Propiedades de Decisión de los Lenguajes Regulares

Alan Reyes-Figueroa

Teoría de la Computación **(Aula 09) 11.agosto.2025**

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 $q \in Q$, $r \in R$.

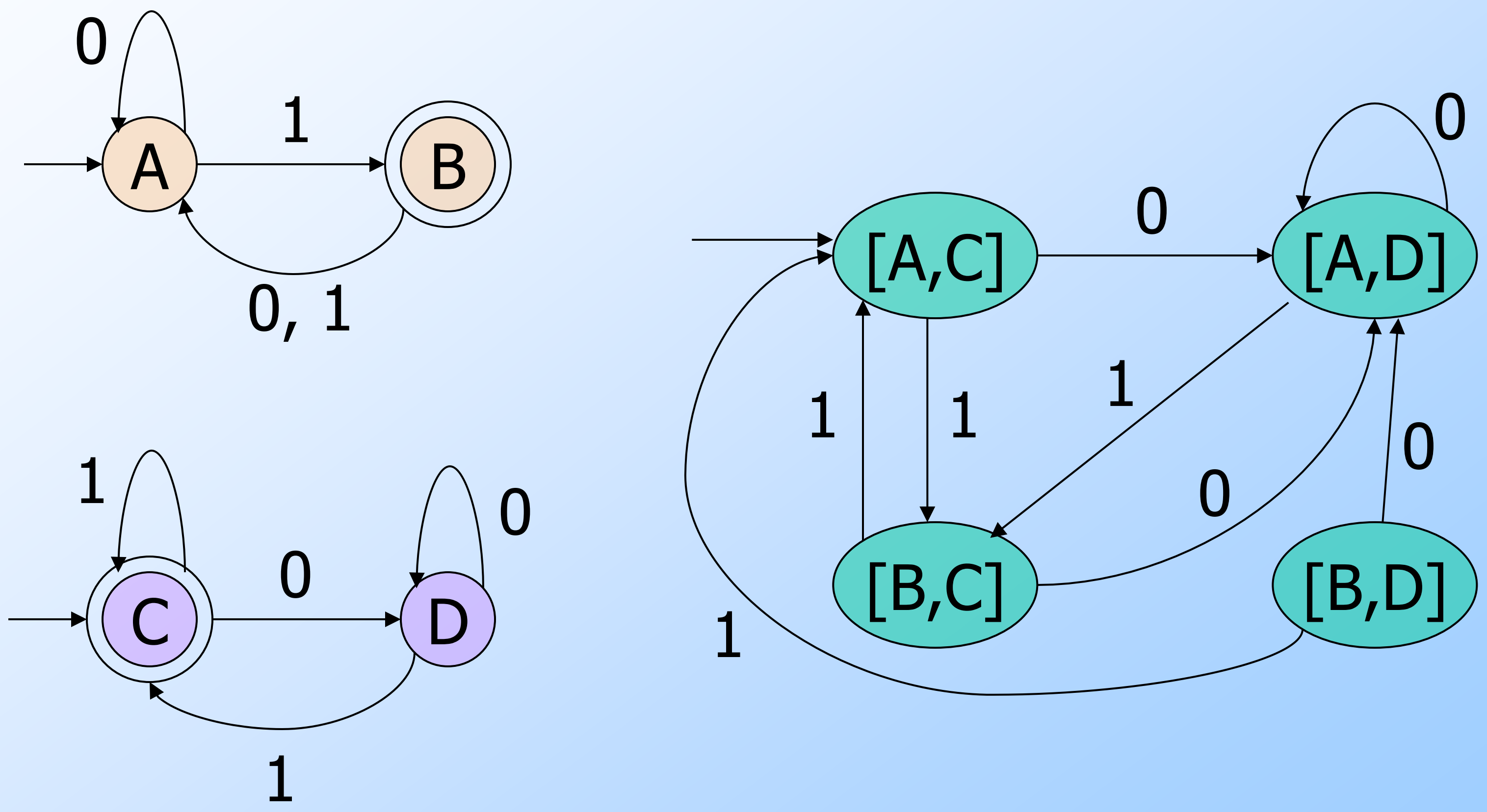
DFA Producto

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

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.

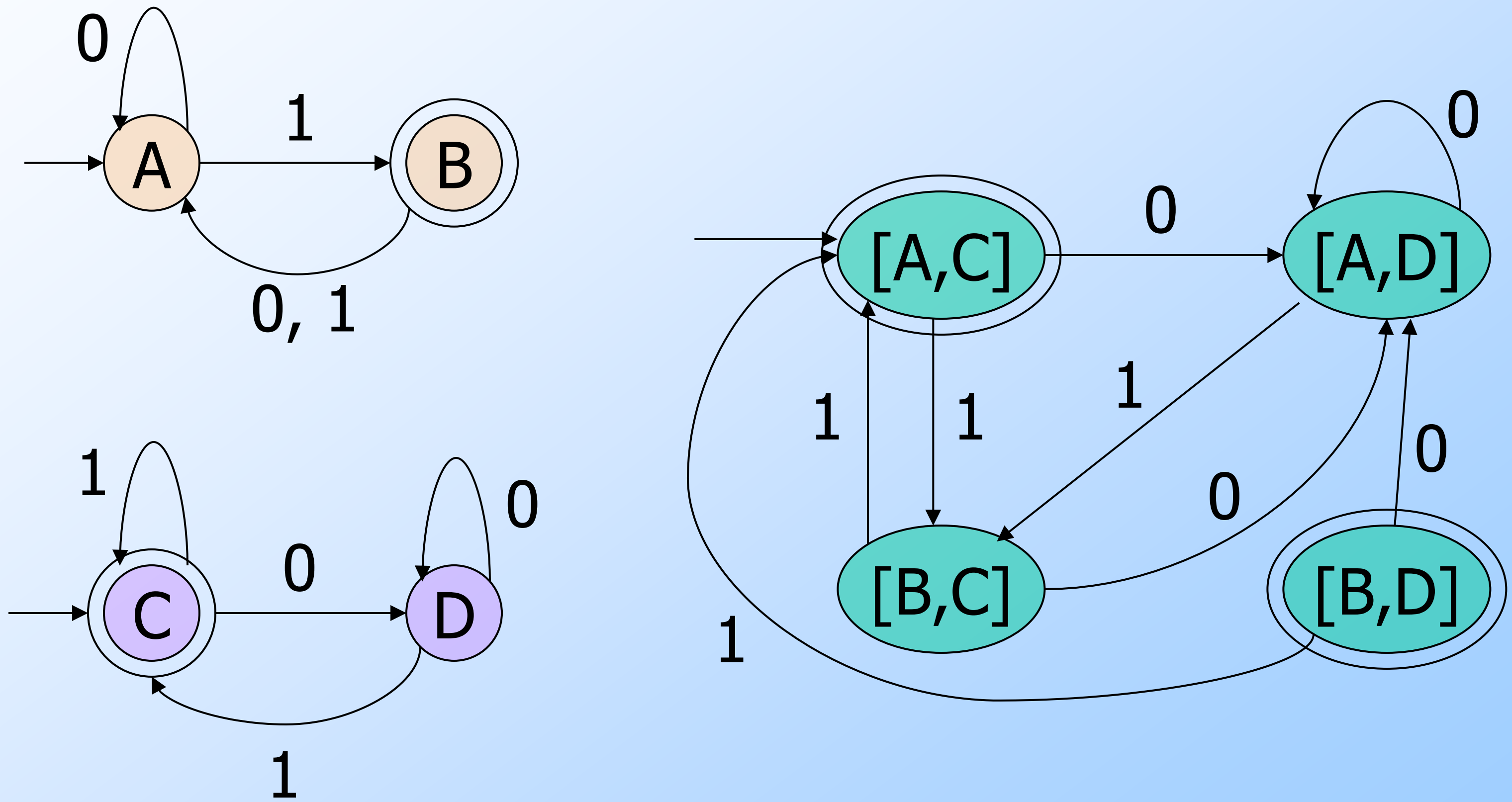
Ejemplo: DFA Producto



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.

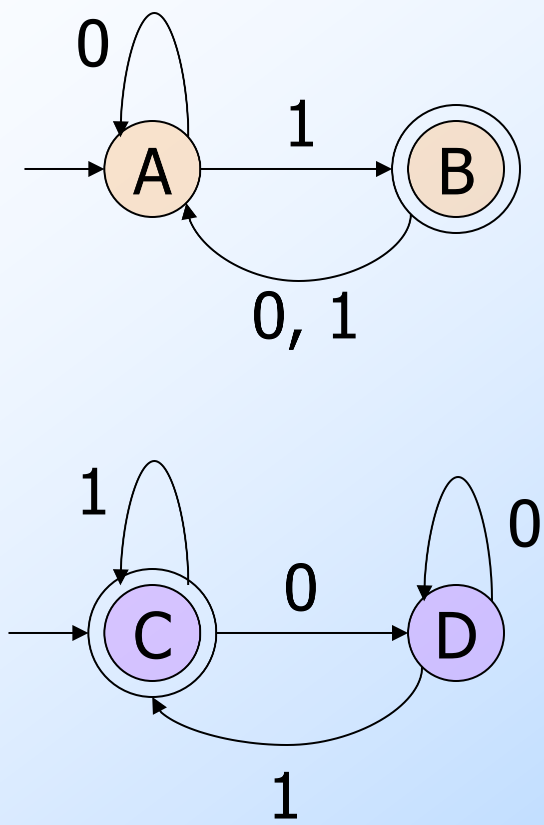
- Si recordamos, ya tenemos un algoritmo para evaluar si el lenguaje generado por un DFA es vacío.

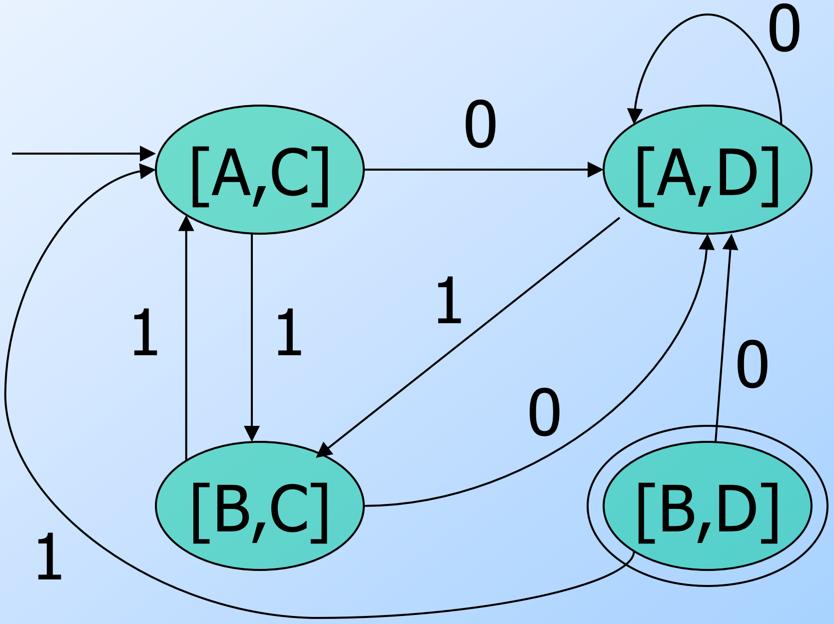
Propiedad: Inclusión

- Dados lenguajes regulares L y M, está $L \subseteq M$?
- Tenemos un algoritmos 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.