORMAIS

Bruno Henrique Xavier - 2016010208

Caíque Cléber Dias de Rezende – 2016003750

Marcelo Henrique Rocha – 2016002064

12 de Novembro de 2019

1 OBJETIVOS DO TRABALHO

Este é um trabalho feito para a disciplina de Compilador, ministrada pela Profª Drª Thatyana de Faria Piola Seraphim na Universidade Federal de Itajubá.

Este trabalho tem por objetivo dar continuidade ao trabalho realizado no semestre passado na disciplina de Linguagens Formais, construindo um compilador funcional para a linguagem descrita pelo grupo anteriormente.

2 CARACTERÍSTICAS DA LINGUAGEM

2.1 Caracteres suportados

Os caracteres suportados pela linguagem são:

letra = [a-zA-Z]

dígito = [0-9]

2.2 Tipos de dados suportados

Os tipos de dados suportados pela linguagem bem como suas regras de uso são apresentados a seguir:

inteiro = dígito⁺

real = dígito⁺ (. dígito⁺)

caracter = letra

string = letra (letra)⁺

2.3 Blocos de comandos

Os blocos de comando da linguagem e suas regras de uso são:

blocoComandos = tokenBegin ... tokenEnd

2.4 Operadores relacionais

| Símbolo | Significado |
|---------|---------------------|
| = | é igual a? |
| ~= | é diferente de? |
| > | é maior que? |
| < | é menor que? |
| >= | é maior ou igual a? |
| <= | é menor ou igual a? |

A regra de uso para os operadores relacionais é:

relacional = identifier (= | ~= | > | < | >= | <=) identifier

2.5 Operadores lógicos

| Símbolo | Significado |
|---------|-------------|
| and | E |
| or | Ou |

A regra de uso para os operadores lógicos é:

lógico = (identifier (and | or) identifier)

2.6 Operadores aritméticos

| Símbolo | Significado |
|---------|---------------|
| + | soma |
| - | subtração |
| * | multiplicação |
| / | divisão |

| | |
|---|---------------|
| ^ | exponenciação |
| % | módulo |

A regra de uso para os operadores aritméticos é:

aritmético = identifier (+ | - | * | / | ^ | %) identifier

2.7 Símbolos especiais

A linguagem suporta os seguintes símbolos especiais:

| | | |
|-----------------|---|------------|
| ponto | = | “ . ” |
| pontoVírgula | = | “ . ” ; |
| abreParênteses | = | “ (” |
| fechaParênteses | = | “) ” |

2.8 Comandos da linguagem

A seguir define-se todos os comandos possíveis que a linguagem apresenta, bem como as regras de cada um deles:

| | |
|-----------------|---|
| identifier: | char |
| operação: | relacional aritmético |
| atribuição: | identifier is (identifier operação); |
| saída: | (nout cout sout) identifier ; |
| condicional: | if (relacional) begin identifier is expNum ; end ; |
| repetiçãoWhile: | while (relacional) begin identifier is aritmético ; end ; |

2.9 Lista de palavras reservadas

A seguinte lista de palavras é reservada para a linguagem:

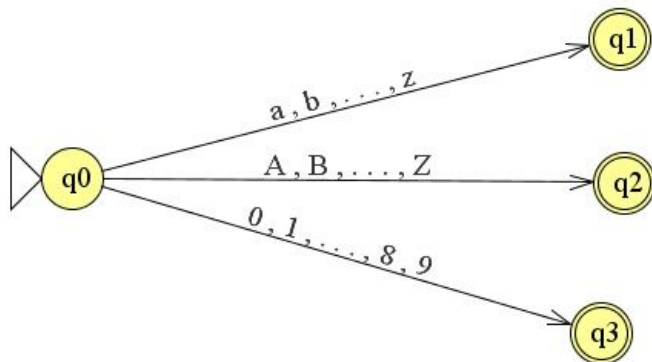
int, float, char, str, is,
and, or, if, while, nout,
cout, sout, begin,
end, exit.

3 TABELA DE TOKENS

| Expressão | Token |
|-----------------------|-------------------|
| is | tokenIs |
| int | tokenInt |
| float | tokenFloat |
| char | tokenChar |
| str | tokenStr |
| [0-9]+ | tokenIntNumber |
| [0-9]*\.[0-9]+"f"? | tokenFloatNumber |
| [a-zA-Z] | tokenIdentifier |
| [a-zA-Z][_a-zA-Z0-9]* | tokenWord |
| begin | tokenBegin |
| end | tokenEnd |
| = | tokenEqual |
| ~= | tokenNotEqual |
| < | tokenLesser |
| <= | tokenLesserEqual |
| > | tokenGreater |
| >= | tokenGreaterEqual |
| if | tokenIf |
| while | tokenWhile |
| and | tokenAnd |
| or | tokenOr |
| nout | tokenOutputNum |
| cout | tokenOutputChar |
| sout | tokenOutputStr |
| exit | tokenExit |

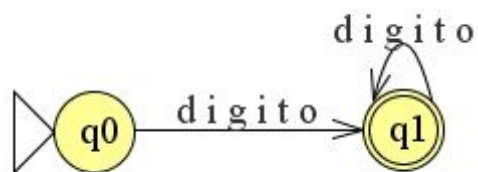
4 AUTÔMATOS

4.1 Caracteres suportados

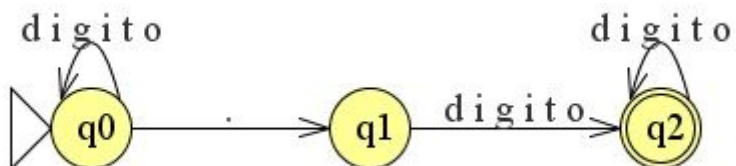


4.2 Tipos de dados suportados

4.2.1 inteiro



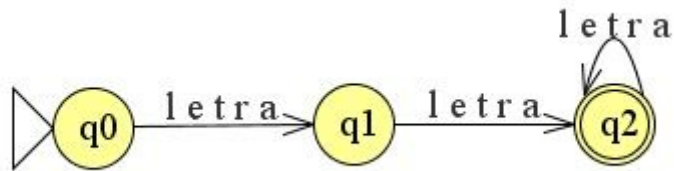
4.2.2 real



4.2.3 caracter



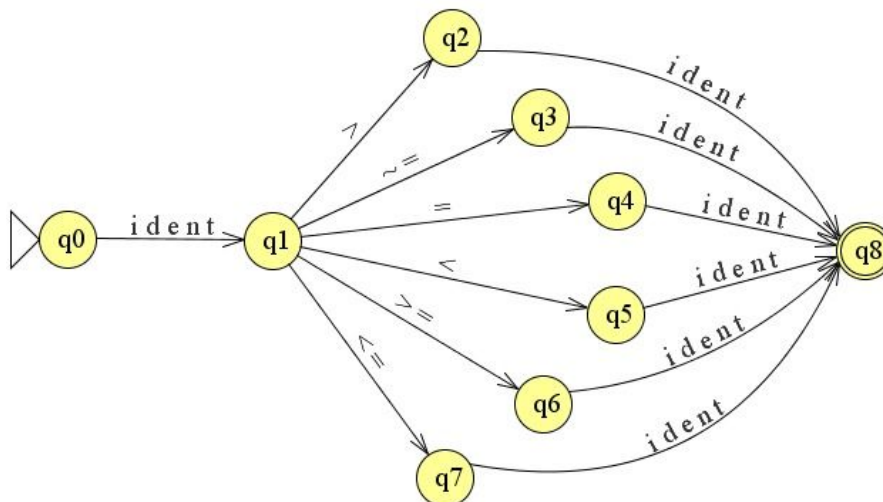
4.2.4 string



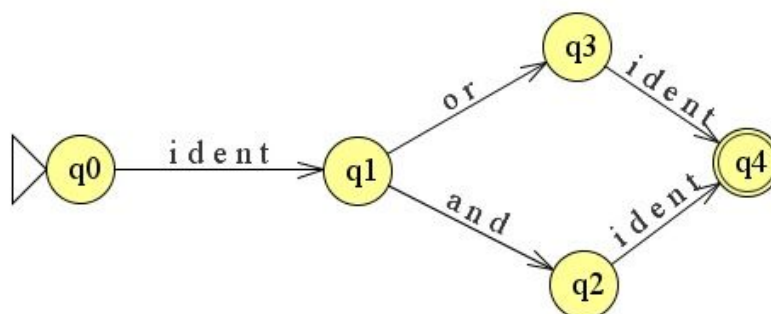
4.3 blocoComandos



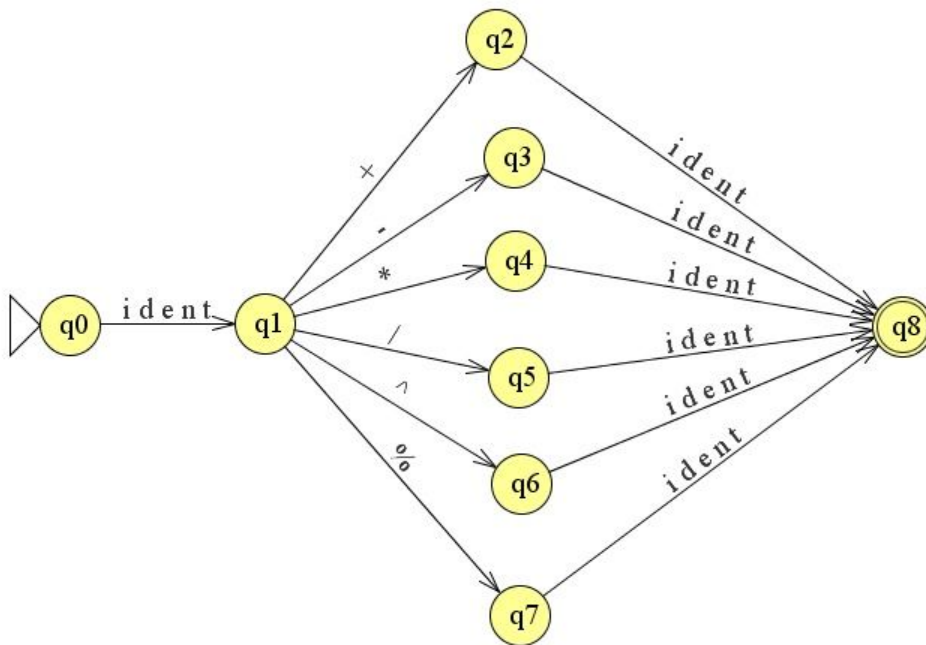
4.4 Operadores relacionais



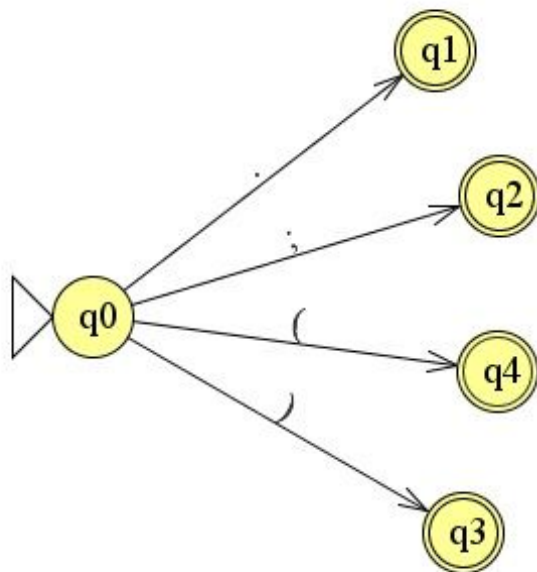
4.5 Operadores lógicos



4.6 Operadores aritméticos

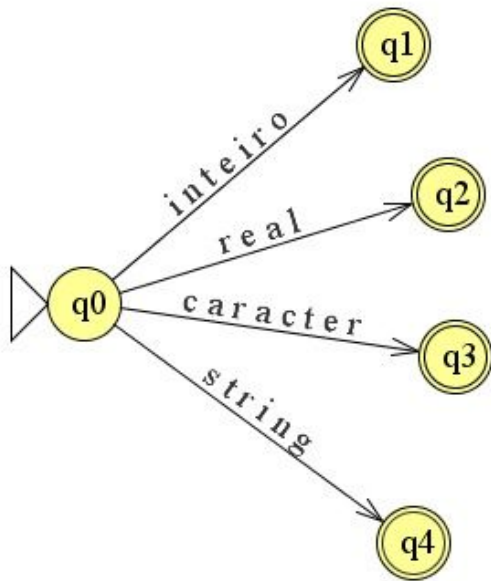


4.7 Símbolos especiais

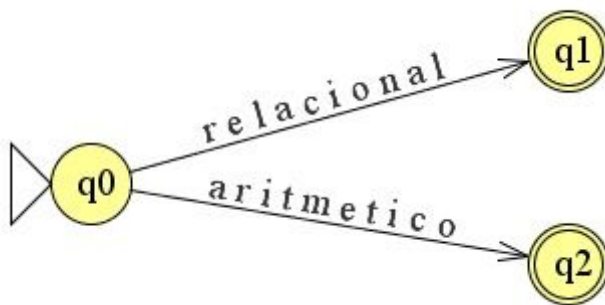


4.8 Comandos da linguagem

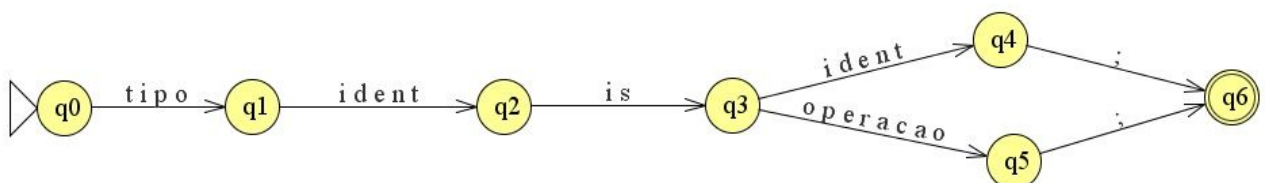
4.8.1 tipo



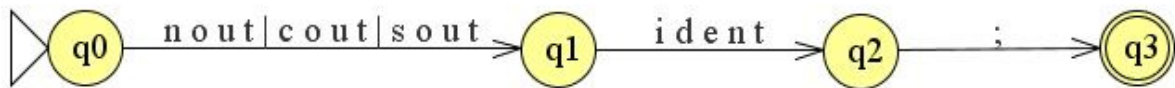
4.8.2 operação



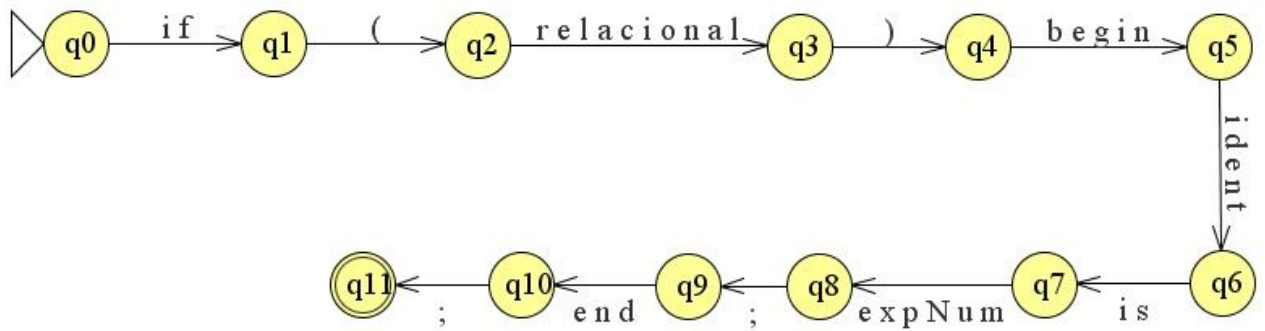
4.8.3 atribuição



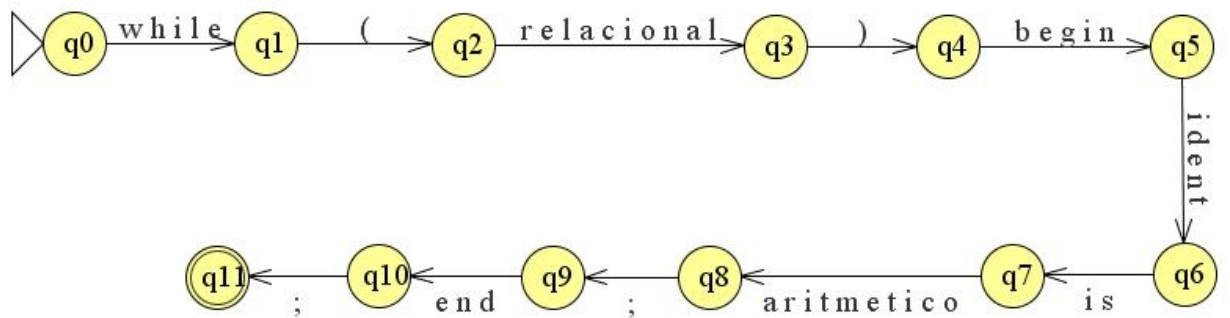
4.8.4 saída



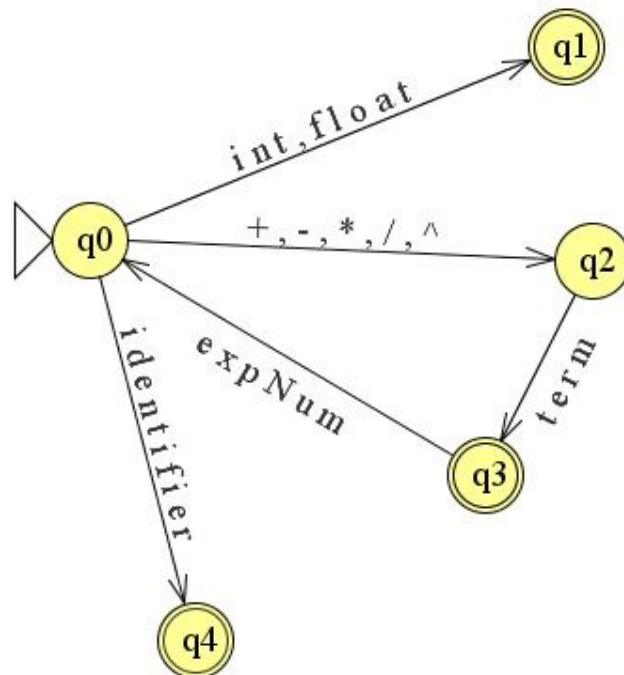
4.8.5 condicional



4.8.6 repetiçãoWhile



4.8.7 expNum



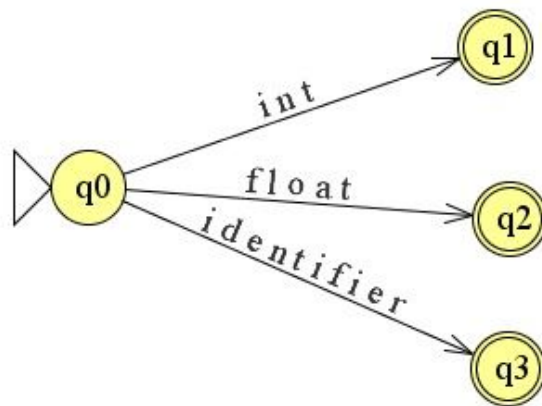
4.8.8 expChar



4.8.9 expStr



4.8.10 term



4.8.11 line

