



INSTITUTO TECNOLÓGICO SUPERIOR DE CHICONTEPEC

REPORTE UNIDAD 3

CARRERA: Ingeniería en Sistemas Computacionales

MATERIA: Lenguajes y Autómatas I

NOMBRE DEL ALUMNO: Jorge Miguel Hernández Domingo

NUMERO DE CONTROL: 1717V0097

DOCENTE: Ing. Efrén Flores Cruz

LUGAR: Chicontepepec Ver.



Nombre

Unidad 3:

Tema

Automatas finitos.

Día

Mes

Año

Folio

Un automata finito (AF) o maquina de estado finito es un modelo computacional que realiza computos en forma automatica sobre una entrada para producir una salida.

Este modelo esta conformado por un alfabeto, un conjunto de estado y un conjunto de transiciones entre dichos estados. Su funcionamiento se basa en una funcion de transición, que recibe a partir de un estado inicial una cadena de caracteres pertenecientes al alfabeto (entrada) y que va leyendo dicha cadena a medida que el automata se desplaza de un estado a otro, para finalmente detenerse en un estado final o de aceptación que representa la salida.

La finalidad de los automatas finitos es la de reconocer lenguajes regulares.

3.1 Definición Formal

Formalmente, un automata finito es una 5-tupla $(Q, \Sigma, q_0, \delta, F)$

Donde:

- * Q es un conjunto finito de estados

- * Σ es un alfabeto finito

- * $q_0 \in Q$ es el estado inicial

- * $\delta: Q \times \Sigma \rightarrow Q$ es una función de transición

- * $F \subseteq Q$ es un conjunto de estados finales o de aceptación.

En el comienzo del proceso de reconocimiento de una cadena de entrada el automata finito se encuentra en el estado inicial y a medida que procesa cada simbolo de la cadena va cambiando de estado



Tema:

Día

Mes

Año

Folio

de acuerdo a lo determinado por la función de transición. Cuando se ha procesado el último de los símbolos de la cadena de entrada, el automata se detiene en el estado final del proceso. Si el estado final en el que se detuvo es un estado de aceptación entonces la cadena pertenece al lenguaje reconocido por el automata; en caso contrario la cadena no pertenece a dicho lenguaje.

3.2 Clasificación de AI

Los automatas se pueden clasificar en:

* Deterministas

Cada combinación (estado, símbolo de Entrada) produce un solo estado.

* No deterministas

Cada combinación (estado, símbolo de Entrada) produce varios estados y además son posibles las transiciones con λ .

3.3 Conversión de un AFND a AFD.

Para convertir un AFD en un AFN que reconozca el mismo lenguaje. Este algoritmo a menudo, es llamado construcción de subconjuntos, es útil para simular un AFN por medio de un programa de computadora. En la tabla de transiciones de un AFN, cada entrada es un conjunto de estados; en la tabla de transiciones de un AFD, cada entrada es tan solo un estado.



Nombre

Tema

Día

Mes

Año

Folio

La idea general tras la construcción AFN a AFD es que cada estado de AFD corresponde a un conjunto de estados del AFN. El AFD utiliza un estado para localizar todos los posibles estados en los que puede estar el AFN los pues de leer cada símbolo de la entrada. Es decir, después de leer la entrada a_1, a_2, \dots, a_n , el AFD se encuentra en un estado que representa al subconjunto T de los estados del AFN alcanzables desde el estado de inicio del AFN a lo largo de algún camino etiquetado con a_1, a_2, \dots, a_n . El número de estados del AFD puede ser exponencialmente en el número de estados del AFN pero en la práctica este par caso ocurre raramente.

Algoritmo Construcción de subconjuntos AFD a AFN)
Entradas: AFN N

Saídas: un AFD D que acepta el mismo lenguaje.

Método: El algoritmo construye una tabla de transiciones $trans_D$ para D . Cada estado del AFD es un conjunto de estados del AFN y se construye $trans_D$ de modo que D simula en paralelo todos los posibles movimientos que N puede realizar con una determinada cadena de entrada.

Se utilizan las operaciones de la sg. tabla para localizar los conjuntos de los estados del AFN. S representa un estado de AFN, y T un conjunto de estados del AFN).



Nombre:		Día		Mes		Año		Folio	
Tema:									
Operación		Descripción.							
Cerradura- \square (S)		Conjunto de estados del AFN alcanzables desde el estado S del AFN con transiciones \square solamente.							
Cerradura- \square (T)		Conjunto de estados del AFN alcanzables desde algun estado S en T con transiciones \square solamente.							
Muevo (T,a)		Conjunto de estados del AFN hacia los cuales hay una transición con el simbolo de entrada a desde algun estado S en T del AFN.							
<p>Antes de detectar el primer simbolo de entrada N se puede encontrar en cualquiera de los estados del conjunto Cerradura-\square(S₀), donde S₀ es el estado de inicio de N. Supongase que exactamente los estados del conjunto T son alcanzables desde S₀ con una secuencia de simbolos de entrada y sea a el siguiente simbolo de entrada.</p> <p>Al ver a, N puede trasladarse a cualquiera de los estados del conjunto mueve(T,a)(T,a). Cuando se permiten transiciones-\square N puede encontrarse cualquiera de los estados de cerradura \square(T,a), despues de ver la a.</p> <p>Se construyen estados D, el conjunto de estados de D y Trans D, la tabla de transiciones de D, de la sig. forma:</p> <p>Cada estado de D corresponde a un conjunto de estado de AFN en los que podria estar N despues de leer alguna secuencia de simbolos de entrada, incluidos todos las posibilidades transiciones-\square anteriores o posteriores a la lectura de simbolos. El estado de inicio de D es Cerradura\square(S₀).</p>									



Nombre		Día	Mes	Año	Folio
Tema					

Se añaden los estados y las transiciones a D utilizando el algoritmo sig:

Un estado de D es un estado de aceptación si es un conjunto de estados AFN que contenga al menos un estado de aceptación de N .

Al inicio, cerradura $\pi(S_0)$ es el único estado dentro de estados D y no está marcado;

While halla un estado no marcado T en estados D do begin
 Marcar T ;
 Por cada símbolo de entrada a do begin
 $U := \text{Cerradura-}\pi(\text{move}(T, a))$;
 If U no está en estados D then
 añadir U como estado no marcado a estados D ;
 $\text{Trans}[T, a] := U$;
 end;
end;

El cálculo de cerradura- (T) es un proceso típico de búsqueda en un grafo de nodos alcanzables desde un conjunto dado de nodos. En este caso, los estados de T son el conjunto dado de nodos, y el grafo está compuesto solamente por las aristas del AFN etiquetadas.

Por ejemplo:

La sig. Cerradura muestra otro AFN N aceptando el lenguaje $(a|b)^*abb$. Se aplica el siguiente algoritmo a N .

El estado de inicio del AFD equivalente es $\text{Cerradura-}\pi(0)$, que es $A = \{0, 1, 2, 4, 7\}$ puesto que estos son exactamente los estados alcanzables desde el estado 0 por un camino en el que todas las aristas están etiquetadas por a .



Tema:

Día

Mes

Año

Folio

Aquí, el alfabeto de símbolos de entrada es $\{a, b\}$. El algoritmo anterior indica que hay que marcar A y después calcular $\text{Cerradura-}\Pi(\text{mueva}(A, a))$.

Primero se calcula $\text{mueva}(A, a)$, el conjunto de estado de N que tiene transiciones en a desde miembros de A . Entre los estados $0, 1, 2, 4, 7$ solo 2 y 7 tienen dichas transiciones, a 3 y a 8 , de modo que:

$$\text{Cerradura-}\Pi(\text{mueva}(\{0, 2, 4, 7\}, a)) = \text{Cerradura-}\Pi(\{3, 8\}) = \{1, 2, 4, 6, 7, 8\}$$

Este conjunto se denomina B . Así $\text{trans}[A, a] = B$.

Entre los estados de A , solo 4 tienen una transición en b a 5 , de modo que el AFD tiene una transición en b desde A a $C = \text{Cerradura-}\Pi(\{5\}) = \{1, 2, 4, 5, 6, 7\}$.

Por tanto, $\text{trans}[A, b] = C$

Si se continúa este proceso con los conjuntos B y C , ahora sin marcar, finalmente se llegará al punto en que todos los conjuntos que son estados del AFD estén marcados. Esto es cierto porque "solo" hay 2 subconjuntos distintos de un conjunto de 8 estados y un conjunto, una vez marcado, queda marcado para siempre.

Los 5 conjuntos de estados realmente construidos son:

$$A = \{0, 1, 2, 4, 7\}$$

$$B = \{1, 2, 3, 4, 6, 7, 8\}$$

$$C = \{1, 2, 4, 5, 6, 7\}$$

$$D = \{1, 2, 4, 5, 6, 7, 9\}$$

$$E = \{1, 2, 4, 5, 6, 7, 10\}$$



Nombre

Tema

Día

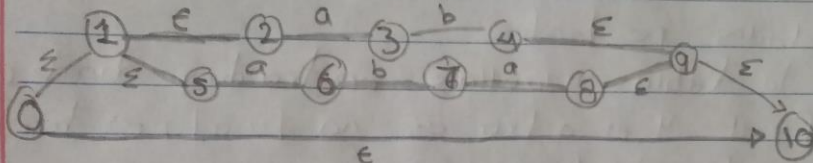
Mes

Año

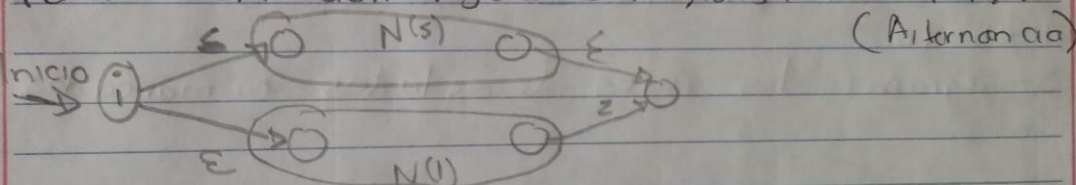
Folio

3.4 Representación de ER usando AFND.

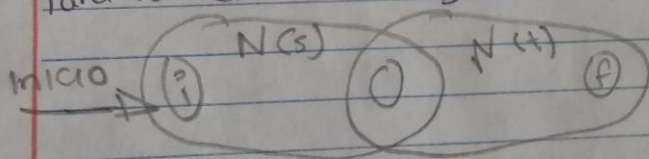
ERs, AFDs y AFNDs son mecanismos equivalentes para denotar los lenguajes regulares. En estas 3 secciones demostraremos esto mediante convertir $ER \rightarrow AFND \rightarrow AFD \rightarrow ER$. Las 2 primeras conversiones son muy relevantes en la práctica, pues permiten construir Verificadores o buscadores eficientes a partir de ERs.



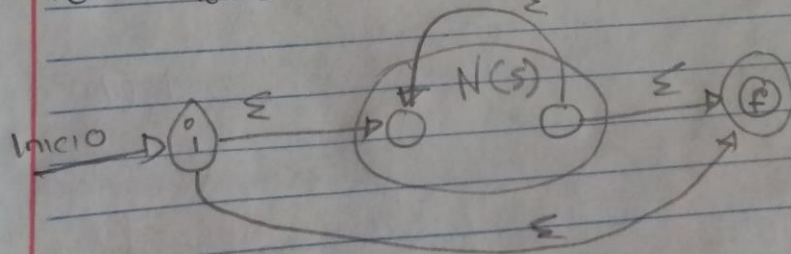
Para la expresión regular $s|t$, construir el AFND.



Para la expresión regular slt , construir el AFND (Concatenación)



Para la expresión regular s'' , construir el AFND $N(s'')$





Nombre:

Tema:

Día

Mes

Año

Folio

3.5 Minimización de Estados en un AF

Un AFD está minimizado, si todos sus estados son distinguibles y alcanzables. Un algoritmo de minimización es el siguiente:

1.- Eliminar los estados inaccesibles del autómata

2.- Construir una tabla con todos los pares (p, q) de estados restantes.

3.- Marcar en la tabla aquellas entradas donde un estado es final y el otro es no final, es decir aquellos claramente distinguibles.

4.- Para cada par (p, q) y cada símbolo a del alfabeto.

5.- agrupar los pares de estados no marcados.