Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2223 - Teoría de Autómatas y Lenguajes Formales

Ayudantía 1

Dante Pinto
26 de Agosto, 2021

Pregunta 1

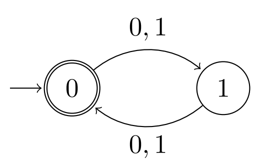
Demuestre que los siguientes lenguajes son regulares:

- $L = \{w \in \{0,1\}^* \mid |w| \mod 2 \equiv 0 \}$
- $L = \{w \in \{0,1\}^* \mid |w| \mod 3 \equiv 0 \}$
- $L = \{w \in \{0,1\}^* \mid |w| \mod 6 \equiv 0 \}$

Para demostrar que estos lenguajes son regulares debemos encontrar autómatas que los definan y luego demostrar que el lenguaje definido por ellos es efectivamente L.

- $L = \{w \in \{0,1\}^* \mid |w| \mod 2 \equiv 0 \}$

  Podemos generar el siguiente DFA, $\mathcal{A}$:



  Ahora para demostrar que los lenguajes son igual, demostramos:

  1. $\mathcal{L}(\mathcal{A}) \subseteq L$

     Si una palabra $w = a_1a_2...a_n$ es aceptada por el autómata, existirá una ejecución de aceptación:

$$\rho : \ p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_0 \xrightarrow{a_3} ... \xrightarrow{a_{n-1}} p_1 \xrightarrow{a_n} p_0$$

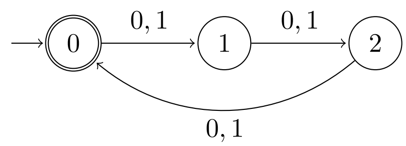
     Dado que nuestro único estado de aceptación es p_0 y sin importar qué letra se lea, siempre intercalaremos entre p_0 y p_1, la única forma de que la ejecución sea de aceptación, es que tenga una cantidad par de pasos, es decir, $n = 2k, k \in \mathbb{N}$.

     Luego, dado que tenemos una cantidad de pasos, estamos leyendo una cantidad par de letras y por tanto el largo de las palabras aceptadas deberá cumplir que $|w| \mod 2 \equiv 0$, lo que significa que $\mathcal{L}(\mathcal{A}) \subseteq L$.

2. $\mathcal{L}(\mathcal{A}) \subseteq L$
   Toda palabra que pertenece al lenguaje debe tener largo par, por lo que podemos demostrar por inducción sobre el largo de las palabras que para todas ellas existirá una ejecución de aceptación en $\mathcal{A}$.

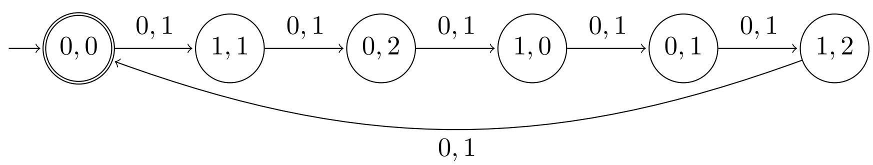
- $L = \{w \in \{0,1\}^* \mid |w| \mod 3 \equiv 0 \}$

  Similar al caso anterior, podemos generar el siguiente DFA:



  La demostración de este caso es análoga a la del anterior.

- $L = \{w \in \{0,1\}^* \mid |w| \mod 6 \equiv 0 \}$ Para construir este autómata, podemos aprovechar que todo número múltiplo de 6 es también múltiplo de 2 y 3, por lo que podemos utilizar los autómatas anteriores para construir:



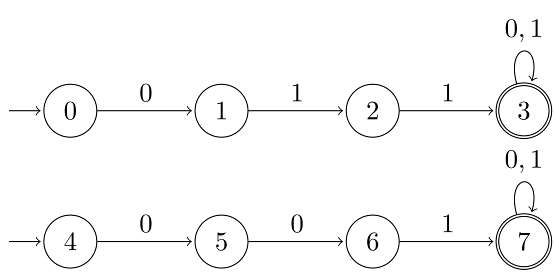
  Y nuevamente la demostración de este caso será análoga a la primera.

Pregunta 2

Considere $\Sigma = \{0, 1\}$ y construya un NFA para cada uno de los siguientes lenguajes:

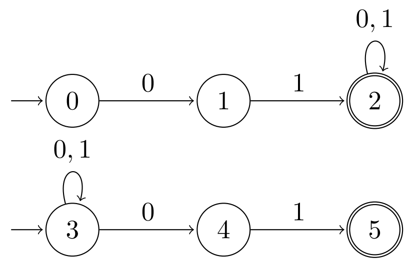
- $L = \{w = a_1a_2 \ldots a_{n-1}a_n \mid a_1a_2a_3 = 011 \vee a_1a_2a_3 = 001 \}$

  Intuitivamente, podemos crear dos autómatas, uno para cada una de las condiciones del *or*, y, dado que buscamos un autómata no determinista, el autómata formado por estos dos definirá el lenguaje que buscamos.



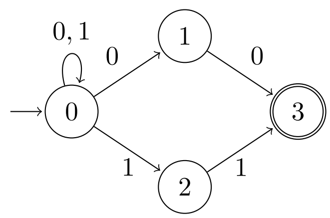
- $L = \{w = a_1 a_2 \ldots a_{n-1} a_n \mid a_1 a_2 = 01 \lor a_{n-1} a_n = 01\}$

  Podemos seguir la misma estrategia en este caso, obteniendo:



- $L = \{w = a_1 a_2 \ldots a_{n-1} a_n \mid a_{n-1} = a_n\}$

  Para este lenguaje, podemos crear el siguiente autómata:



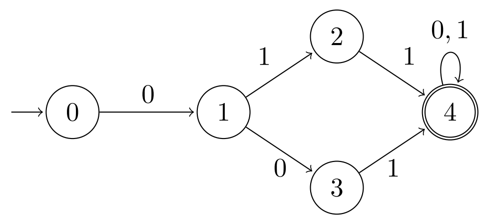
Pregunta 3

Construya un DFA para cada uno de los lenguajes de la pregunta 2. ¿Cómo se comparan con los NFAs construidos?

Para construir los DFA que buscamos, tenemos dos opciones, intentar construirlos desde 0 o determinizar los NFA que ya tenemos y dependiendo del lenguaje, el enfoque más conveniente puede ser diferente.

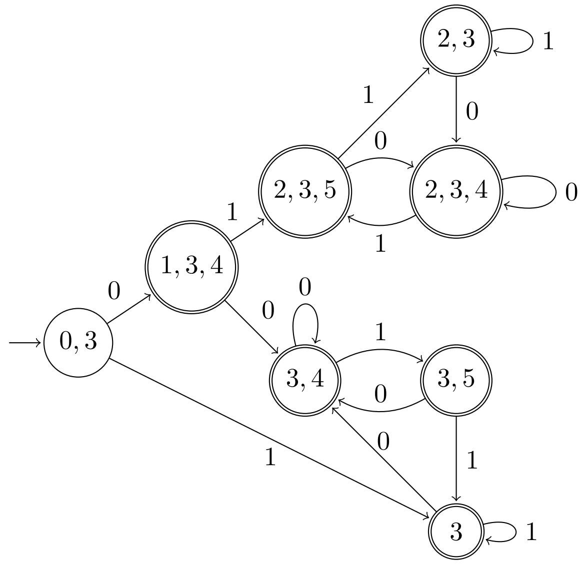
- $L = \{w = a_1 a_2 \ldots a_{n-1} a_n \mid a_1 a_2 a_3 = 011 \lor a_1 a_2 a_3 = 001\}$

  En este caso hay una autómata determinista equivalente al que escribimos antes que no es muy difícil de ver, por lo que podemos construirlo:



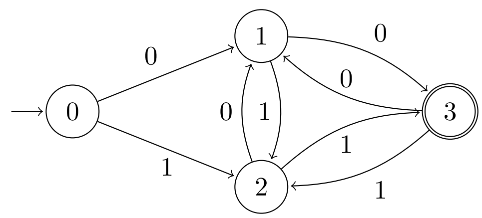
- $L = \{w = a_1 a_2 \ldots a_{n-1} a_n \mid a_1 a_2 = 01 \vee a_{n-1} a_n = 01 \}$

  En este caso no es trivial como cosntruir el NFA que buscamos, por lo que podemos determinizar el autómata anterior, encontrando:

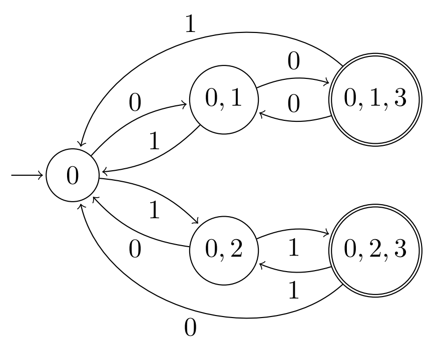


- $L = \{w = a_1 a_2 \ldots a_{n-1} a_n \mid a_{n-1} = a_n \}$

  Nuevamente, podemos diseñar un autómata para este lenguaje, obteninedo:



  Por otra parte, si decidimos determinizar el autómata anterior, tenemos:



Observando ahora los autómatas encontrados, podemos ver que los NFA son, por lo general, más simples y compactos en su construcción que los DFA equivalentes. Esto se hace particularmente notorio al observar los autómatas generados al determinizar los NFA, pues el número de estados puede llegar a ser $Q' = 2^Q$.

Pregunta 4

Sea Σ un alfabeto finito y $L_1, L_2 \subseteq \Sigma^*$ dos lenguajes. Se define el lenguaje

$$L_1 \star L_2 = \{w \in \Sigma^* \mid \text{existen } u_1, u_2, u_3 \in \Sigma^* \text{ tal que } w = u_1u_2u_3,\ u_1u_2 \in L_1 \text{ y } u_2u_3 \in L_2\}$$

Demuestre que si L_1 y L_2 son lenguajes regulares, entonces $L_1 \star L_2$ es regular.
Para demostrar que $L_1 \star L_2$ es un lenguaje regular, podemos encontrar un autómata que lo defina. Como L_1 y L_2 son lenguajes regulares, sabemos que existirán dos DFA $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$, $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ tales que $\mathcal{L}(\mathcal{A}_1) = L_1$ y $\mathcal{L}(\mathcal{A}_2) = L_2$.
Podemos construir un NFA $\mathcal{A}$ que defina $L_1 \star L_2$ a partir de $\mathcal{A}_1$ y $\mathcal{A}_2$; intuitivamente, $\mathcal{A}$ debe leer u_1 y ejecutar $\mathcal{A}_1$, luego (de manera no determinista) comenzar a leer u_2, ejecutando ambos autómatas a la vez, para finalmente leer u_3 ejecutando solamente $\mathcal{A}_2$.
La construcción de $\mathcal{A}$ será $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$, con:

- $Q = Q_1 \cup Q_2 \cup Q_1 \times Q_2$
- $I = \{q_{01}\}$
- $F = F_2$
-

$$\begin{aligned}
\Delta = & \big\{(p, a, q) \mid \delta_1(p, a) = q \lor \delta_2(p, a) = q\big\} && \text{(Transiciones originales de los automatas)} \\
& \cup \big\{((p_1, p_2), a, (q_1, q_2)) \mid \delta_1(p_1, a) = q_1 \land \delta_2(p_2, a) = q_2\big\} && \text{(Transiciones del producto cruz)} \\
& \cup \big\{(p_1, a, (q_1, q_2)) \mid \delta_1(p_1, a) = q_1 \land q_2 = q_{02}\big\} && \text{(Continuo } \mathcal{A}_1 \text{ y ejecuto } \mathcal{A}_2 \text{ del inicio)} \\
& \cup \big\{((p_1, p_2), a, q_2) \mid \delta_2(p_2, a) = q_2 \land p_1 \in F_1\big\} && \text{(Ejecuto } \mathcal{A}_2 \text{ solo si } \mathcal{A}_1 \text{ terminó)}
\end{aligned}$$

Teniendo nuestra construcción, ahora debemos demostrar $\mathcal{A}$ define $L_1 \star L_2$.

- $L_1 \star L_2 \subseteq \mathcal{L}(\mathcal{A})$

  Sea $w = u_1u_2u_3 \in L_1 \star L_2$. Existen entonces dos palabras u_1u_2 y u_2u_3 tales que $u_1u_2 \in L_1$ y $u_2u_3 \in L_2$. Tendremos entonces las siguientes ejecuciones de acpetación de $\mathcal{A}_1$ y $\mathcal{A}_2$, respectivamente:

  $$\rho_1 :\ q_{01} \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p_n \ \land\ p_n \in F_1$$
  $$\rho_2 :\ q_{02} \xrightarrow{b_1} q_1 \xrightarrow{b_2} \dots \xrightarrow{b_m} q_m \ \land\ q_m \in F_2$$

  Considerando ahora que $|u_1| = i$, $|u_2| = j$, podemos reescribir las ejecuciones como:

  $$\rho_1 :\ q_{01} \xrightarrow{a_1} \dots \xrightarrow{a_i} p_i \xrightarrow{a_{i+1}} \dots \xrightarrow{a_n} p_n$$
  $$\rho_2 :\ q_{02} \xrightarrow{b_1} \dots \xrightarrow{b_j} q_j \xrightarrow{b_{j+1}} \dots \xrightarrow{b_m} q_m$$

  Como todas las transiciones y estados de $\mathcal{A}_1$ y $\mathcal{A}_2$ están también en $\mathcal{A}$, las siguientes secuencias (que definen a u_1 y u_3) también lo estarán:

  $$\rho_{u_1} :\ q_{01} \xrightarrow{a_1} \dots \xrightarrow{a_{i-1}} p_{i-1} \xrightarrow{a_i} p_i$$
  $$\rho_{u_3} :\ q_j \xrightarrow{b_{j+1}} q_{j+1} \xrightarrow{b_{j+2}} \dots \xrightarrow{b_m} q_m$$

Por otra parte, el resto de las ejecuciones ρ_1 y ρ_2 definen a la misma palabra, u_2, y debido a la construcción de $\mathcal{A}$, podemos armar la siguiente secuencia, válida para $\mathcal{A}$:

$$\rho_{u_2}:\ (p_i, q_{02}) \xrightarrow{a_{i+1}=b_1} (p_{i+1}, q_1) \xrightarrow{a_{i+2}=b_2} \ldots \xrightarrow{a_n=b_j} (p_n, q_j)$$

Finalmente, podemos unir las 3 secuencias usando dos transiciones, una de cada uno de los últimos conjuntos de la definición de Δ, $\big(p_{i-1}, a_i, (p_i, q_{02})\big)$ y $\big((p_n, q_j), b_{j+1}, q_{j+1}\big)$, formando la siguiente ejecución:

$$\rho_w:\ q_{01} \xrightarrow{a_1} \ldots \xrightarrow{a_{i-1}} p_{i-1} \xrightarrow{a_i} (p_i, q_{02}) \xrightarrow{a_{i+1}=b_1} \ldots \xrightarrow{a_n=b_j} (p_n, q_j) \xrightarrow{b_{j+1}} q_{j+1} \xrightarrow{b_{j+2}} \ldots \xrightarrow{b_m} q_m$$

Es claro que ρ_w es una ejecución de aceptación, pues $q_{01} \in I$, $q_m \in F$ y todo paso de la ejecución sigue una transición de Δ.

Por lo tanto, $L_1 \star L_2 \subseteq \mathcal{L}(\mathcal{A})$.

- $\mathcal{L}(\mathcal{A}) \subseteq L_1 \star L_2$

  Sea w una palabra tal que $w \in \mathcal{L}(\mathcal{A})$. Sabemos que existirá una ejecución de aceptación de $\mathcal{A}$ sobre w de la forma:

$$\rho_w:\ q_{01} \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots \xrightarrow{a_k} p_k\ \wedge\ p_k \in F$$

  Debido a la forma de $\mathcal{A}$, para llegar a un estado final debe pasar por sus 3 ramas (Q_1, Q_2 y $Q_1 \times Q_2$), pues el estado inicial se encuentra en la rama Q_1, los finales en la rama Q_2 y no hay transiciones entre estas ramas, por lo que para llegar de una a otra se debe pasar por $Q_1 \times Q_2$.

  Esto significa entonces que dentro de ρ deben existir dos transiciones de la forma $(p_{i-1}, a, (\delta_1(p_{i-1}), q_{02}))$ y $((p_n, q_j), a, \delta_2(q_j)),$ con $p_n \in F_1$. Cabe señalar que los índices se seleccionaron por conveniencia para asemejarse a la parte anterior.

  Tomando en cuenta estas dos transiciones, podemos separar la palabra w en los 3 segmentos que se forman entre ellas, nombrar estas palabras como u_1, u_2, u_3 y reescribir la ejecución como:

$$\rho_w:\ q_{01} \xrightarrow{a_1} \ldots \xrightarrow{a_{i-1}} p_{i-1} \xrightarrow{a_i} (p_i, q_{02}) \xrightarrow{a_{i+1}} \ldots \xrightarrow{a_n} (p_n, q_j) \xrightarrow{a_{n+1}} p_{n+1} \xrightarrow{a_{n+2}} \ldots \xrightarrow{a_k} p_k$$

  Luego, sabemos que toda la ejecución antes de la transición $(p_{i-1}, a, (\delta_1(p_{i-1}), q_{02}))$ ocurre solamente con estados y transiciónes de $\mathcal{A}_1$ (debido a la definición de Δ). Además toda la ejecución entre este punto y la transición $((p_n, q_j), a, \delta_2(q_j)),$ con $p_n \in F_1$ ocurre dentro del producto cruz entre los autómatas, lo que significa que para cada transición de la forma $(p_i, q_j) \xrightarrow{a_{i+1}} (p_{i+1}, q_{j+1})$ se cumplirá que $\delta_1(p_i, a_{i+1}) = p_{i+1}$. Uniendo esto con que, por definición, la existencia de $(p_{i-1}, a, (\delta_1(p_{i-1}, a), q_{02}))$ implica que $\delta_1(p_{i-1}, a) = p_i$, podemos armar la siguiente ejecución de $\mathcal{A}_1$ sobre $u_1 u_2$:

$$\rho_1:\ q_{01} \xrightarrow{a_1} \ldots \xrightarrow{a_i} p_i \xrightarrow{a_{i+1}} \ldots \xrightarrow{a_n} p_n\ \wedge\ p_n \in F_1$$

  Similarmente, sabemos que para toda transición de la forma $(p_i, q_j) \xrightarrow{a_{i+1}} (p_{i+1}, q_{j+1})$ se cumplirá que $\delta_2(q_j, a_{i+1}) = q_{j+1}$ y además que toda la ejecución desde la transición $((p_n, q_j), a, \delta_2(q_j)),$ con $p_n \in F_1$ hasta el final ocurre solamente con estados y transiciones de $\mathcal{A}_2$, por lo que usando esto y que la transición $((p_n, q_j), a, \delta_2(q_j)),$ con $p_n \in F_1$ implica que $delta_2(q_j, a) = q_{j+1}$, podemos armar la siguiente ejecución de $\mathcal{A}_2$ sobre $u_2 u_3$:

$$\rho_2:\ q_{02} \xrightarrow{a_{i+1}} \ldots \xrightarrow{a_n} q_j \xrightarrow{a_{n+1}} \ldots \xrightarrow{a_k} p_k\ \wedge\ p_k \in F_2$$

Luego, como tenemos ejecuciones de aceptación de $\mathcal{A}_1$ sobre u_1u_2 y de $\mathcal{A}_2$ sobre u_2u_3, la palabra $w = u_1u_2u_3$ deberá pertenecer a $L_1 \star L_2$ y entonces $L_1 \star L_2 \subseteq \mathcal{L}(\mathcal{A})$.

Por lo tanto, $\mathcal{L}(\mathcal{A}) = L_1 \star L_2$