

Nombre de la práctica	AUTOMATAS FINITOS-U3			No.	2
Asignatura:	LENGUAJES Y AUTÓMATAS I	Carrera:	INGENIERÍA EN SISTEMAS COMPUTACIONALES-3501	Duración de la práctica (Hrs)	5 horas

NOMBRE DEL ALUMNO: Jesús Navarrete Martínez

GRUPO:

I. Competencia(s) específica(s):

Crea y reconoce autómatas finitos en un lenguaje de programación para la solución de un problema.

Encuadre con CACEI: Registra el (los) atributo(s) de egreso y los criterios de desempeño que se evaluarán en la materia.

No. atributo	Atributos de egreso del PE que impactan en la asignatura	No. Criterio	Criterios de desempeño	No. Indicador	Indicadores
2	El estudiante diseñará esquemas de trabajo y procesos, usando metodologías congruentes en la resolución de problemas de Ingeniería en Sistemas Computacionales	CD1	Identifica metodologías y procesos empleados en la resolución de problemas	I1	Identificación y reconocimiento de distintas metodologías para la resolución de problemas
		CD2	Diseña soluciones a problemas, empleando metodologías apropiadas al área	I1	Uso de metodologías para el modelado de la solución de sistemas y aplicaciones
				I2	Diseño algorítmico (Representación de diagramas de transiciones)
3	El estudiante plantea soluciones basadas en tecnologías empleando su juicio ingenieril para valorar necesidades, recursos y resultados esperados.	CD1	Emplea los conocimientos adquiridos para el desarrollar soluciones	I1	Elección de metodologías, técnicas y/o herramientas para el desarrollo de soluciones
				I2	Uso de metodologías adecuadas para el desarrollo de proyectos
				I3	Generación de productos y/o proyectos
		CD2	Analiza y comprueba resultados	I1	Realizar pruebas a los productos obtenidos
				I2	Documentar información de las pruebas realizadas y los resultados

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

Laboratorio de cómputo y equipo de cómputo personal.

III. Material empleado:

- Equipo de cómputo
- Software para desarrollo

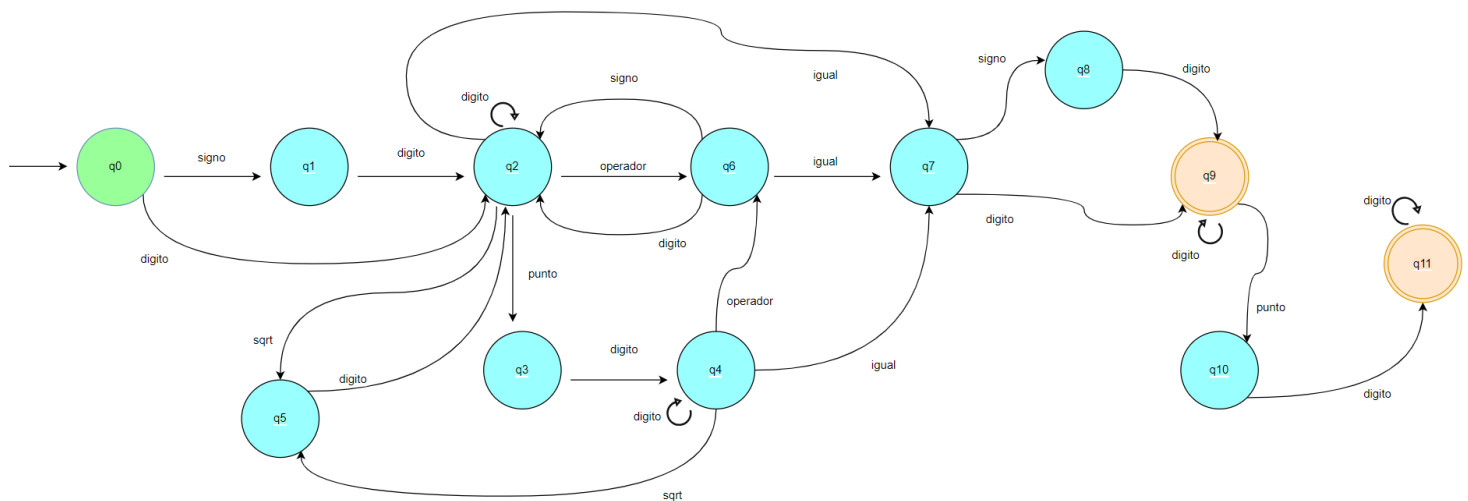
IV. Desarrollo de la práctica:

EJERCICIO 1

DESCRIPCIÓN DEL PROBLEMA:

Genera un autómata finito no determinístico que permita reconocer números enteros y decimales, para reconocer las operaciones aritméticas básicas (suma, resta, división, multiplicación, potencia), así como la raíz cuadrada. El ultimo símbolo que se genere en las cadenas deberá considerar el igual y la designación del resultado.

AUTÓMATA FINITO NO DETERMINÍSTICO (Diagrama):



DEFINICIÓN FORMAL: $AFN = \langle S, \Sigma, p, i, F \rangle$

- $S = \{ q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11 \}$
- $\Sigma = \{ \text{signo, digito, operador, punto, sqrt, igual} \}$
 - ❖ $\text{signo} = \{ +, - \}$
 - ❖ $\text{digito} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$
 - ❖ $\text{operador} = \{ +, -, *, \wedge, / \}$
 - ❖ $\text{punto} = \{ . \}$
 - ❖ $\text{sqrt} = \{ \sqrt{} \}$
 - ❖ $\text{igual} = \{ = \}$
- $P = \{ (q0, \text{signo}, q1), (q0, \text{digito}, q2), (q1, \text{digito}, q1), (q2, \text{digito}, q2), (q2, \text{operador}, q6), (q2, \text{punto}, q3), (q2, \text{sqrt}, q5), (q2, \text{igual}, q7), (q3, \text{digito}, q4), (q4, \text{digito}, q4), (q4, \text{operador}, q6), (q4, \text{sqrt}, q5), (q4, \text{igual}, q7), (q5, \text{digito}, q2), (q6, \text{signo}, q2), (q6, \text{digito}, q2), (q6, \text{igual}, q7), (q7, \text{signo}, q8), (q7, \text{digito}, q9), (q8, \text{digito}, q9), (q9, \text{digito}, q9), (q9, \text{punto}, q10), (q10, \text{digito}, q11), (q11, \text{digito}, q11) \}$

TABLA DE TRANSICIÓN:

		signo	digito	operador	punto	sqrt	igual
inicial	q0	q1	q2				
	q1		q1				
	q2		q2	q6	q3	q5	q7
	q3		q4				
	q4		q4	q6		q5	q7
	q5		q2				
	q6	q2	q2				q7
	q7	q8	q9				
	q8		q9				
final	q9		q9		q10		
	q10		q11				
final	q11		q11				

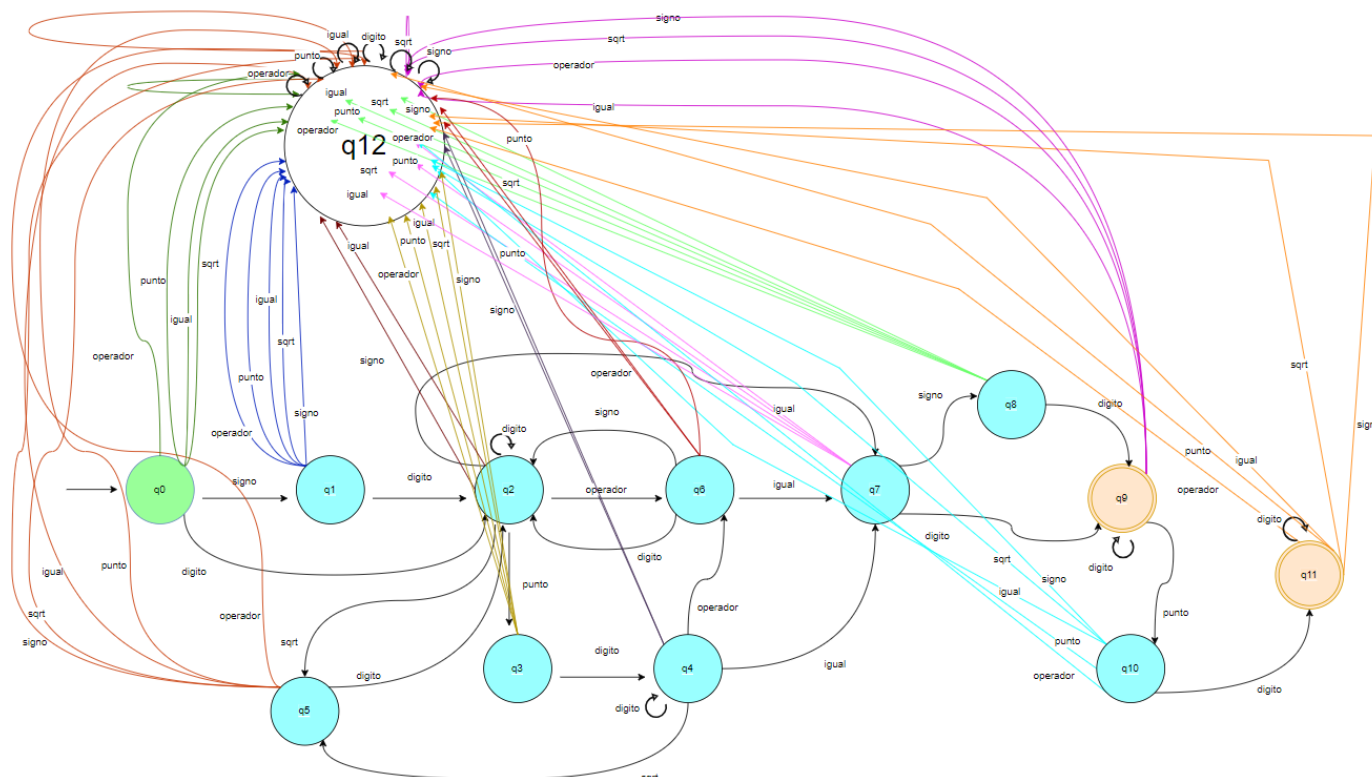
- $i = \{ q0 \}$
- $F = \{ q9, q11 \}$

LENGUAJE POR COMPRENSIÓN:

$L = \{ w \in \{ \text{signo, digito, operador, punto, sqrt, igual} \}^* \mid w \text{ cumple con } ((\text{signo digito+ punto digito+} (\text{operador|raiz}) \text{ signo digito+ punto digito+ (operador| igual)})^* \text{ signo digito+ punto digito+ }) \}$

CONVERSIÓN A AUTÓMATA FINITO DETERMINÍSTICO:

✓ Si se puede convertir modificándolo estructuralmente



DEFINICIÓN FORMAL UNA VEZ CONVERTIDO: $AFD = \langle S, \Sigma, \delta, i, F \rangle$

• $S = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}\}$

• $\Sigma = \{\text{signo, digito, punto, operador, igual, raiz}\}$

❖ signo = {positivo + , negativo -}

❖ digito = {0,1,2,3,4,5,6,7,8,9}

❖ operador = {+, -, *, /, ^}

❖ punto = {.}

❖ igual = {=}

❖ raíz = {√}

• $\delta = \{ (q_0, \text{signo}, q_1), (q_0, \text{digito}, q_2), (q_0, \text{operador}, q_{12}), (q_0, \text{punto}, q_{12}), (q_0, \text{raiz}, q_{12}), (q_0, \text{igual}, q_{12}), (q_1, \text{digito}, q_1), (q_1, \text{signo}, q_{12}), (q_1, \text{operador}, q_{12}), (q_1, \text{punto}, q_{12}), (q_1, \text{raiz}, q_{12}), (q_1, \text{igual}, q_{12}), (q_2, \text{signo}, q_{12}), (q_2, \text{digito}, q_2), (q_2, \text{operador}, q_6), (q_2, \text{punto}, q_3), (q_2, \text{raiz}, q_5), (q_2, \text{igual}, q_7), (q_3, \text{digito}, q_4), (q_3, \text{signo}, q_{12}), (q_3, \text{operador}, q_{12}), (q_3, \text{punto}, q_{12}), (q_3, \text{raiz}, q_{12}), (q_3, \text{igual}, q_{12}), (q_4, \text{digito}, q_4), (q_4, \text{operador}, q_6), (q_4, \text{raiz}, q_5), (q_4, \text{igual}, q_7), (q_4, \text{signo}, q_{12}), (q_4, \text{punto}, q_{12}), (q_5, \text{digito}, q_2), (q_5, \text{signo}, q_{12}), (q_5, \text{operador}, q_{12}), (q_5, \text{punto}, q_{12}), (q_5, \text{raiz}, q_{12}), (q_5, \text{igual}, q_{12}), (q_6, \text{signo}, q_2), (q_6, \text{digito}, q_2), (q_6, \text{igual}, q_7), (q_6, \text{operador}, q_{12}), (q_6, \text{punto}, q_{12}), (q_6, \text{raiz}, q_{12}), (q_7, \text{signo}, q_8), (q_7, \text{digito}, q_9), (q_7, \text{operador}, q_{12}), (q_7, \text{punto}, q_{12}), (q_7, \text{raiz}, q_{12}), (q_7, \text{igual}, q_{12}), (q_8, \text{digito}, q_9), (q_8, \text{signo}, q_{12}), (q_8, \text{operador}, q_{12}), (q_8, \text{punto}, q_{12}), (q_8, \text{raiz}, q_{12}), (q_8, \text{igual}, q_{12}), (q_9, \text{digito}, q_9), (q_9, \text{punto}, q_{10}), (q_9, \text{signo}, q_{12}), (q_9, \text{operador}, q_{12}), (q_9, \text{raiz}, q_{12}), (q_9, \text{igual}, q_{12}), (q_{10}, \text{digito}, q_{11}), (q_{10}, \text{signo}, q_{12}), (q_{10}, \text{operador}, q_{12}), (q_{10}, \text{punto}, q_{12}), (q_{10}, \text{raiz}, q_{12}), (q_{10}, \text{igual}, q_{12}), (q_{11}, \text{digito}, q_{11}), (q_{11}, \text{signo}, q_{12}), (q_{11}, \text{operador}, q_{12}), (q_{11}, \text{punto}, q_{12}), (q_{11}, \text{raiz}, q_{12}), (q_{11}, \text{igual}, q_{12}), (q_{12}, \text{signo}, q_{12}), (q_{12}, \text{digito}, q_{12}), (q_{12}, \text{operador}, q_6), (q_{12}, \text{punto}, q_3), (q_{12}, \text{raiz}, q_5), (q_{12}, \text{igual}, q_7), \}$

TABLA DE TRANSICIÓN:

		signo	digito	operador	punto	sqrt	igual
inicial	q0	q1	q2	q12	q12	q12	q12
	q1	q12	q1	q12	q12	q12	q12
	q2	q12	q2	q6	q3	q5	q7
	q3	q12	q4	q12	q12	q12	q12
	q4	q12	q4	q6	q12	q5	q7
	q5	q12	q2	q12	q12	q12	q12
	q6	q2	q2	q12	q12	q12	q7
	q7	q8	q9	q12	q12	q12	q12
	q8	q12	q9	q12	q12	q12	q12
final	q9	q12	q9	q12	q10	q12	q12
	q10	q12	q11	q12	q12	q12	q12
final	q11	q12	q11	q12	q12	q12	q12
	q12	q12	q11	q12	q12	q12	q12

• $i = \{q_0\}$

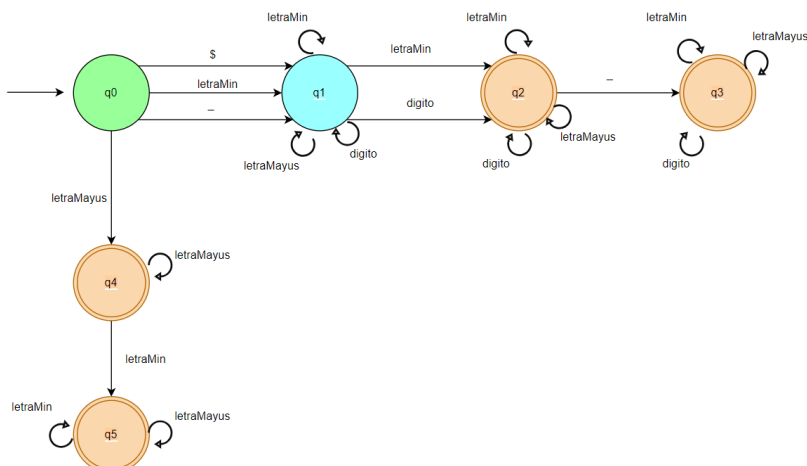
• $F = \{q_9, q_{11}\}$

EJERCICIO 2

DESCRIPCIÓN DEL PROBLEMA:

Genera el autómata finito no determinístico que reconozca nombres de variables, constantes y clases.

AUTÓMATA FINITO NO DETERMINÍSTICO:



DEFINICIÓN FORMAL: $AFN = \langle S, \Sigma, p, i, F \rangle$

- $S = \{ q0, q1, q2, q3, q4, q5 \}$
- $\Sigma = \{ \text{letraMin}, \text{letraMayus}, \text{digito}, \text{carácter}, \text{guion bajo} \}$
 - $\text{letraMin} = \{ [a-z] \}$
 - $\text{letraMayus} = \{ [A-Z] \}$
 - $\text{digito} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$
 - $\text{carácter} = \{ \$ \}$
 - $\text{guion bajo} = \{ _ \}$
- $P = \{ (q0, \text{letraMin}, q1), (q0, \text{letraMayus}, q4), (q0, \text{carácter}, q1), (q0, \text{guion bajo}, q1), (q1, \text{letraMin}, q1), (q1, \text{letraMin}, q2), (q1, \text{letraMayus}, q1), (q1, \text{digito}, q1), (q1, \text{digito}, q2), (q2, \text{letraMin}, q2), (q2, \text{letraMayus}, q2), (q2, \text{digito}, q2), (q2, \text{guion bajo}, q3), (q3, \text{letraMin}, q3), (q3, \text{letraMayus}, q3), (q3, \text{digito}, q3), (q4, \text{letraMin}, q5), (q4, \text{letraMayus}, q4), (q5, \text{letraMin}, q5), (q5, \text{letraMayus}, q5) \}$

TABLA DE TRANSICIÓN:

	letraMin	letraMayus	digito	carácter	guion bajo
q0	q1	q4		q1	q1
q1	q1,q2	q1	q1,q2		
q2	q2	q2	q2		q3
q3	q3	q3	q3		
q4	q5	q4			
q5	q5	q5			

- $i = \{ q0 \}$
- $F = \{ q2, q3, q4, q5 \}$

LENGUAJE POR COMPRENSIÓN:

- $L = \{ w \in \{ \text{letraMin}, \text{letraMayus}, \text{digito}, \text{carácter}, \text{guion bajo} \}^* \mid w \text{ cumple con } ((\$ \mid \text{letraMin} \mid _) \text{letraMin}^* \text{digito}^* \text{letraMayus}^* (\text{letraMin} \mid \text{digito}) \text{letraMin}^* \text{digito}^* \text{letraMayus}^*) \mid (\$ \mid \text{letraMin} \mid _) \text{letraMin}^* \text{digito}^* \text{letraMayus}^* (\text{letraMin} \mid \text{digito}) \text{letraMin}^* \text{digito}^* \text{letraMayus}^* _ \text{letraMin}^* \text{digito}^* \text{letraMayus}^*) \mid (\text{letraMayus}^+) \mid (\text{letraMayus}^+ (\text{letraMin}^+ \text{letraMayus}^* \text{letraMin}^*)) \}$

CONVERSIÓN A AUTÓMATA FINITO DETERMINÍSTICO:

 No se puede convertir porque afecta el lenguaje generado

V. Conclusiones:

En esta práctica se realizó la conversión de autómatas finitos no determinísticos (AFND) a autómatas finitos determinísticos (AFD), un proceso crucial en la teoría de lenguajes formales y la computación. La transformación de AFND a AFD es importante porque los autómatas deterministas son más fáciles de implementar en sistemas reales, ya que siempre tienen una única transición posible para cada símbolo de entrada. Sin embargo, este proceso puede incrementar significativamente el número de estados en el AFD, especialmente cuando el AFND presenta muchos caminos posibles en sus transiciones.

Para comprender correctamente esta conversión, fue necesario definir formalmente tanto el AFND como el AFD.

El procedimiento seguido fue el algoritmo de subconjuntos, en el cual se exploran todas las posibles combinaciones de estados que el AFND puede alcanzar en cada transición. De esta forma, se construye un AFD donde cada conjunto de estados del AFND es tratado como un único estado en el AFD resultante. Este método garantiza que el autómata determinista procese cualquier cadena de entrada de manera unívoca, sin la ambigüedad inherente de los AFND.

Definir formalmente estos autómatas es esencial para entender la estructura y funcionamiento de cada uno. La formalización no solo ayuda a tener un marco claro, sino que es un paso fundamental para poder implementar algoritmos precisos en la conversión de AFND a AFD. En resumen, la práctica permitió comprender cómo eliminar la indeterminación en autómatas, lo que resulta en sistemas más eficientes y predecibles, a pesar del posible incremento en el número de estados.