PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
FACULTAD DE MATEMÁTICAS
DEPARTAMENTO DE MATEMÁTICA

Tarea 1

Teoría de Autómatas y Lenguajes Formales — IIC2223

Fecha de Entrega: 2020-08-27

Nicholas Mc-Donnell

Problema 1:

Sea $\Sigma = \{\#, a\}$. Para $i \geq 1$ se define $a^i = a \overset{i-\text{veces}}{\cdots} a$. Para $n \geq 2$ se define el Lenguaje $L_n \subseteq \Sigma^*$ de todas las palabras de la forma:

$$\#a^{i_1}\#a^{i_2}\ldots\#a^{i_k}$$

para algún $k \geq 0$ tal que $1 \leq i_j \leq n$ para todo $1 \leq j \leq k$. Por ejemplo, para $n = 3$ se tiene que $\#aa\#a$ y $\#a\#aaa\#aa$ pertenecen a L_3, pero $\#aaaa$ no pertenece a L_3. Notar que cuando $k = 0$ se tiene que $w = \varepsilon$, y por lo tanto siempre se cumple que $\varepsilon \in L_n$.

(a) Para un n arbitrario, muestre como construir un autómata finito determinista $\mathcal{A}_n$ tal que $\mathcal{L}(\mathcal{A}_n) = L_n$. Explique por qué su construcción cumple con lo pedido.

(b) Para un n arbitrario, muestre como construir un autómata finito no-determinista $\mathcal{B}_n$ con $n + 1$ estados tal que $\mathcal{L}(\mathcal{B}_n) = L_n$. Explique por qué su construcción cumple con lo pedido.

Solución problema 1:

(a) Sea $Q = \{q_1, q_2, ..., q_{n+1}, q_{n+2}\}$, $F = \{q_1, ..., q_n\}$ y $q_0 = q_n$. Ahora, definimos δ de la siguiente manera:

$$\delta(q_i, a) = \begin{cases} q_{i+1} & \text{si } 1 \leq i < n \\ q_1 & \text{si } i = n+1 \\ q_{n+2} & \text{si } i = n \text{ o } i = n+2 \end{cases}$$
$$\delta(q_i, \#) = \begin{cases} q_{n+1} & \text{si } 1 \leq i \leq n \\ q_{n+2} & \text{si } n+1 \leq i \leq n+2 \end{cases}$$

Con lo anterior se define $\mathcal{A}_n$. Por claridad, se denota q_{n+2} como un estado 'basura', q_{n+1} como un 'comienzo'[1] y q_i con $1 \leq i \leq n$ como un estado de i 'a'. Ahora, sea $w \in L_n$, se nota que $w = \#a^i \cdot z$ con $z \in L_n$, donde $\cdot$ es la operación de concatenación. Dado lo anterior, se nota que es suficiente demostrar que ε es aceptada por $\mathcal{A}_n$ y que dado una palabra $w = \#a^i \cdot z$ en L_n, con $z \in L_n$ aceptada por $\mathcal{A}_n$, es aceptada por

[1] En el sentido de que representa un string que comienza con #

$\mathcal{A}_n$. Lo anterior se puede expresar en la siguiente inducción sobre k. Para el caso base, si $k = 0$, se tiene la palabra vacía ε, como es la ejecución vacía se queda en q_0 el cual pertenece a F, por lo que la palabra es aceptada. Ahora, sea $w = \#a^{i_{k+1}} \cdot z$ donde $z = \#a^{i_k} \dots \#a^{i_1}$ y los $1 \leq i_j \leq n$, se tiene que z es aceptada por $\mathcal{A}_n$ por hipótesis inductiva. Luego, se procesa w, comenzando en q_0 se tiene la secuencia de ejecución es $q_n \xrightarrow{\#} q_{n+1} \xrightarrow{a} q_1$ ahora se tiene i_{k+1} 'a', por lo que se tiene que de $q_1 \xrightarrow{a^{i_{k+1}-1}} q_{i_{k+1}}$ donde $i_{k+1} \leq n$, con lo que se empieza la ejecución de z en el estado $q_{i_{k+1}}$, y notamos que $\delta(q_j, \#) = q_n$ para $1 \leq j \leq n$, lo cual es cierto en nuestro caso, por lo que se usa la hipótesis inductiva, y se tiene que para z hay una ejecución que la acepta. Juntando esto con la ejecución de $\#a^{i_{k+1}}$ se tiene una ejecución que acepta a w, con lo que se cumple la hipótesis inductiva para $k + 1$.

Con lo anterior se tiene que $L_n \subseteq \mathcal{L}(\mathcal{A}_n)$. Para demostrar la otra contención, sea $w \in \Sigma^* \setminus L_n$, como $w \notin L_n$, se tiene que $w = z_1 \cdot \#\# \cdot z_2$, $w = z_1 \cdot a^{n+1} \cdot z_2$, $w = z_1 \cdot \#$ o $w = a \cdot z_1$ donde $z_1, z_2 \in \Sigma^*$.

I. Para el primer caso, s.p.d.g. se asume que hay una ejecución de z_1 tal que no termine en q_{n+2}. Dado eso, se tiene que el estado en el que se empieza a evaluar $\#\# \cdot z_2$ es q_i con $1 \leq i \leq n+1$. Si $i = n+1$ se tiene que $\delta(q_{n+1}, \#) = q_{n+2}$, por lo que se entra al estado 'basura', con lo que w no es aceptada. En cambio, si $i \leq n$ se tiene que $\delta(q_i, \#) = q_{n+1}$, pero ahora volvemos a la situación anterior, ya que $\delta(q_{n+1}, \#) = q_{n+2}$, por lo que w no es aceptada.

II. Al igual que el caso anterior, se asume que existe una ejecución de z_1 tal que no termine en q_{n+2}. Por lo que el estado donde se empieza a evaluar $a^{n+1} \cdot z_2$ es q_i con $1 \leq i \leq n+1$. Si $i \leq n$, se tiene que al llegar a $a^{n+1-(n-i)} \cdot z_2$ se está en el estado q_n, pero el siguiente carácter es a, y $\delta(q_n, a) = q_{n+2}$. En el otro caso, $i = n+1$ con lo que $\delta(q_{n+1}, a) = q_1$ y queda $a^n \cdot z_2$ para procesar, similarmente al caso recién mencionado se llega a $a \cdot z_2$ en el estado q_n, pero $\delta(q_n, a) = q_{n+2}$.

III. Como se menciono en los casos anteriores se asume que existe una ejecución de z_1 que no termina en q_{n+2}. Ahora, si $i = n+1$ se tiene que $\delta(q_{n+1}, \#) = q_{n+2}$, en cambio si $i \leq n$ se tiene que $\delta(q_i, \#) = q_{n+1}$ y ninguno es un estado de aceptación, por lo que w no es aceptada.

IV. Como $q_0 = q_n$ y $\delta(q_n, a) = q_{n+2}$ se tiene que w no es aceptada.

Como $w \notin \mathcal{L}(\mathcal{A}_n)$, se tiene que $L_n = \mathcal{L}(\mathcal{A}_n)$.

(b) Para construir $\mathcal{B}_n$ se toma $\mathcal{A}_n$ y se quita el estado basura y las transiciones asociadas al mismo. Se nota que este NFA tiene $n+1$ estados y que cuando se llega a una transición no definida la palabra no es aceptada, se puede usar la misma explicación de la parte anterior, con la diferencia de que en vez de hacer la transición a q_{n+2} y esperar a que se termine de evaluar la palabra, inmediatamente se ve que la palabra no es aceptada.

$\blacksquare$