



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2223 — Teoría de Autómatas y Lenguajes Formales — 2' 2021

## Tarea 1 – Respuesta Pregunta 1

Sabemos que si  $L$  es un lenguaje regular, existe un DFA que acepta el dicho lenguaje. En otras palabras, existe un DFA  $A = (Q, \Sigma, \delta, q_0, F)$  tal que  $L(A) = L$ .

Se define la DNF  $A' = (Q, \Sigma, \delta, q_0, F')$ , donde  $F'$  se define como

$$F' = \{q \in Q \mid \delta(q, x) = p, p \in F, x \in \Sigma\}$$

En otras palabras,  $F'$  es el conjunto de estados tales que tienen una transición a un estado final en la máquina  $A$  que acepta  $L$ . Se quiere demostrar que  $L - 1 = L(A')$ , es decir,  $A'$  define el lenguaje  $L - 1$ .

$$L - 1 \subseteq L(A')$$

Sea  $w = a_1a_2\dots a_n$ , tal que  $w \in \Sigma^*$  y  $w \in L - 1$ . Por definición de  $L - 1$ , existe un  $a \in \Sigma$  tal que  $w \cdot a \in L$ . A continuación, se muestra una ejecución de  $w \cdot a$  sobre el DFA  $A$  que posee el lenguaje  $L$ :

$$q_0 \rightarrow p_1 \rightarrow \dots \rightarrow p_{n-1} \rightarrow p_n$$

Donde  $p_i \in Q$  para todo  $i \leq n$ , y  $p_n \in F$ .

$A$  tiene la misma función de transición  $\delta$ , estado inicial  $q_0$ , alfabeto  $\Sigma$ , y estados  $Q$  que  $A'$ , por lo que para un mismo input, sus ejecuciones son equivalentes, y la ejecución de  $A$  y  $A'$  para el input  $w \cdot a$  son idénticas. Si ejecutamos  $A'$  con  $w$ , el estado final de la ejecución será  $p_{n-1}$ . Por definición de  $A'$ ,  $p_{n-1} \in F'$ , por lo que la ejecución también será de aceptación, y  $w \in L(A')$ .

$$L(A') \subseteq L - 1$$

Por contradicción, supongamos que existe un  $w = a_1a_2\dots a_n$ , tal que  $w \in L(A')$ , pero  $w \notin L - 1$ . Entonces,  $w$  posee la siguiente ejecución de aceptación en  $L(A')$ :

$$q_0 \rightarrow p_1 \rightarrow \dots \rightarrow p_{n-1} \rightarrow p_n$$

Ahora, por definición de  $L - 1$ , si  $w \notin L - 1$ , entonces  $w \cdot a \notin L$ , para cualquier  $a \in \Sigma$ . La máquina  $A$  que posee el lenguaje  $L$  tendría la siguiente ejecución sobre  $w \cdot a$ .

$$q_0 \rightarrow p_1 \rightarrow \dots \rightarrow p_n \rightarrow p_{n+1}$$

Esta ejecución rechaza  $w \cdot a$ , por lo que  $p_{n+1} \notin F$ . Aquí, se pueden dar dos casos:

1. Existe otro  $x \in \Sigma$  para el cual  $A$  acepta  $w \cdot x$ . Aquí se evidencia una contradicción, debido a que si existe otro  $x$ , entonces, por definición de  $L - 1$ ,  $w \in L - 1$ .
2. No existe ningún otro  $x \in \Sigma$  para el cual  $A$  acepta  $w \cdot x$ . En este caso, surge otra contradicción, debido a que los estados finales de  $A'$  fueron definidos de la siguiente manera:  $F' = \{q \in Q \mid \delta(q, x) = p, p \in F, x \in \Sigma\}$ , por lo que sería imposible que  $p_n \in F'$  y  $w \in L(A')$ .

Por lo tanto,  $w$  **debe** pertenecer al lenguaje  $L - 1$ , y en consecuencia,  $L(A') \subseteq L - 1$ . Dado que se encontró un DFA  $A'$  que posee el lenguaje  $L - 1$  (para  $L$  regular), se puede concluir que  $L - 1$  es un lenguaje regular.  $\square$



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

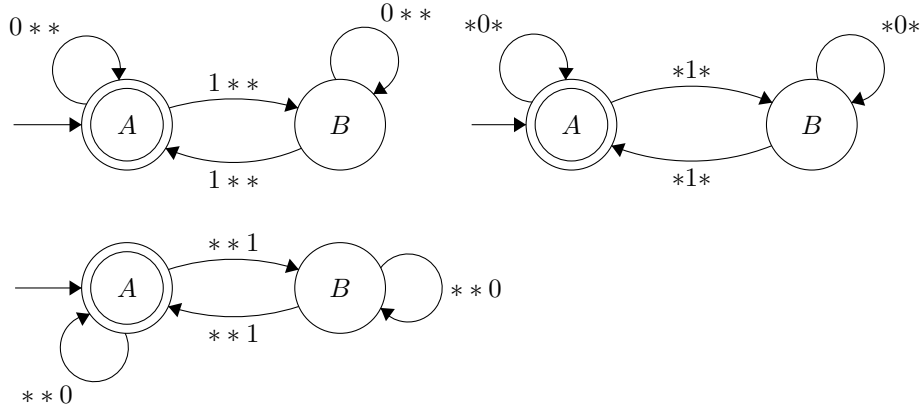
IIC2223 — Teoría de Autómatas y Lenguajes Formales — 2' 2021

## Tarea 1 – Respuesta Pregunta 2

### Parte 1

$$L_1 = \{\vec{v}_1 \vec{v}_2 \dots \vec{v}_n \in (\{0, 1\}^3)^* \mid n \geq 1 \wedge \sum_{i=1}^n \vec{v}_i = [0 \ 0 \ 0]^t\}$$

Para definir un autómata capaz de aceptar este lenguaje, primero se definen 3 autómatas



El símbolo \* representa que en esa posición puede haber un 0 o un 1. Cada una de estas máquinas sólo cambia de estado si en la coordenada que revisa existe un 1. Si al terminar la ejecución se encuentra con un número par de unos en su coordenada, significa que la suma total de esa coordenada resulta en 0, y si es impar en 1, según la definición de suma del enunciado. Estas tres máquinas deben ejecutarse sobre un mismo  $w$  “paralelo”, y si todas aceptan, entonces  $w \in L_1$ . Para lograr este comportamiento, se pueden componer resultando en el siguiente DFA:

$$A = (Q, \Sigma, \delta, q_0, F)$$

$$Q = \{(A, A, A), (A, A, B), (A, B, A), (A, B, B), (B, A, A), (B, A, B), (B, B, A), (B, B, B)\}$$

$$\Sigma = \{0, 1\}^3$$

$$q_0 = (A, A, A)$$

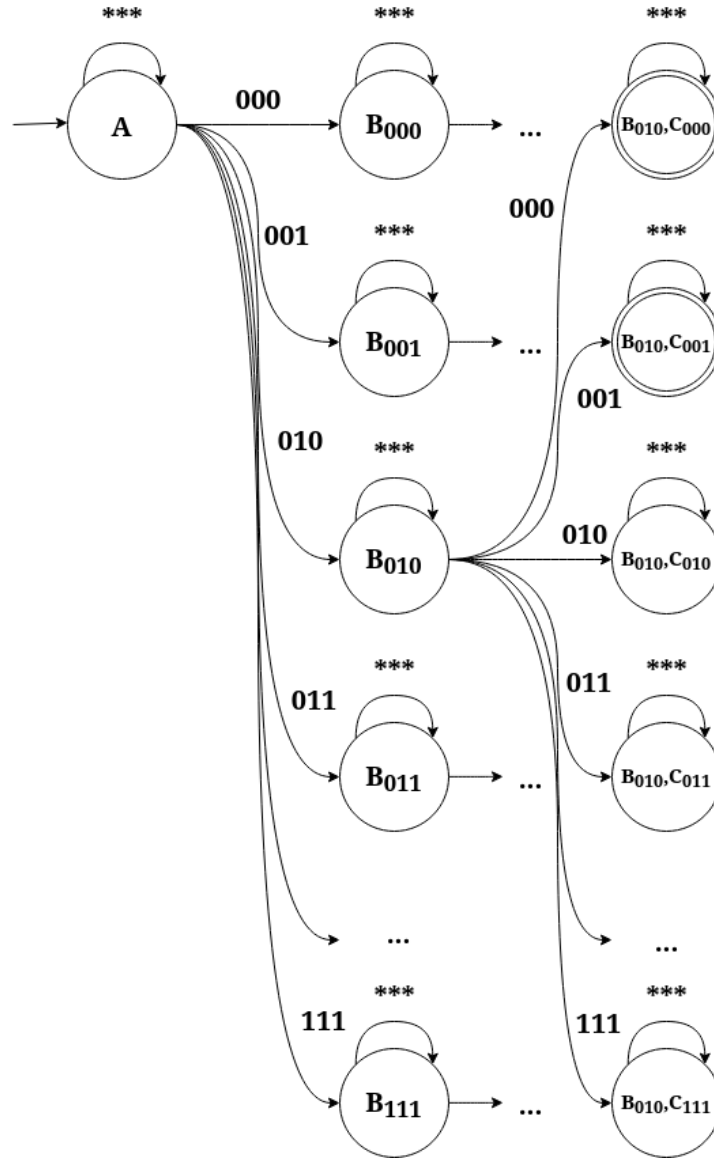
$$F = \{(A, A, A)\}$$

$$\begin{array}{lll} \delta((A, x_1, x_2), 1**) = (B, x_1, x_2) & \delta((x_1, A, x_2), *1*) = (x_1, B, x_2) & \delta((x_1, x_2, A), **1) = (x_1, x_2, B) \\ \delta((B, x_1, x_2), 1**) = (A, x_1, x_2) & \delta((x_1, B, x_2), *1*) = (x_1, A, x_2) & \delta((x_1, x_2, B), **1) = (x_1, x_2, A) \\ \delta((A, x_1, x_2), 0**) = (A, x_1, x_2) & \delta((x_1, A, x_2), *0*) = (x_1, A, x_2) & \delta((x_1, x_2, A), **0) = (x_1, x_2, A) \\ \delta((B, x_1, x_2), 0**) = (B, x_1, x_2) & \delta((x_1, B, x_2), *0*) = (x_1, B, x_2) & \delta((x_1, x_2, B), **0) = (x_1, x_2, B) \end{array}$$

Para simplificar la definición de  $\delta$ ,  $x_1$  y  $x_2$  pueden tomar los valores del estado  $A$  o  $B$ , pero se mantiene constante dentro de una transición. Es simple ver que la construcción es correcta, basándose en la abstracción inicial, debido a que resulta de componer las 3 máquinas previamente definidas y ejecutarlas en paralelo. El estado final de aceptación es único (e igual al inicial), y corresponde a la intersección de los estados de aceptación de las primeras tres máquinas. La máquina sólo se encuentra en este estado cuando ha visto una cantidad par de ceros en todas las coordenadas, o en otras palabras, la suma actual de los vectores, resulta en el vector nulo.

## Parte 2

$$L_2 = \{\vec{v}_1\vec{v}_2...\vec{v}_n \in (\{0,1\}^3)^* \mid n \geq 1 \wedge \exists i, j. i \neq j \wedge \vec{v}_i^t \cdot \vec{v}_j = 0\}$$



El diagrama anterior describe una autómatas no determinista que posee el lenguaje pedido. Similar a la

pregunta anterior, el  $*$  indica que la coordenada del vector puede tener valor 0 o 1. Su funcionamiento es el siguiente:

El primer estado (estado inicial  $A$ ) funciona de manera no determinista, lo que permite que pueda tomar cualquier vector del input  $w$  como valor para  $v_i$ .

El segundo set de estados, corresponde a aquellos de la forma  $B_{v_1v_2v_3}$ , donde  $v_1v_2v_3$  correspondería al vector “elegido” en la ejecución actual como  $v_i$ . Nuevamente, estos estados funcionan de manera no determinista, para permitir la comparación del vector elegido con cualquier otro vector  $v_j$  de  $w$ , con  $j > i$ .

El tercer y último set de estados, corresponde a aquellos de la forma  $(B_{v_1v_2v_3}, C_{u_1u_2u_3})$ , donde ahora  $v_1v_2v_3$  correspondería al vector “elegido” como  $v_j$  para compararlo con  $v_i = v_1v_2v_3$ . El estado pertenece a  $F$  (es decir, es final) si y sólo si  $v_i \cdot v_j = 0$ . Para los casos donde  $v_i \cdot v_j = 1$ ,  $(B_{v_1v_2v_3}, C_{u_1u_2u_3}) \notin F$ . Los estados apuntan a sí mismo en todas sus transiciones, para no rechazar  $w$  en el caso en que  $j = n$ .

Dado que la máquina en general es no determinista, y la manera en que se diseñaron los estados, es posible “realizar todas las comparaciones” de  $v_i$  y  $v_j$  en  $w$ , con  $i < j$  y  $j \leq n$ . Si existe una combinación donde  $v_i \cdot v_j = 0$  en  $w$ , entonces existirá una ejecución de aceptación donde se alcance  $(B_{v_1v_2v_3}, C_{u_1u_2u_3}) \in F$ . En caso contrario, donde no exista una combinación posible o  $n = 1$ , ninguna ejecución logrará llegar a un estado final y el NFA rechaza la palabra (y por lo tanto,  $w \notin L_2$ ).

La definición formal de esta máquina estaría dada por:

$$A = (Q, \Sigma, \delta, q_0, F)$$

$$Q = \{A, B_{000}, B_{001}, \dots, B_{111}, (B_{000}, C_{000}), (B_{000}, C_{001}), \dots, (B_{000}, C_{111}), (B_{001}, C_{000}), \dots, (B_{111}, C_{111})\}$$

$$\Sigma = \{0, 1\}^3$$

$$I = \{A\}$$

$$F = \{(B_{v_1v_2v_3}, C_{u_1u_2u_3}) \in Q \mid v_1v_2v_3, u_1u_2u_3 \in \Sigma \wedge [v_1 \ v_2 \ v_3] \cdot [v_1 \ v_2 \ v_3]^t = 0\}$$

Las transiciones se definen como:

$$\begin{aligned} \Delta = & \{(q, ***, q) \mid q \in Q\} \\ & \cup \{(A, v_1v_2v_3, B_{v_1v_2v_3}) \mid v_1v_2v_3 \in \Sigma\} \\ & \cup \{(B_{v_1v_2v_3}, u_1u_2u_3, (B_{v_1v_2v_3}, C_{u_1u_2u_3})) \mid v_1v_2v_3, u_1u_2u_3 \in \Sigma\} \end{aligned}$$