Análise Léxica

Noemi Pereira Scherer[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Campo Mourão – PR – Brasil

{noemischerer13}@gmail.com

Abstract.

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

1. Introdução

Análise léxica é responsável por analisar a entrada de linhas de caracteres do código-fonte de um programa de computador e produzir uma sequência de símbolos denominados *tokens*, que sendo melhores manipulados por um parser (leitor de saída) [Wikipedia 2017].

1.1. Objetivo

O principal objetivo desse trabalho é analisar um determinado código escrito na linguagem T++ e retornar o conjunto de *tokens* encontrado.

2. A Linguagem T++

A linguagem de programação T++ foi desenvolvida especialmente para ser utilizada na disciplina de compiladores. Ela é uma linguagem simples, contendo algumas palavras reservadas, símbolos e arranjos uni e bidimensionais, não é sensível ao contexto, ou seja, palavras iguais com letras maiúsculas e minúsculas são consideradas as mesmas. A Tabela 1 mostra todas as palavras reservadas e símbolos que a linguagem T++ permite.

A construção do código é inteiramente em Português. Os números podem ser inteiros, flutuantes (notação científica ou não), e as declarações de variáveis devem obrigatoriamente começar com letras precedente de várias letras e/ou números.

Tabela 1. *Tokens* permitidos pela Linguagem T++

Palavras Reservadas	Símbolos
se	+ soma
então	- subtração
senão	* multiplicação
fim	/ divisão
repita	= igualdade
flutuante	, vírgula
retorna	:= atribuição
até	< menor
leia	> maior
escreve	<= menor-igual
inteiro	>= maior-igual
notação científica	() abre e fecha parênteses
	: dois pontos
	[] abre e fecha colchetes
	{} comentário
	| | ou-lógico
	&& e-lógico
	! negação

3. Métodos e Resultados

3.1. Expressões Regulares e Autômatos

Uma expressão regular é responsável por identificar uma cadeias de caracteres de uma determinada linguagem, como palavras ou padrões. Elas são escritas numa linguagem formal que pode ser interpretada por um processador de expressão regular [AHO 2008].

Essa expressão pode ser associada com a aritmética, ao invés dela denotar um número (como 2+2), a expressão regular denota uma linguagem regular. Por exemplo: a0+, que resultará em {”a0”, ”a00”, ”a000”, ...} [AHO 2008].

As expressões regulares são utilizadas para a especificação léxica de uma determinada linguagem.

Para cada *token* permitido na Linguagem T++ é criado uma expressão regular, o qual tem como objetivo analisar uma cadeia de caracteres no código. As expressões regulares utilizadas nesse trabalho estão descritas na Tabela 2.

Tabela 2. Expressões regulares

Tokens	Expressão regulares
ID	[a-zA-Zà-úÀ-Ú][_0-9a-zà-úA-ZÀ-Ú]*
Notação Científica	[0-9]+(\.[0-9]+)*(e| E)+(\+ |\-)*[0-9]+(\.[0-9])*
Flutuante	[0-9]+(\.[0-9]+)(e(\+| \-)?(d+))?
Inteiro	[0-9]+
Comentário	{ [^\{ ^\}] }
Nova Linha	\n+

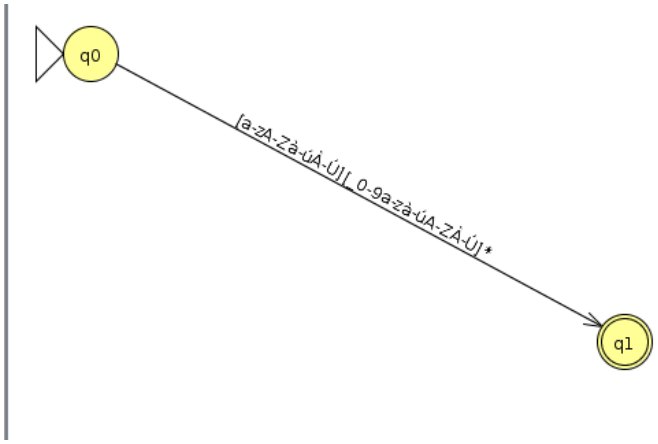

Figura 1. Expressão regular resumida do ID

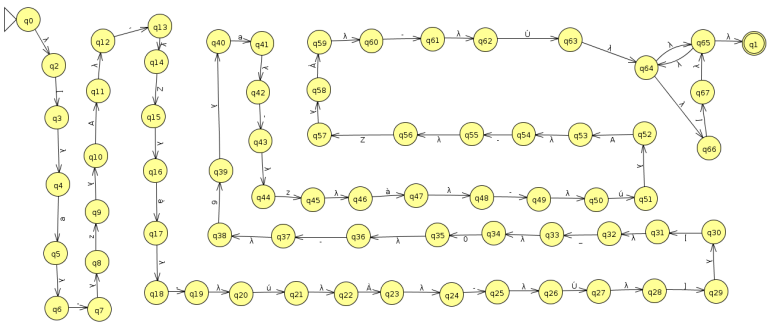

Figura 2. Expressão regular completa do ID

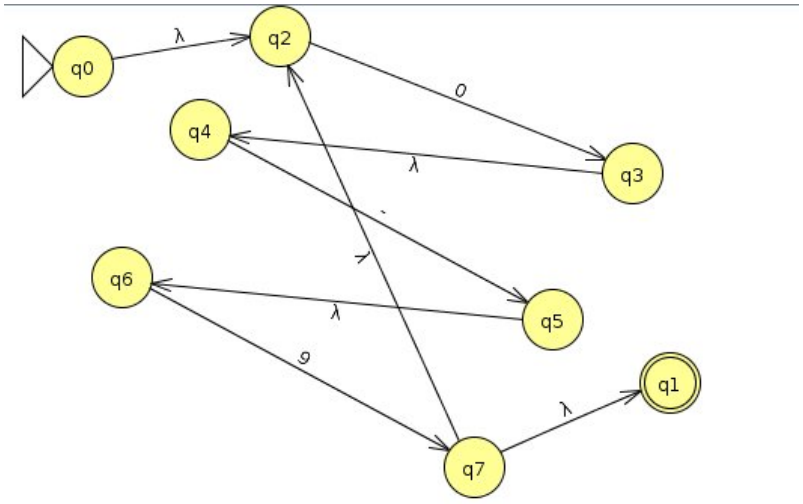

Figura 3. Expressão regular do número inteiro

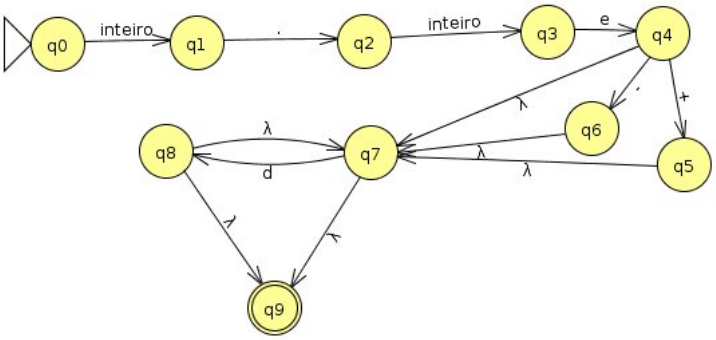

Figura 4. Expressão regular do número flutuante

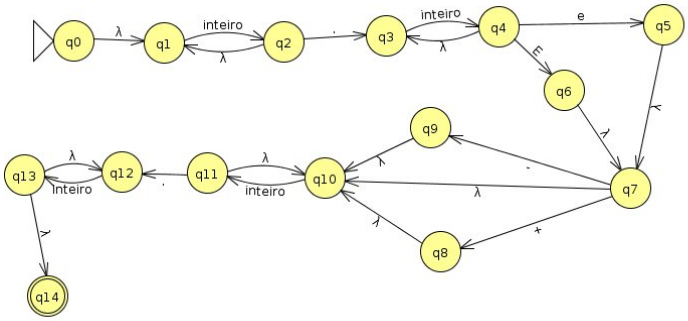

Figura 5. Expressão regular do número flutuante em notação científica

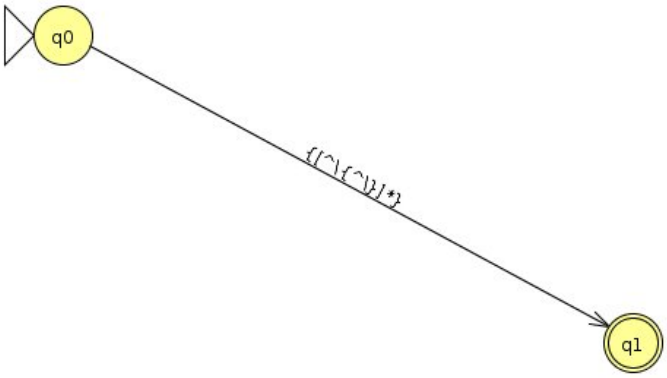

Figura 6. Expressão regular resumida do comentário

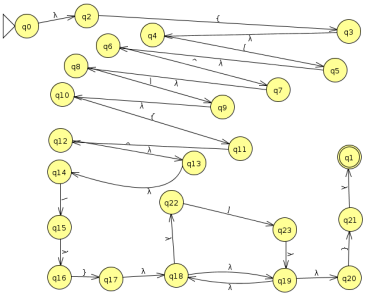

Figura 7. Expressão regular completa do comentário

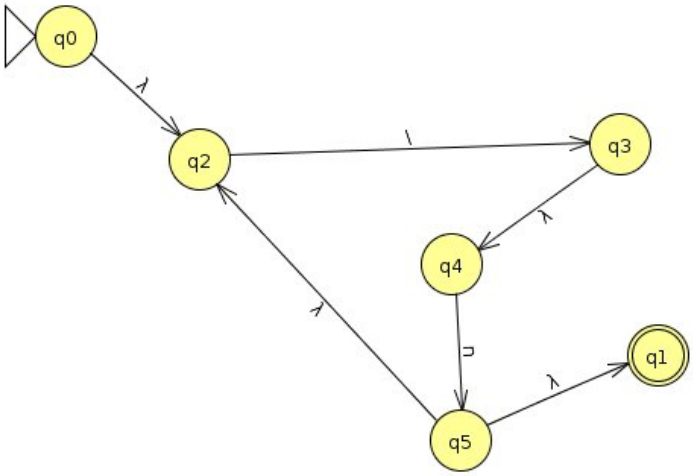

Figura 8. Expressão regular da nova linha

Referências

AHO, A. V. (2008). *Compuladores: PrincÃpios, tÃ©cnicas e ferramentas*. SP: Pearson Addison-Wesley, 2nd edition.

Wikipedia (2017). AnÃ¡lise lÃ©xica. `https://pt.wikipedia.org/wiki/An%C3%A1lise_l%C3%A9xica`. Acessado em 25-03-2018.