

Propiedades de Cerradura de los Lenguajes Regulares

Alan Reyes-Figueroa

Teoría de la Computación

(Aula 08) 06.agosto.2025

Unión, Intersección, Diferencia,
Concatenación, Cerradura de,
Lenguaje Reverso, Homomorfismo
y Homomorfismo Inverso

Propiedades de Cerradura

- Recordemos que una propiedad de cierre es un enunciado sobre cierta operación de lenguajes, que cuando se aplica a una clase de lenguajes L (por ejemplo, los lenguajes regulares), produce un resultado que también está en esa clase L .
- Para lenguajes regulares, unamos cualquiera de sus representaciones para mostrar una propiedad de cerradura.

Cerradura bajo la unión

- Si L y M son lenguajes regulares, también lo es $L \cup M$.
- **Prueba:** Sean L y M lenguajes representados por las expresiones regulares R y S , respectivamente.
- Entonces, $R+S$ es una expresión regular cuyo lenguaje es $L \cup M$.
 - Recordar el automata de la unión (la hamburguesa).

Cerradura de la Concatenación y de Kleene

- **Prueba:** La misma idea del caso anterior: si L y M están representados por las expresiones regulares R y S , respectivamente.
- RS es una expresión regular para LM .
 - Recordar el autómata (cadena) para la regexp RS .
- R^* es una expresión regular para L^* .
 - Recordar el autómata (el barquito) para la regexp R^* .

Producto de Autómatas

Sean $M_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$

$M_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$

autómatas AFD.

Definimos un nuevo autómata AFD, llamado el *autómata producto* de M_1 y M_2 como el autómata con la estructura

$$M_1 \times M_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_1 \times \delta_2, [q_{01}, q_{02}], G)$$

Producto de Autómatas

Donde

- $Q = Q_1 \times Q_2$ es el producto cartesiano de los conjuntos de estados de los autómatas M_1 y M_2 .
- $\Sigma = \Sigma_1 \cup \Sigma_2$ es la unión de todos los símbolos usados en M_1 y M_2 .
- $q_0 = [q_{01}, q_{02}]$ es el estado inicial (es el par formado por los estados iniciales de M_1 y M_2).
- $\delta = \delta_1 \times \delta_2$ es la función producto:
$$(\delta_1 \times \delta_2)[a, b] = [\delta_1(a), \delta_2(b)]$$

Producto de Autómatas

Los estados ahora son de la forma:

$$q = [q_1, q_2], \text{ con } q_1 \in Q_1, q_2 \in Q_2$$

□ Las transiciones ahora son de la forma

$$\delta(q_1, q_2) = (\delta_1 \times \delta_2)[q_1, q_2] = [\delta_1(q_1), \delta_2(q_2)]$$

(ambas funciones actúan sobre el mismo alfabeto.)

□ Los estados de aceptación G de $M_1 \times M_2$ se pueden definir de diferentes formas, según la utilidad del autómata producto.

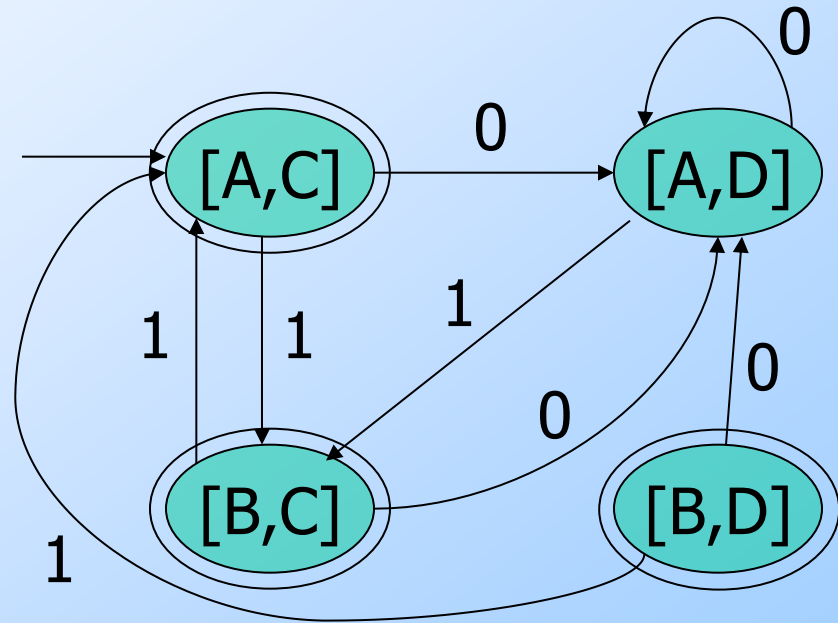
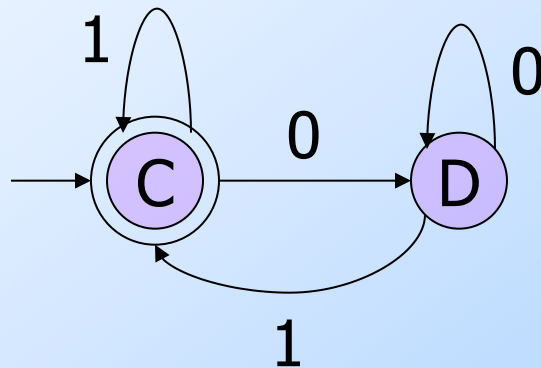
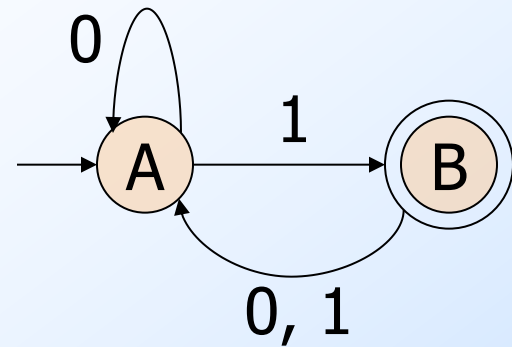
Cerradura de la Unión

Propiedad: Si L y M son lenguajes regulares, entonces también lo es $L \cup M$.

Prueba: Sean A y B dos autómatas AFD cuyos lenguajes son L y M , respectivamente.

- Construimos $C = A \times B$, el autómata producto de A y B .
- Hacemos los estados finales de C , aquellos pares $[q, r]$, donde q es estado final de A o r es estado final de B (al menos uno del par es de aceptación).

Ejemplo: AFD para la unión



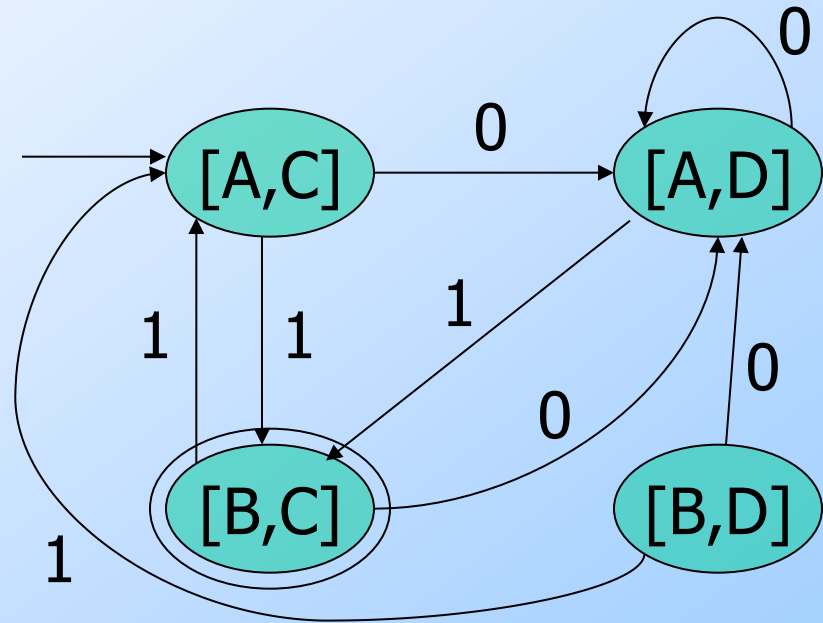
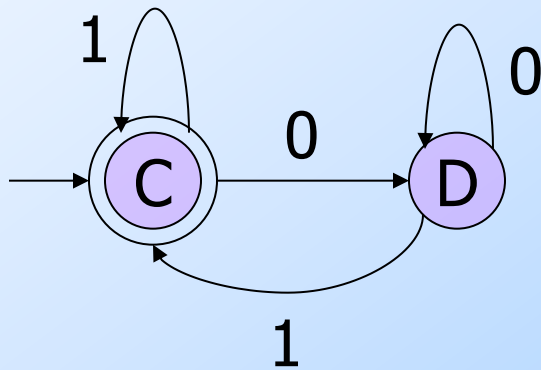
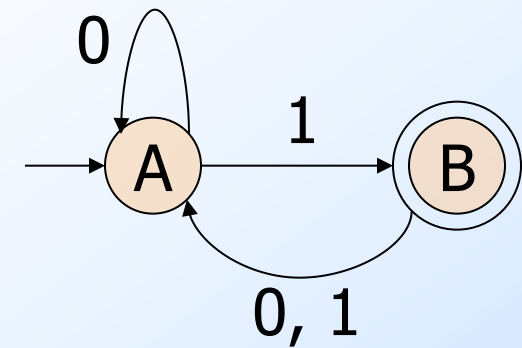
Cerradura de la Intersección

Propiedad: Si L y M son lenguajes regulares, entonces también lo es $L \cap M$.

Prueba: Sean A y B dos autómatas AFD cuyos lenguajes son L y M , respectivamente.

- Construimos $C = A \times B$, el autómata producto de A y B .
- Hacemos los estados finales de C , aquellos pares $[q, r]$, donde q es estado final de A y r es estado final de B (ambos elementos del par son estados de aceptación).

Ejemplo: AFD para intersección



Cerradura bajo la Diferencia

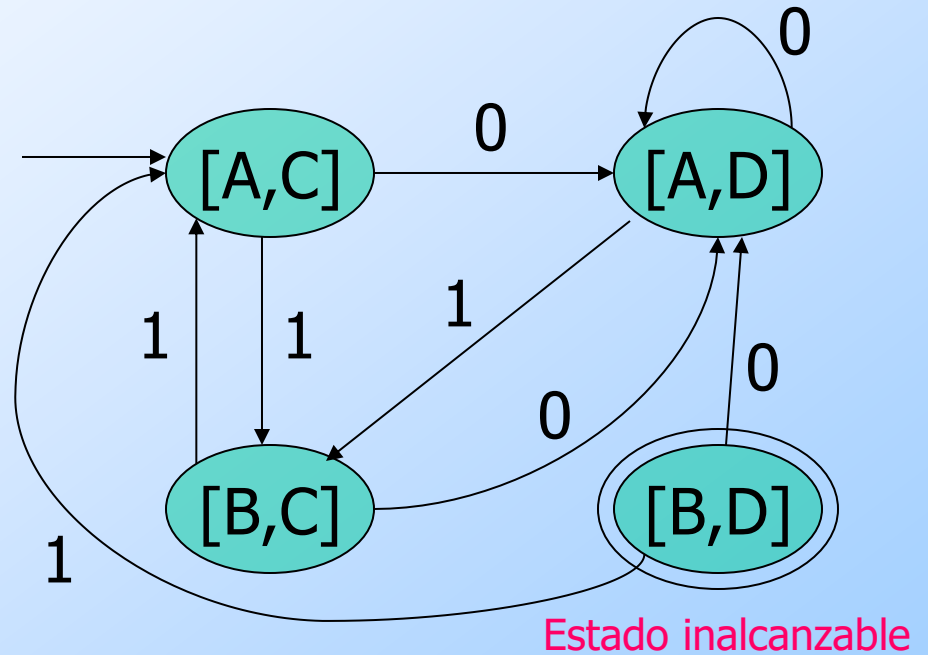
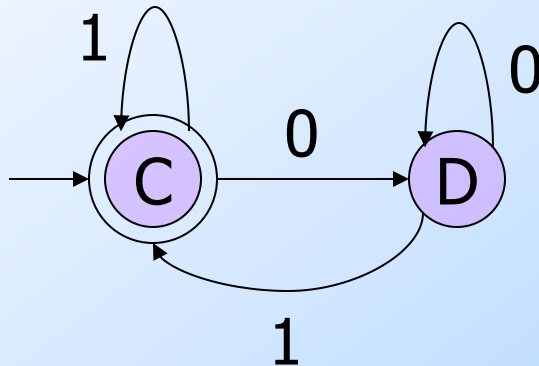
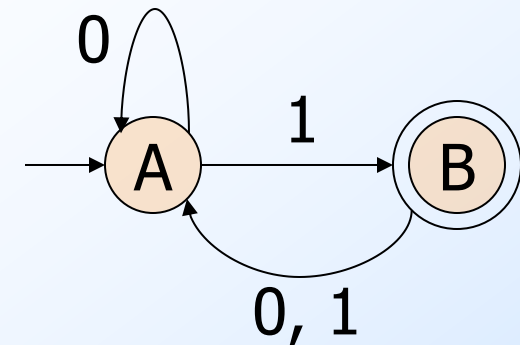
Propiedad: Si L y M son lenguajes regulares, entonces también lo es $L - M$.

(queremos las cadenas en L , pero no en M).

Prueba: Sean A y B autómatas AFD cuyos lenguajes son L y M , respectivamente.

- Construimos $C = A \times B$, el autómata producto.
- Hacemos los estados finales de C , aquellos pares $[q, r]$, donde q es estado final de A , pero r no es estado final de B (el primer element del par es de aceptación y el segundo no lo es).

Ejemplo: AFD para diferencia



Nota: observe en este ejemplo que la dif. es el lenguaje vacío.

Cerradura bajo Complemento

- El *complemento* de un lenguaje L (con respecto al alfabeto Σ , con Σ^* conteniendo L) es $\Sigma^* - L$.
- Como Σ^* es ciertamente un lenguaje regular, el complemento de un lenguaje regular L es también un lenguaje regular.

Cerradura bajo Reversa

- Dado un lenguaje L , L^R es el conjunto de todas las cadenas cuya cadena reversa está en L .
- **Ejemplo:** $L = \{0, 01, 100\}$;
 $L^R = \{0, 10, 001\}$.
- Sea E una expresión regular para L .
- Mostraremos cómo revertir E , y producir una expresión regular E^R para L^R .

Reversa de una Regexp

- **Base:** Si E es un símbolo a , ϵ , ó $\emptyset$, entonces $E^R = E$.
- **Inducción:** Si E es
 - $F + G$, entonces $E^R = F^R + G^R$.
 - FG , entonces $E^R = G^R F^R$
 - F^* , entonces $E^R = (F^R)^*$.

Ejemplo: Reversa de *regexp*

- Sea $E = \mathbf{01^* + 10^*}$.
- $E^R = (\mathbf{01^* + 10^*})^R = (\mathbf{01^*})^R + (\mathbf{10^*})^R$
- $= (\mathbf{1^*})^R \mathbf{0^R} + (\mathbf{0^*})^R \mathbf{1^R}$
- $= (\mathbf{1^R})^* \mathbf{0} + (\mathbf{0^R})^* \mathbf{1}$
- $= \mathbf{1^*0 + 0^*1}$.

Homomorfismos

- Un *homomorfismo* sobre un alfabeto Σ es una función que asigna una cadena a cada símbolo del alfabeto.

$$h: \Sigma \rightarrow \Sigma^*$$

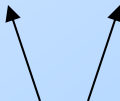
- Ejemplo: $h(0) = ab$; $h(1) = t$.
- Extendemos h a cadenas mediante
$$h(a_1 \dots a_n) = h(a_1) \dots h(a_n).$$
- Ejemplo: $h(01010) = abtabtab$.

Cerradura bajo homomorfismos

- Si L es un lenguaje regular, y h es un homomorfismo sobre su alfabeto Σ , entonces $h(L) = \{h(w) : w \in L\}$ es también un lenguaje regular.
- **Prueba:** Sea E una expresión regular de L .
- Aplicar h a cada símbolo en E .
- El lenguaje de la regexp resultante es $h(L)$.

Ejemplo: Homomorfismos

- Considere $h(0) = ab$; $h(1) = \epsilon$.
- Sea L es el lenguaje generado por la expresión regular **$01^* + 10^*$** .
- Entonces $h(L)$ es el lenguaje generado por la expresión **$ab\epsilon^* + \epsilon(ab)^*$** .

 **Note:** usamos paréntesis para clarificar el agrupamiento.

Ejemplo: Homomorfismos

- $\mathbf{ab}\epsilon^* + \epsilon(\mathbf{ab})^*$ puede simplificarse.
- $\epsilon^* = \epsilon$, así que $\mathbf{ab}\epsilon^* = \mathbf{ab}\epsilon$.
- ϵ es la identidad bajo la concatenación.
 - Esto es, $\epsilon E = E\epsilon = E$.
- Luego, $\mathbf{ab}\epsilon^* + \epsilon(\mathbf{ab})^* = \mathbf{ab}\epsilon + \epsilon(\mathbf{ab})^* = \mathbf{ab} + (\mathbf{ab})^*.$
- Finalmente, $L(\mathbf{ab})$ está contenido en $L((\mathbf{ab})^*)$, y la *regexp* de $h(L)$ es $(\mathbf{ab})^*$.

Homomorfismo inverso

- Sea h un homomorfismo y L un lenguaje cuyo alfabeto Σ es el lenguaje *output* obtenido de aplicar h .
- Definimos el homomorfismo inverso por
$$h^{-1}(L) = \{w: h(w) \in L\}.$$

Ejemplo: Homomorfismo inverso

- Tome $h(0) = ab$; $h(1) = \epsilon$.
- Consideremos $L = \{abab, baba\}$.
- $h^{-1}(L) =$ el lenguaje con dos 0's y cualquier cantidad de 1's
 $= L(\mathbf{1^*01^*01^*})$.

Nota: ninguna cadena se mapea en baba; cualquier cadena con dos 0's va a abab.