

AUTÓMATAS Y LENGUAJES

LENGUAJES REGULARES- Minimización de AFD - Prop. de Clausura - AÑO 2024

LIC. EN CS. DE LA COMPUTACIÓN - 4to. AÑO

Objetivos de la clase

Continuamos con el estudio de los Lenguajes Regulares, abordaremos propiedades importantes de los mismos:



Minimización de AFD

- *Intuitivamente se puede decir que un objetivo importante, en la teoría de autómatas finitos, es el de poder hallar para cualquier lenguaje regular, un autómata finito determinístico (AFD) con el menor número de estados posible, que además sea único.*
- ***Para ello veremos cómo minimizar AFD's.***

La idea es, entonces, que podamos tomar cualquier AFD y encontrar un AFD equivalente, con el menor número de estados posible.

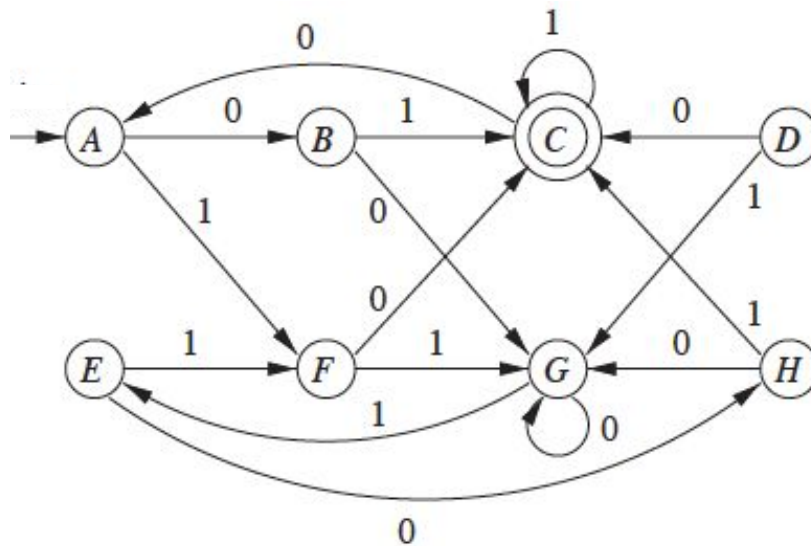
Algoritmo de Minimización

Dado un AFD $M = (Q, \Sigma, \delta, q_0, F)$:

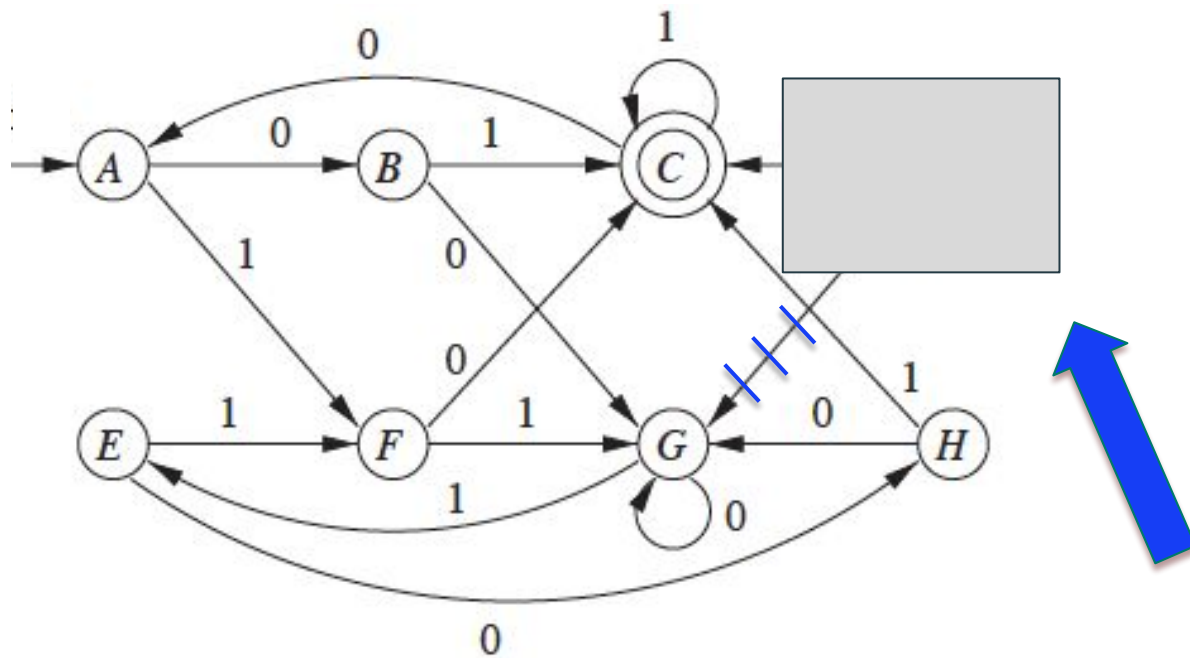
1. Eliminar todos los estados que no pueden ser alcanzados desde el estado de comienzo (estados inalcanzables).
2. Usar el algoritmo de tabla completa para encontrar todos los pares de estados equivalentes.
3. Particionar el conjunto de estados Q en bloques de estados mutuamente equivalentes.
4. Construir el AFD mínimo equivalente N , usando los bloques como sus estados.

Comenzamos a aplicar cada uno de los pasos del algoritmo

- Consideremos el siguiente AFD, para aplicar cada uno de los pasos del algoritmo de minimización:



1-Eliminar estados inaccesibles



2 -Testeo de estados equivalentes

El objetivo es entender cuando dos estados distintos, digamos p y q , se comportan de manera equivalente.

Decimos que los estados p y q son equivalentes si: **para toda cadena de entrada w , $\hat{\delta}(p, w)$ es un estado final sí y sólo si $\hat{\delta}(q, w)$ es un estado final.**

Notar que no es necesario que $\hat{\delta}(p, w)$ y $\hat{\delta}(q, w)$ sean el mismo estado, sólo que o ambas den como resultado estados finales o ambas sean estados no finales.

Si dos estados no son equivalentes, entonces decimos que son *distinguibles*. Es decir, el estado p es distinguible del estado q si hay al menos una cadena w tal que uno de $\hat{\delta}(p, w)$ y $\hat{\delta}(q, w)$ es final, y el otro es no final.

2 -Testeo de estados equivalentes

Algoritmo de Tabla Completa

El **algoritmo de tabla completa** , da un procedimiento recursivo para descubrir estados distinguibles en un AFD $M = (Q, \Sigma, \delta, q_0, F)$:

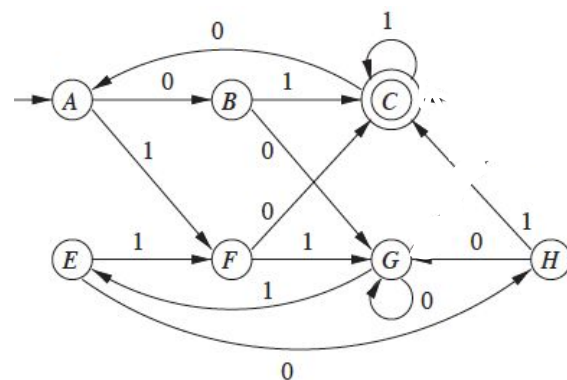
Base: si p es un estado final y q es un estado no final, entonces el par $\{p, q\}$ es distinguible.

Inducción: sean p y q estados tales que, para algún símbolo de entrada a , $\delta(p, a) = r$ y $\delta(q, a) = s$, si r y s son un par de estados distinguibles, entonces el par $\{p, q\}$ es un par de estados distinguibles, también.

Aplicando el Algoritmo de Tabla Completa

La idea es armar una tabla donde colocaremos una *X* para indicar que un par de estados es distinguible. Para aquellos pares de estados que no resulten distinguibles, se deja un *blanco* que indica que dichos estados son equivalentes. Trabajamos sobre el AFD dado previamente:

- *Inicialmente no hay X's en la tabla.*
- *Dado que C es el único estado final, entonces, ¿qué debemos hacer para empezar?*



Aplicando el Algoritmo de Tabla Completa (Cont.)

Paso Base: tomemos C y G y la cadena λ , dado que $\hat{\delta}(C, \lambda)$ es un estado final y $\hat{\delta}(G, \lambda)$ no lo es, entonces estos estados son distinguibles.

B						
C	X	X				
E			X			
F			X			
G			X			
H			X			
	A	B	C	E	F	G

Y así tenemos que proceder para marcar como distinguibles todos los estados finales con los no finales.

Aplicando el Algoritmo de Tabla Completa (Cont.)

Tomemos los estados A y G , la cadena λ no los distingue, la cadena 0 ?, la cadena 1 ? Y con la cadena 01 ?

Debido a que el resultado para ambas δ 's son estados no finales. La cadena 0 tampoco los distingue dado que ellos van a B y G , respectivamente y ambos estados son no finales. Lo mismo ocurre con la cadena 1 . Sin embargo, 01 los distingue.

B						
C	X	X				
E			X			
F			X			
G			X			
H			X			
	A	B	C	E	F	G

¿Por qué?

Tomemos los estados A y H , qué ocurre para el símbolo 1 ?

Aplicando el Algoritmo de Tabla Completa (Cont.)

Tomemos $\{C, G\}$ que son distinguibles, vemos que los estados B y F van a C y G para la entrada 1, entonces B y F son también distinguibles.

Dado que $\{C, H\}$ son distinguibles, y los estados E y F , para la entrada 0 tienen definida una transición a los estados H y C , respectivamente, entonces $\{E, F\}$ son también distinguibles.

Por ejemplo, el par $\{B, H\}$ será distinguible?.

De esta manera se procede con cada par de estados, cuando no podemos determinar que un par de estados es distinguible, entonces el mismo será equivalente.

Aplicando el Algoritmo de Tabla Completa (Cont.)

Terminar de aplicar el método y corroborar si la siguiente tabla completa es correcta:

B	X					
C	X	X				
E		X	X			
F	X	X	X	X		
G	X	X	X	X	X	
H	X		X	X	X	X
	A	B	C	E	F	G

3-Particionar el conjunto de estados

Teorema: si dos estados no son distinguibles por el algoritmo de tabla completa, entonces los estados son equivalentes.

Teorema: la equivalencia de estados es transitiva. Esto significa que, si en algún AFD $A = (Q, \Sigma, \delta, q_0, F)$ podemos encontrar que los estados p y q son equivalentes y también encontramos que q y r son equivalentes, entonces p y r son equivalentes.



Ver demostración en la bibliografía.

3-Particionar el conjunto de estados

El último teorema (de la página anterior) permite justificar el algoritmo de particionado de estados:

Para cada estado q , construimos un bloque que consiste de q y todos los estados que son equivalentes a q . Y se puede demostrar que, los bloques resultantes son una partición, o sea que, ningún estado puede estar en dos bloques distintos.

Para el ejemplo que venimos trabajando, ¿cuáles serían los bloques de estados?

4- Obtener el AFD mínimo

Construir el AFD mínimo equivalente N , a partir del AFD $M = (Q, \Sigma, \delta, q_0, F)$, usando los bloques como sus estados.

1. El estado de comienzo de N es el bloque que contiene el estado de comienzo de M .
2. El conjunto de estados finales de N es el conjunto de bloques que contienen estados finales de M .
3. La función de transición de N , a la cual llamaremos γ , se define de la siguiente manera:

Supongamos S un conjunto de estados equivalentes de M , sea $a \in \Sigma$, entonces debe existir un bloque T de estados tal que $\forall q \in S, \delta(q, a)$ es un miembro del bloque T . Como consecuencia podemos escribir $\gamma(S, a) = T$.

Continuando con el ejemplo

B	X						
C	X	X					
E		X	X				
F	X	X	X	X			
G	X	X	X	X	X		
H	X		X	X	X	X	
	A	B	C	E	F	G	

En primer lugar debemos determinar cuáles son los bloques de estados equivalentes, para ello debemos inspeccionar la tabla que nos dió los estados equivalentes y los distinguibles.

De aquí y aplicando el método de estados mutuamente equivalentes obtenemos los bloques $([A, E], [B, H], [C], [F], [G])$.

El estado de comienzo es $[A, E]$, dado que A era el estado inicial del autómata original.

El único estado final es $[C]$ dado que C era el único estado final de M .

Continuando con el ejemplo

Ahora veamos como obtenemos la función de transición γ , por ejemplo:

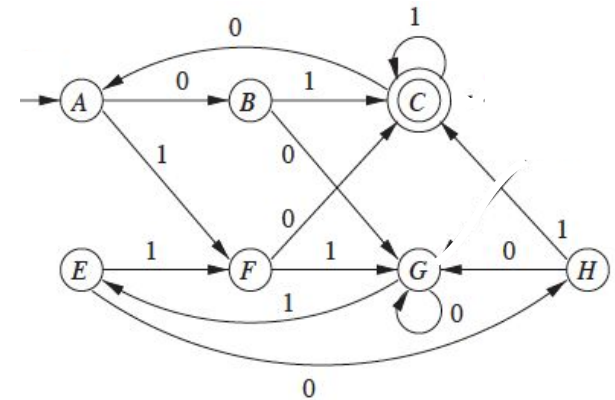
$$\gamma([A, E], 0) = [B, H] \quad \gamma([B, H], 0) = [G]$$

$$\gamma([A, E], 1) = [F] \quad \gamma([B, H], 1) = [C]$$

$$\gamma([C], 0) = [A, E] \quad \gamma([F], 0) = [C]$$

$$\gamma([C], 1) = [C] \quad \gamma([F], 1) = [G]$$

$$\gamma([G], 0) = [G] \quad \gamma([G], 1) = [A, E]$$



Continuando con el ejemplo

Obtenga el diagrama de transición del AFD mínimo construido.

Propiedades de Clausura

A continuación se listan algunas de las propiedades más importantes:

- ❖ La **unión** de dos lenguajes regulares es regular.
- ❖ La **intersección** de dos lenguajes regulares es regular.
- ❖ El **complemento** de un lenguaje regular es regular.
- ❖ La **diferencia** de dos lenguajes regulares es regular.
- ❖ El **reverso** de un lenguaje regular es regular.
- ❖ La **clausura de Kleene** de un lenguaje regular es regular.
- ❖ La **concatenación** de lenguajes regulares es regular.

Propiedades de Clausura

Unión

Teorema: Si L_1 y L_2 son lenguajes regulares, entonces también lo es $L_1 \cup L_2$.

Demostración: Dado que L_1 y L_2 son regulares, ellos son denotados por alguna expresión regular, digamos $L_1 = L(R)$ y $L_2 = L(S)$. Entonces $L_1 \cup L_2 = L(R + S)$, por definición del operador $+$ para expresiones regulares.

Propiedades de Clausura

Complemento

Teorema: Si L es un lenguaje regular sobre el alfabeto Σ , entonces $\bar{L} = \Sigma^* - L$ también es un lenguajes regular.

Demostración: Sea $L = L(A)$ para algún AFD

$A = (Q, \Sigma, \delta, q_0, F)$. Entonces $\bar{L} = L(B)$, donde B es el AFD $B = (Q, \Sigma, \delta, q_0, Q - F)$. O sea que, B es exactamente como A , pero los estados finales de A son estados no finales de B y viceversa. Entonces w está en $L(B)$ sí y sólo sí $\hat{\delta}(q_0, w)$ está en $Q - F$, lo cual ocurre sí y sólo sí no está en $L(A)$.

Aplicación propiedades de clausura

Dada una ER, ¿Podremos hallar el complemento de dicha ER?

Piense y escriba una solución al problema planteado.

¿DUDAS?

