



TRABAJO PRÁCTICO N°2

GRUPO N° 16

CURSO: K2152

PROFESOR: Ing. Carboni Miguel

ASISTE LOS DÍAS: MARTES

EN EL TURNO: Noche

INTEGRANTES PRESENTES EL DÍA QUE SE REALIZÓ	
--	--

Dario Victor Ozuna	Mauricio Lopez Llorenty
--------------------	-------------------------

Raul Alejandro Sequeira	Tomás Gimenez
-------------------------	---------------

Juan Santiago Hiriart	Maximiliano Villca Cayampi
-----------------------	----------------------------

Objetivo

El objetivo principal de este trabajo es desarrollar un sistema capaz de reconocer expresiones matemáticas. Para lograr esto, se emplearán principios de modularización y se aplicarán conceptos relacionados con autómatas finitos. Además, se utilizarán herramientas de compilación y construcción de ejecutables como parte del proceso. El resultado final será la implementación del autómata finito determinístico en ANSI C.

Material utilizado

“El lenguaje de programación C, 2da Edición” [K&R 1988]
“Sintaxis y Semántica de los Lenguajes” Jorge D. Muchnik, 2010

Desarrollo y marco teórico

Un lenguaje regular es un lenguaje formal que puede ser definido por expresiones regulares, generador por una gramática regular y reconocido por un autómata finito. Los autómatas finitos para reconocer los lenguajes regulares necesitan estar formados por un alfabeto, un conjunto de estados finitos, funciones de transición.

El tipo de autómata finito que nos pide el trabajo es autómata finito determinístico, que es aquel que aparte de compartir la expresión formal, tiene la característica funcional es que para cualquier estado en que se encuentre el autómata en un momento dado, la lectura de un carácter determina, SIN AMBIGÜIDADES, cuál será el estado de llegada en la próxima transición.

Procedimiento

Luego de la relectura del material bibliográfico, decidimos hacer realizar un bosquejo del autómata que reconozca el lenguaje formado por expresiones matemáticas del alfabeto { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, (,) }. Y con ello empezamos a crear la expresión formal del autómata.

Previo a la diagramación del autómata se identificó la categoría léxica de cada token de la siguiente manera

Categoría léxica	Lexema
------------------	--------

Operador	$+, -, *, /, (,)$
Constante numérica	0,1,2,3,4,5,6,7,8,9

El comportamiento de nuestro autómata estará definido por la BNF:

expresion : operador
 cteNumerica
 expresion cteNumerica
 expresion operador

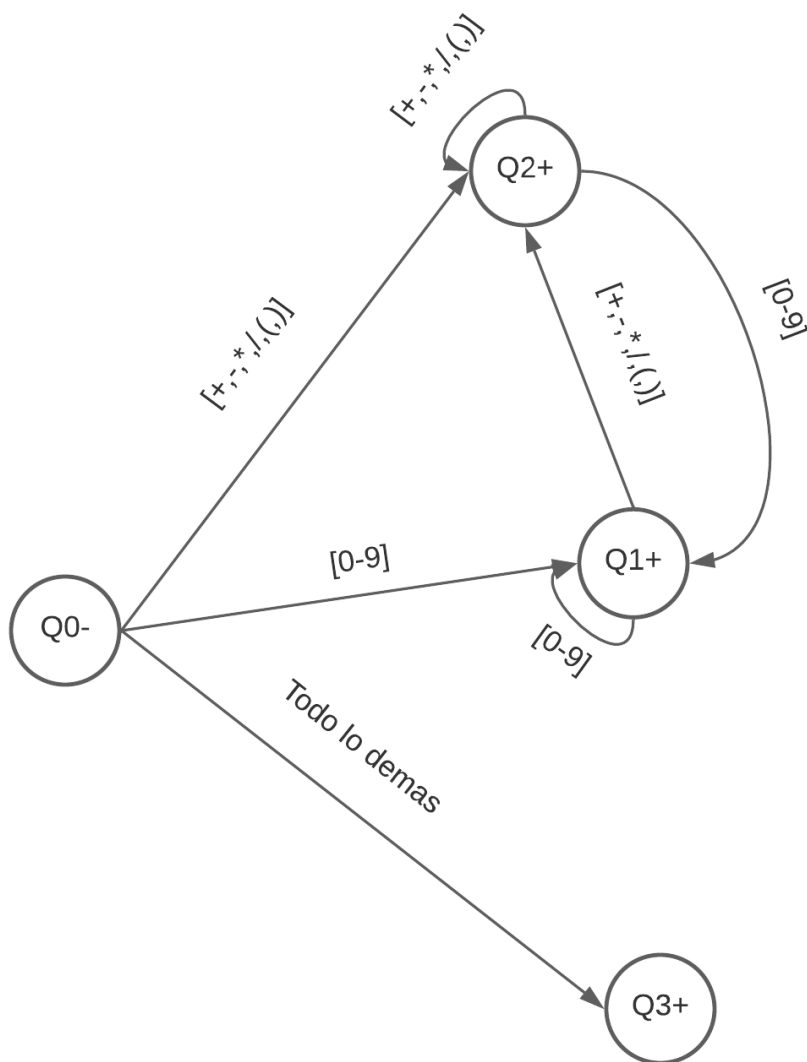
cteNumerica: unoDe 0 1 2 3 4 5 6 7 8 9

operador: unoDe +, -, *, /, (,)

En el que se concluye en que la expresión formal de nuestro autómata es una quintupla que consta de:

- Alfabeto $L: \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, (,) \}$
- Un conjunto finito de estados: $\{Q0; Q1; Q2; Q3\}$
- Una Tabla de transición
- Un estado inicial: $\{Q0\}$
- Un conjunto de estados finales: $\{Q1; Q2; Q3\}$

Llegando al siguiente diseño que nos indicará cuál fue la categoría léxica del último carácter reconocido (Ver siguiente pagina)



Se podría documentar aún más el comportamiento del autómata dado con la siguiente tabla que representa sus transiciones

Estado	Operador	Constante Num	Otro
Q0-	Q2	Q1	Q3
Q1+	Q2	Q1	Q3
Q2+	Q2	Q1	Q3
Q3	Q3	Q3	Q3

Posteriormente se sometió al AFD, creado, a diferentes cadenas de caracteres, para probar su efectividad. Con resultados positivos brindados por evaluación se procedió a la creación de un programa, en lenguaje ANSI C, que reproduce la función de nuestro autómata. A dicho programa también se lo somete a pruebas.