Propiedades de Cerradura de los Lenguajes Regulares

Alan Reyes-Figueroa

Teoría de la Computación **(Aula 09b) 16.agosto.2023**

Unión, Intersectión, 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 propiedade decerradura.

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 la expresiones regulares R y S, respectivamente.
- Entonces, R+S es una expresión regular cuyo lenguaje es $L \cup M$.

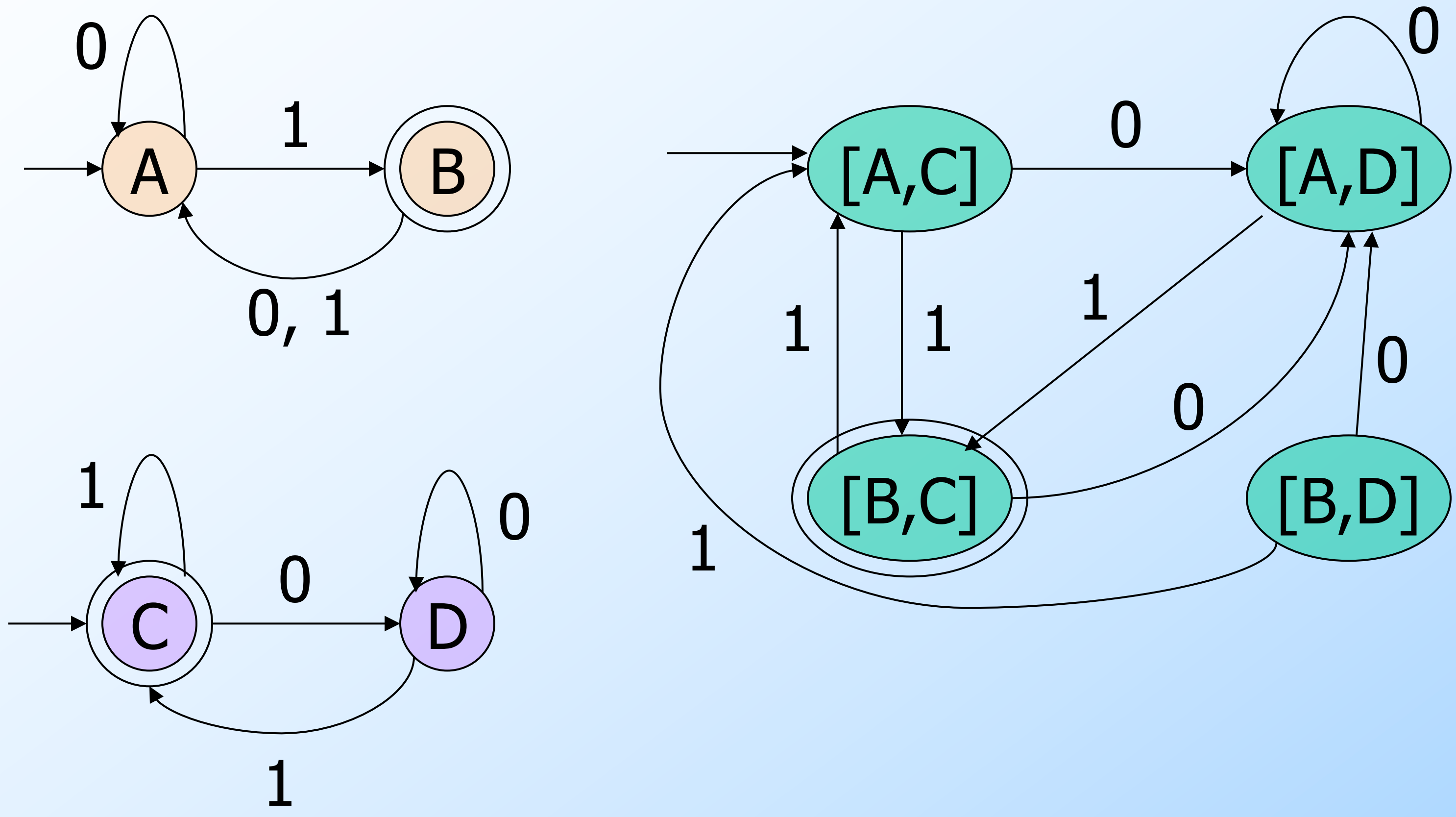
Cerradura de la Concatenación y de Kleene

- Prueba: La misma idea:
  - RS es una expresión regular cuyo lenguaje es LM.
  - R* es una expresión regular cuyo lenguaje es L*.

Cerradura de la Intersección

- 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, resp.
- Construímos $C = A \times B$, 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.

Ejemplo: DFA para intersección

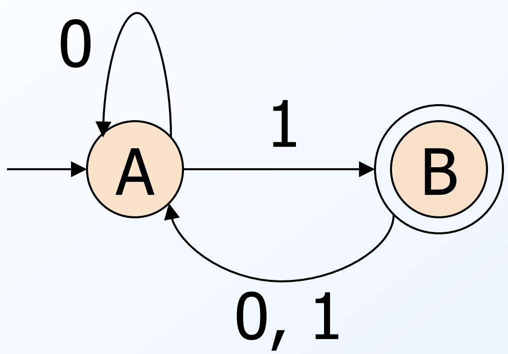


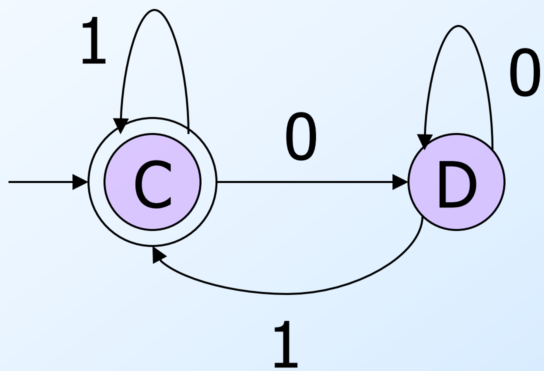
Cerradura bajo la Diferencia

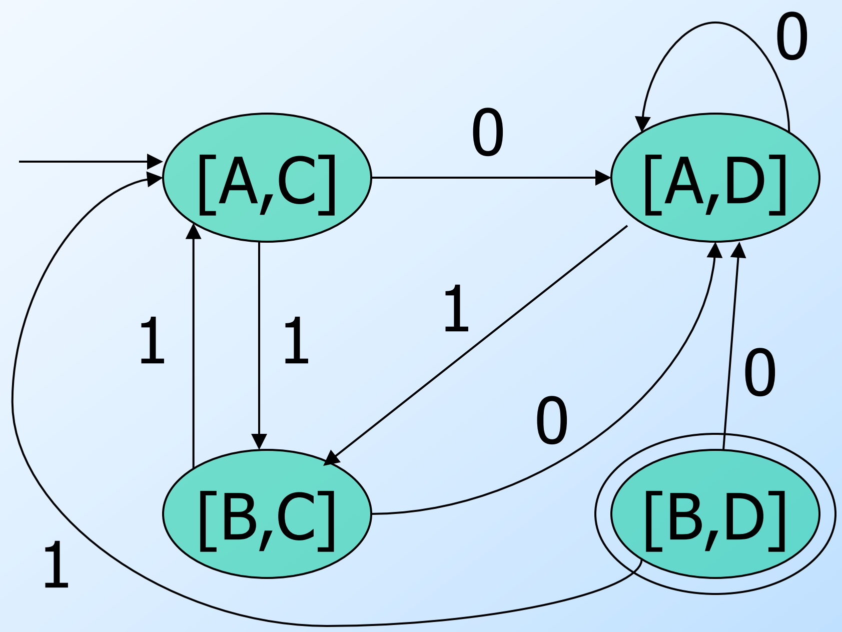
- Si L y M son lenguajes regulares, entonces también lo es $L - M$.

  (las cadenas en L, pero no en M).
- Prueba: Sean A y B autómatas AFD cuyos lenguajes son L y M, resp.
- Construímos C=A×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.

Ejemplo: DFA para diferencia







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

Cerradura bajo Complemento

- El *complemento* de un lenguaje L (con respecto al alfabeto Σ, con Σ* conteniendo L) es Σ* – 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 espresión 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 = $\mathbf{01}^* + \mathbf{10}^*$.
- $E^R = (\mathbf{01}^* + \mathbf{10}^*)^R = (\mathbf{01}^*)^R + (\mathbf{10}^*)^R$
- $\quad = (\mathbf{1}^*)^R\mathbf{0}^R + (\mathbf{0}^*)^R\mathbf{1}^R$
- $\quad = (\mathbf{1}^R)^*\mathbf{0} + (\mathbf{0}^R)^*\mathbf{1}$
- $\quad = \mathbf{1}^*\mathbf{0} + \mathbf{0}^*\mathbf{1}$.

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

$$h(a_1 \ldots a_n) = h(a_1) \ldots h(a_n).$$

- 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 $\mathbf{01}^* + \mathbf{10}^*$.
- Entonces h(L) es el lenguaje generado por la expresión $\mathbf{ab}\epsilon^* + \epsilon(\mathbf{ab})^*$.

Note: usamos paréntesis para clarificar el agrupamiento.

Ejemplo: Homomorfismos

- **ab**ϵ* + ϵ(**ab**)* puede simplificarse.
- ϵ* = ϵ, así que **ab**ϵ* = **ab**ϵ.
- ϵ es la identidad bajo la concatenación.
  - Esto es, ϵE = Eϵ = E.
- Luego, **ab**ϵ* + ϵ(**ab**)* = **ab**ϵ + ϵ(**ab**)*
  = **ab** + (**ab**)*.

- Finalmente, L(**ab**) está contenido en L((**ab**)*), y la *regex* de h(L) es (**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(**1***01***01***).

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

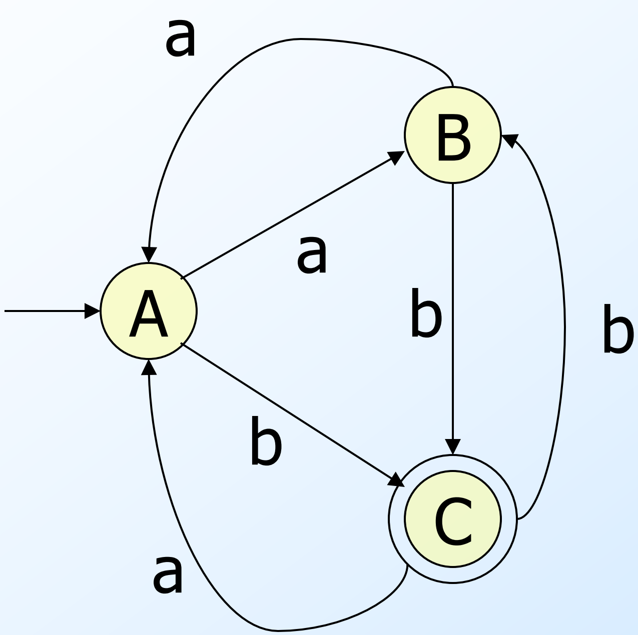
Closure Proof for Inverse Homomorphism

- Start with a DFA A for L.
- Construct a DFA B for $h^{-1}(L)$ with:
  - The same set of states.
  - The same start state.
  - The same final states.
  - Input alphabet = the symbols to which homomorphism h applies.

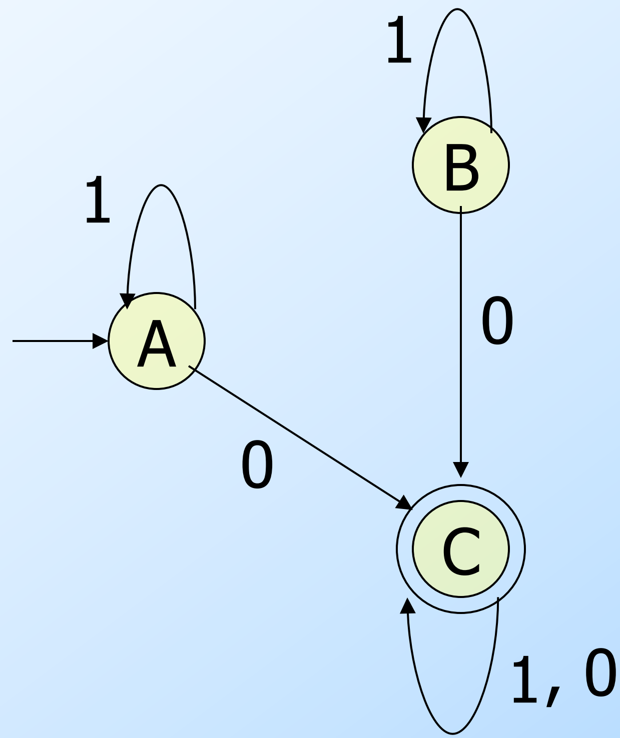
Proof – (2)

- The transitions for B are computed by applying h to an input symbol *a* and seeing where A would go on sequence of input symbols h(a).
- Formally, $\delta_B(q, a) = \delta_A(q, h(a))$.

Example: Inverse Homomorphism Construction



$h(0) = ab$
$h(1) = \epsilon$



Since
$h(1) = \epsilon$

Since
$h(0) = ab$

Proof – (3)

- Induction on |w| shows that $\delta_B(q_0, w) = \delta_A(q_0, h(w))$.
- Basis: $w = \epsilon$.
- $\delta_B(q_0, \epsilon) = q_0$, and $\delta_A(q_0, h(\epsilon)) = \delta_A(q_0, \epsilon) = q_0$.

Proof – (4)

- Induction: Let $w = xa$; assume IH for x.
- $\delta_B(q_0, w) = \delta_B(\delta_B(q_0, x), a)$.
- $= \delta_B(\delta_A(q_0, h(x)), a)$ by the IH.
- $= \delta_A(\delta_A(q_0, h(x)), h(a))$ by definition of the DFA B.
- $= \delta_A(q_0, h(x)h(a))$ by definition of the extended delta.
- $= \delta_A(q_0, h(w))$ by def. of homomorphism.