Propiedades de Decisión de los Lenguajes Regulares

Alan Reyes-Figueroa
Teoría de la Computación **(Aula 09a) 16.agosto.2023**

Discusión general de "Propiedades"
Algoritmos para
Pertenencia, Vacuidad, Finitud, Etc.

Propiedad: Equivalencia

- Dados lenguajes regulares L y M, cómo verificar si ¿L = M?
- Hay un algoritmo que envuelve la construcción del *producto de dos DFA* a partir de los DFA para L y M.
- Estos DFA's tienen conjuntos de estados Q y R, respectivamente.
- El DFA producto posee estados $Q \times R$.
  - res [q, r] con $q \in Q$, $r \in R$.

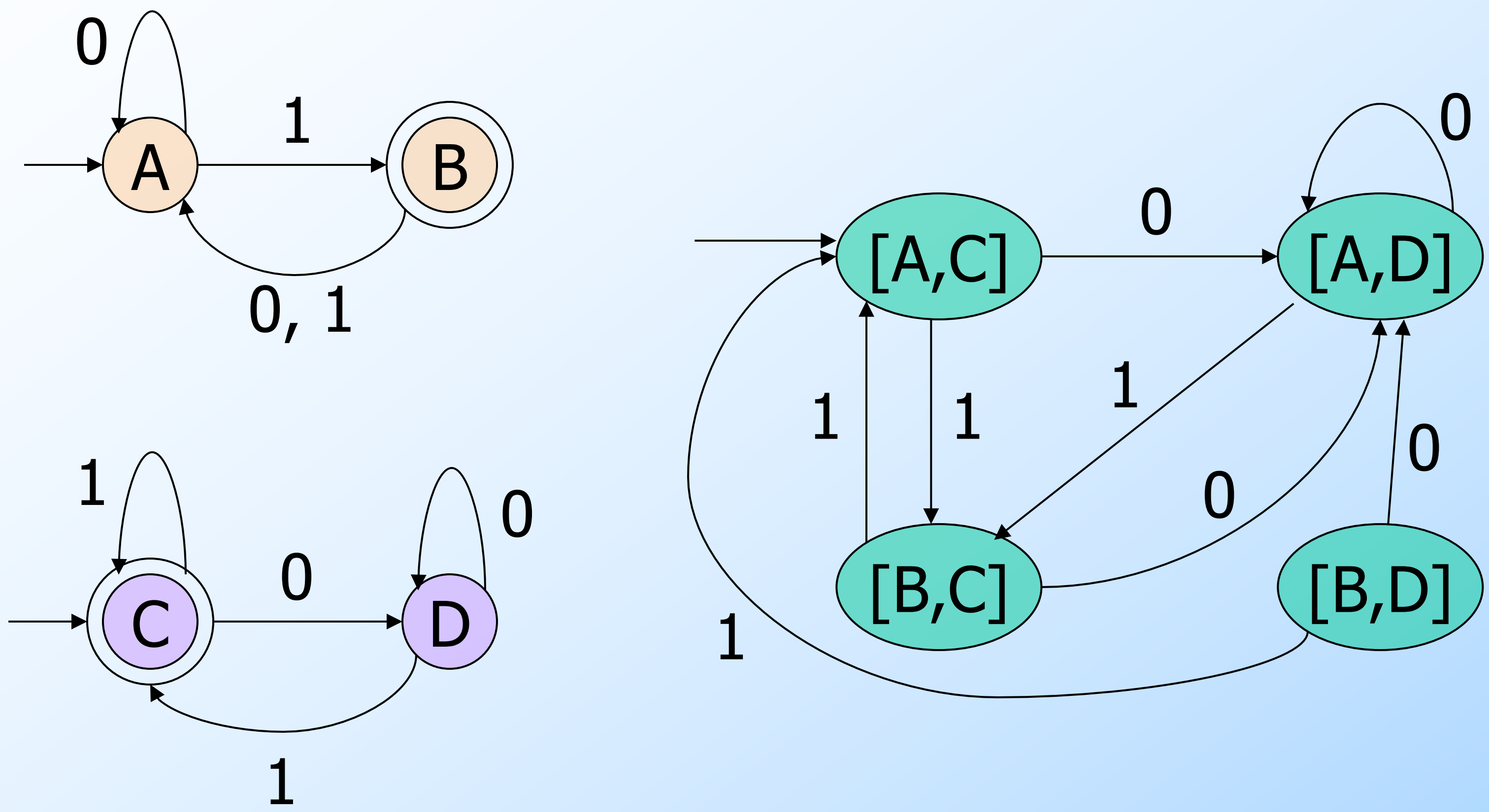
DFA Producto

$A_L = (Q,\Sigma,q_0,F_L,\delta_L)$, $A_M = (R,\Sigma,r_0,F_M,\delta_M)$

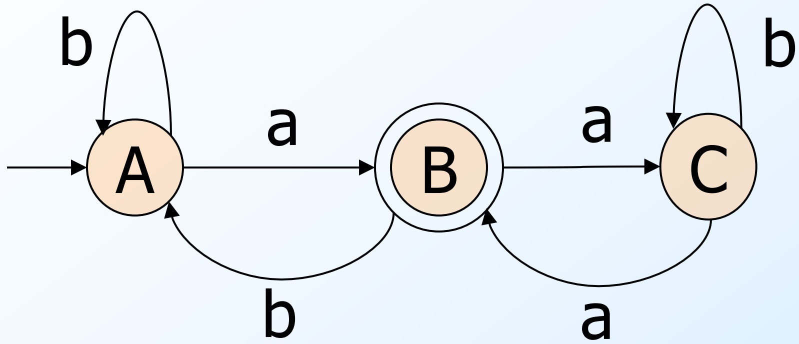
Construimos el producto:

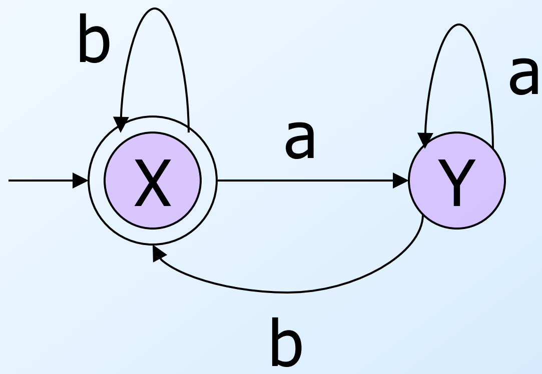
- Estado inicial = $[q_0,r_0]$
- Transiciones:
  $\delta([q,r], a) = [\delta_L(q,a), \delta_M(r,a)]$
  - δ_L, δ_M son las funciones de transición de los autómatas de L y M, resp.
  - Básicamente, simulamos el producto en moviéndonos en dos componentes.

Ejemplo: DFA Producto



Ejemplo: DFA Producto





Propiedades: Cerradura

- Dados lenguajes regulares L y M, y sus autómatas AFD A_L y A_M, respectivamente, podemos construir autómatas para
  - La unión $L \cup M$
  - La intersección $L \cap M$
  - Las diferencias $L - M$ y $M - L$
  - La diferencia simétrica $L \oplus M$
  - El complemento L^c
- Todas usan el autómata producto.
  (Sólo cambia la forma en cómo se definen los estados de aceptación).

Propiedades: Cerradura

- Unión $L \cup M$:
  [q,r] es estado de aceptación si q o r (o ambos) son estados de aceptación.

- Intersección $L \cap M$:
  [q,r] es estado de aceptación si q y r (ambos) son estados de aceptación.

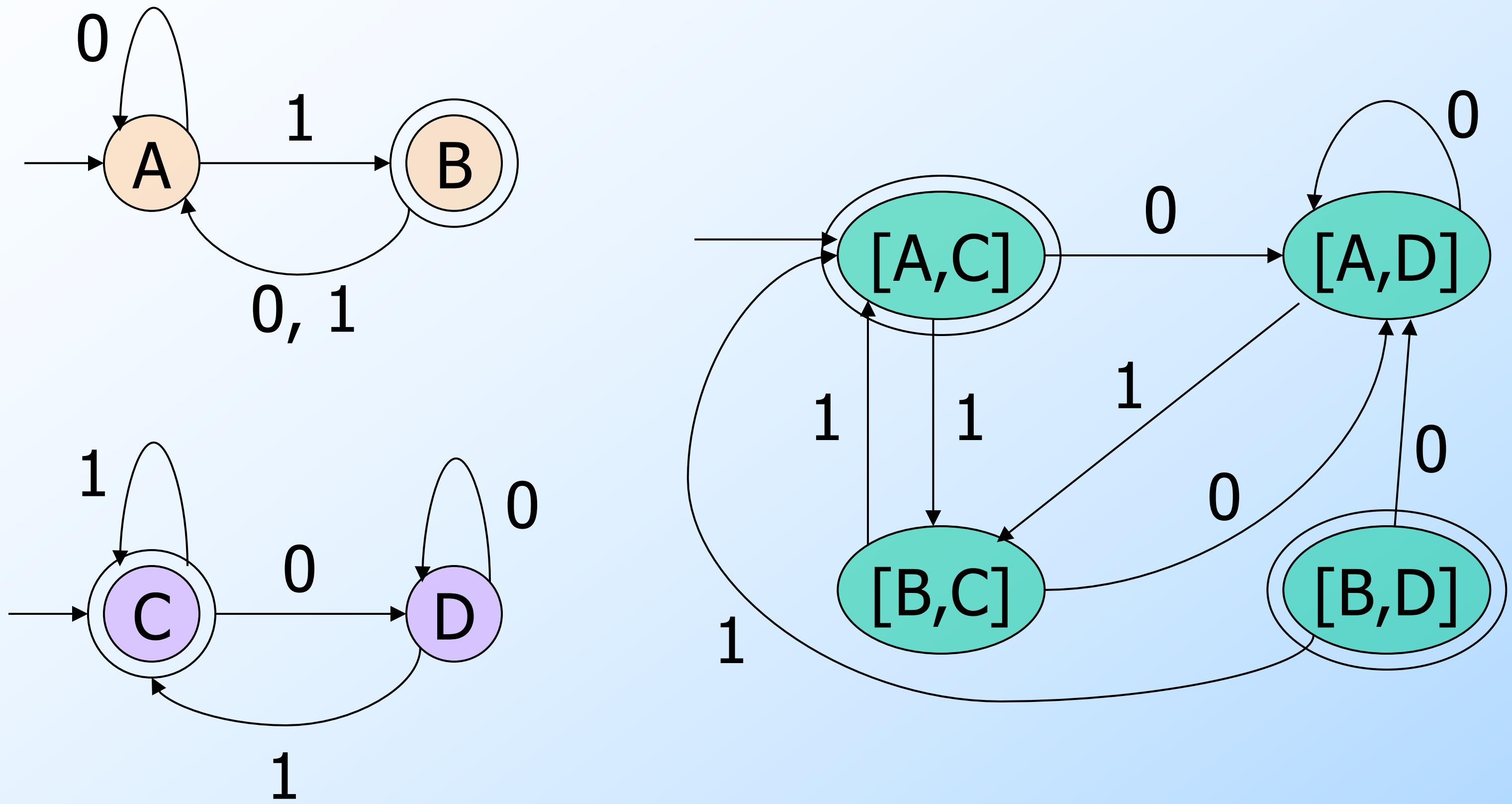
- Diferencia $L - M$:
  [q,r] es estado de aceptación si q es de aceptación, y r no lo es.

- Diferencia Simétrica $L \oplus M$:
  [q,r] es estado de aceptación si q o r son estados de aceptación, pero no ambos.

Algoritmo de Equivalencia

- Los estados finales del DFA producto corresponden a aquellos pares [q, r] tales que exactamente uno de q ó r son estados finales de su respectivo DFA (pero no ambos).
- Así, el autómata producto acepta w si, y sólo si, w está en exactamento uno de los lenguajes L ó M (pero no ambos).

Ejemplo: Equivalencia



Algoritmo de Equivalencia

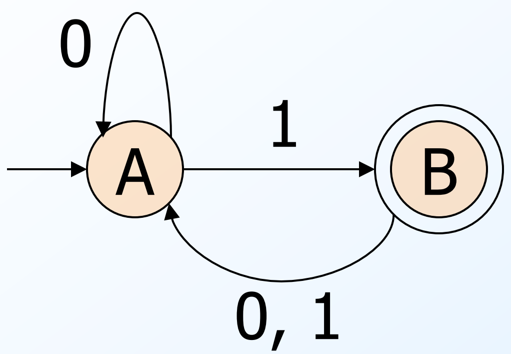
- El lenguaje asociado al DFA producto es vacío si, y sólo si, L = M.
- Si recordamos, ya tenemos un algoritmo para evaluar si el lenguaje generado por un DFA es vacío.

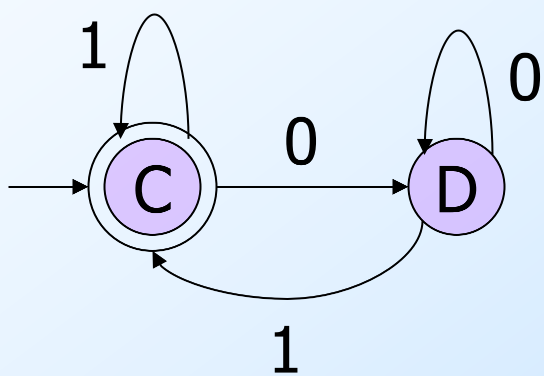
Propiedad: Inclusión

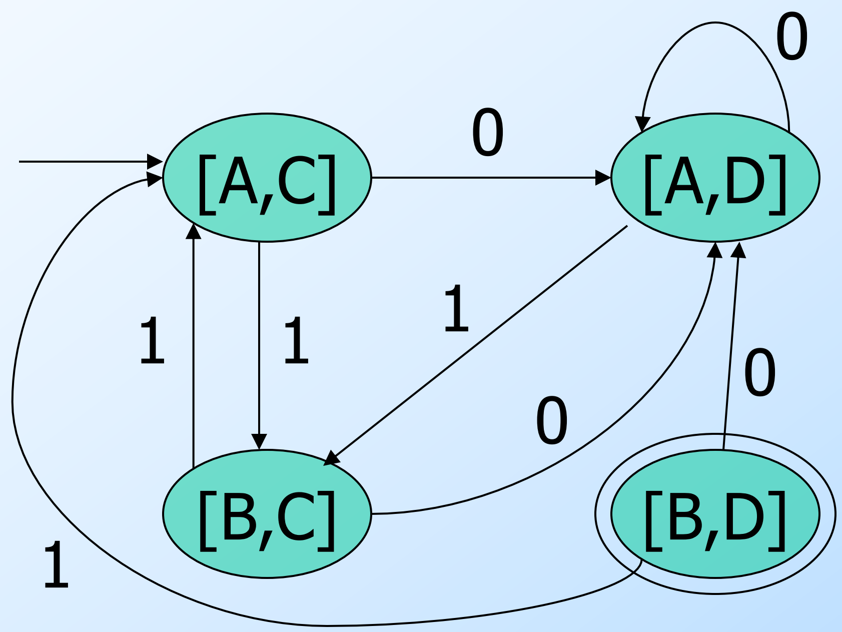
- Dados lenguajes regulares L y M, está $L \subseteq M$?
- Tenemos un algoritmo para esto, el cual también usa el DFA producto.
- ¿Cómo definiría los estados finales [q, r] del producto para que el lenguaje acetado sea vacío, si y sólo si, $L \subseteq M$?

Respuesta: q es final; r no lo es.

Ejemplo: Inclusión







Nota: el único estado final es inaccesible, de modo que el lenguaje es vacío, y L $\subseteq$ M.

DFA minimal

- En principio, ya que podemos verificar la equivalencia de DFAs, dado un autómata A podemos hallar el DFA con la menor cantidad de estados, que acepta el lenguaje L(A).
- 1a Solución: Testar todos los DFAs menores a ver si son equivalentes con *A*.
- Es un muy mal algoritmo.

Minimización Eficiente de estados

- Construir una tabla con todos los pares de estados.
- Si hallamos una cadena *distinguida,* esto es dos estados (que torna exactamente uno de ellos en un estado de aceptación), marcamos ese par.
- El algoritmo es una recursión sobre la longitud de la menor cadena distinguida.

Minimización de estados

- Eliminamos inaccesibles desde q_0.
- Base: Marcamos [q, r] si exactamente uno de ellos es un estado final.
- Inducción: Marcamos [q, r] si existe algún símbolo *a* tal que [δ(q,a), δ(r,a)] está marcado.
- Cuando ya no hay más marcas posibles, los pares no marcados son equivalentes y pueden fusionarse en un solo estado.

Transitividad de “Indistinguibles”

- Si el estado p es indistinguible de q, y q es indistinguible de r, entonces p es indistinguible de r.
- Prueba: La salida (aceptar o no aceptar) de p y q en la entrada w es la misma, y la salida de q y r en la entrada w es la misma, así que es la misma salida para p y r.

Construcción del DFA Minimal

- Suponga $q_1,\ldots,q_k$ son estados indistinguibles.
- Los reemplazamos por un estado q.
- Luego, $\delta(q_1,a),\ldots, \delta(q_k,a)$ son todos estados indistinguibles.
  - Punto clave: caso contrario, deberíamos haber marcado al menos un par más.
- Sea $\delta(q, a)$ = es estado representativo de ese grupo.

Ejemplo: Minimización

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}	{2,4,6,8}	{1,3,7,9}
{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
{1,3,5,7}	{2,4,6,8}	{1,3,5,7,9}
* {1,3,7,9}	{2,4,6,8}	{5}
* {1,3,5,7,9}	{2,4,6,8}	{1,3,5,7,9}

	r	b
→ A	B	C
B	D	E
C	D	F
D	D	G
E	D	G
* F	D	C
* G	D	G

La misma tabla pero con labels más simples.

Recuerdan esta autómata? Fue el que construimos como ejemplo del tablero en la construcción de un AFN a un AFD.

Ejemplo: Minimización

		r	b
→	A	B	C
	B	D	E
	C	D	F
	D	D	G
	E	D	G
*	F	D	C
*	G	D	G

	G	F	E	D	C	B
A	X	X				
B	X	X				
C	X	X				
D	X	X				
E	X	X				
F						

Comenzar con marcas para los pares con uno de los estados finales F ó G.

Ejemplo: Minimización

	r	b
→ A	B	C
B	D	E
C	D	F
D	D	G
E	D	G
* F	D	C
* G	D	G

	G	F	E	D	C	B
A	X	X				
B	X	X				
C	X	X				
D	X	X				
E	X	X				
F						

El input *r* no ayuda, ya que el par [B, D] no está marcado.

Ejemplo: Minimización

		r	b
→	A	B	C
	B	D	E
	C	D	F
	D	D	G
	E	D	G
*	F	D	C
*	G	D	G

	G	F	E	D	C	B
A	x	x	x	x	x	
B	x	x	x	x	x	
C	x	x				
D	x	x				
E	x	x				
F	x					

Pero el input *b* distingue a {A,B,F} de {C,D,E,G}. Por ejemplo, [A, C] es marcado ya que [C, F] lo es.

Ejemplo: Minimización

	r	b
→ A	B	C
B	D	E
C	D	F
D	D	G
E	D	G
* F	D	C
* G	D	G

	G	F	E	D	C	B
A	X	X	X	X	X	
B	X	X	X	X	X	
C	X	X	X	X		
D	X	X				
E	X	X				
F	X					

[C, D] y [C, E] son marcados ya que las transiciones en b al par marcado [F, G].

Ejemplo: Minimización

		r	b
→	A	B	C
	B	D	E
	C	D	F
	D	D	G
	E	D	G
*	F	D	C
*	G	D	G

	G	F	E	D	C	B
A	X	X	X	X	X	X
B	X	X	X	X	X	
C	X	X	X	X		
D	X	X				
E	X	X				
F	X					

[A, B] es marcado
ya que hay transiciones r
al par marcado [B, D].

[D, E] nunca es marcado,
ya que ambos imputs D, E
van al mismo estado.

Ejemplo: Minimización

		r	b
→	A	B	C
	B	D	E
	C	D	F
	D	D	G
	E	D	G
*	F	D	C
*	G	D	G

		r	b
→	A	B	C
	B	H	H
	C	H	F
	H	H	G
*	F	H	C
*	G	H	G

	G	F	E	D	C	B
A	x	x	x	x	x	x
B	x	x	x	x	x	
C	x	x	x	x		
D	x	x				
E	x	x				
F	x					

Reemplazamos D y E por H.
El resultado es el DFA mínimo.

Eliminando estados no alcanzables

- Desafortunadamente, al combinar estados indistinguibles podríamos resultar con estados no alcanzables en el DFA "mínimo".
- Entonces, tarde o terprano, debemos remover aquellos estados que no son alcanzables desde el estado inicial.

Punto clave

- Hemos combinado estados del autómata DFA siempre que es posible.
- Pregunta: ¿Puede existir otro DFA, completamente distinto, con menor número de estados?
- Respuesta: **No.**
  La prueba envuelve minimizar el DFA derivado del hipotético mejor DFA.

Prueba: DFA Minimal

- Sea A nuestro DFA minimizado, y sea B un menor DFA equivalente.
- Consideramos un autómata con los estados de A y B combinados.
- Usamos "distinguible" en su forma contrapositiva:
  - Si los estados p y q son indistinguibles, entonces también lo son $\delta(q,a)$ y $\delta(p,a)$.

Indistinguibilidad

$$q_0 \xrightarrow{a} q \xrightarrow{b} r$$

Estados iniciales de A y B indistinguibles ya que L(A)= L(B).

Deben ser indistinguibles

Deben ser indistinguibles

$$p_0 \xrightarrow{a} p \xrightarrow{b} s$$

Hipótesis Inductiva

- Todo estado q de A es indistinguible de cualquier estado de B.
- La inducción se hace sobre la longitud n de la menor cadena que lleva desde el estado inicial de A hacia q.

Prueba – (2)

- Base: Los estados iniciales de A y B son indistinguibles, ya que L(A) = L(B).
- Inducción: Suponer que w = xa es una menor cadena desde A al estado q.
- Por hipótesis, x lleva A a un estado r que es indistinguible de algún estado p de B.
- Entonces $\delta(r, a) = q$ es indistinguible de $\delta(p, a)$.

Prueba – (3)

- Sin embargo, dos estados de A no pueden ser distinguidos desde el mismo estado de B, o en caso contrario, ellos serían distinguibles uno desde el otro.
  - Esto contradice la transitividad de los "indistinguibles."
- Así, B posee al menos el mismo número de estados que A.