Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2223 – Teoría de Autómatas y Lenguajes Formales
Segundo semestre de 2024
Profesor: Cristian Riveros
Ayudante: Amaranta Salas

Ayudantia 3

No Determinismo y Expresiones Regulares

Problema 1

Sea Σ un alfabeto finito y $L \subseteq \Sigma^*$ un lenguaje. Se definen los lenguajes

$$L^{suf} = \{v \mid \exists u.\ u \cdot v \in L\}$$

$$L^{\circlearrowleft} = \{w \mid \exists u \exists v.\ w = u \cdot v \ \wedge\ v \cdot u \in L\}$$

Construya L^{suf} para construir $L^{\circlearrowleft}$ y demuestre que si L es lenguaje regular, entonces $L^{\circlearrowleft}$ es regular.

Solución

Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ el autómata que define el mismo lenguaje que L. La idea para construir un autómata que defina el mismo lenguaje que L^{suf} es ejecutar el autómata $\mathcal{A}$, pero desde cualquier estado. Es decir, se parte en cualquier punto de la ejecución hasta llegar a un estado final. La construcción de $\mathcal{A}^{suf} = (Q^{suf}, \Sigma, \Delta^{suf}, I^{suf}, F^{suf})$ queda como

- $Q^{suf} = Q$
- $I^{suf} = Q$
- $F^{suf} = F$
- $\Delta^{suf} = \{(p, a, q) \mid \delta(p, a) = q\}$

Luego, podemos pensar $\mathcal{A}^{\circlearrowleft}$ como leer el sufijo y luego leer la palabra desde el inicio llegando hasta el estado donde se partió leyendo. Esto se puede emular con 2 copias del autómata $\mathcal{A}$ partiendo desde cualquier estado de este, llegando a un estado final, pasar al segundo autómata leyendo desde el estado inicial y llegar al estado en el que se partió en el primero. Para esto necesitamos recordar en que estado se partió y en cual autómata estamos. La construcción de $\mathcal{A}^{\circlearrowleft} = (Q^{\circlearrowleft}, \Sigma, \Delta^{\circlearrowleft}, I^{\circlearrowleft}, F^{\circlearrowleft})$ queda como

- $Q^{\circlearrowleft} = Q \times Q \times \{0, 1\}$
- $I^{\circlearrowleft} = \{(q, q, 0) \mid q \in Q\}$
- $F^{\circlearrowleft} = \{(q, q, 1) \mid q \in Q\}$
- $\Delta^{\circlearrowleft} = \{((p, q_{inicial}, l), a, (q, q_{inicial}, l)) \mid \delta(p, a) = q \ \wedge\ l \in \{0, 1\}\} \cup \{((p_f, q_{inicial}, 0), a, (q_1, q_{inicial}, 1)) \mid \delta(q_0, a) = q_1 \ \wedge\ p_f \in F\}$

Ahora debemos demostrar que $L(\mathcal{A}^{\circlearrowleft}) = L^{\circlearrowleft}$.

- $L(\mathcal{A}^{\circlearrowleft}) \subseteq L^{\circlearrowleft}$

  Sea $w \in \mathcal{A}^{\circlearrowleft}$, podemos separar w como $w = u \cdot v$ con $u = a_{i+1} \dots a_n$ y $v = a_1 \dots a_i$ y existe una ejecución de aceptación $\rho^{\circlearrowleft}$ de $\mathcal{A}^{\circlearrowleft}$

  $$\rho^{\circlearrowleft} : (p_i, p_i, 0) \xrightarrow{a_{i+1}} (p_{i+1}, p_i, 0) \xrightarrow{a_{i+2}} \dots \xrightarrow{a_n} (p_n, p_i, 0) \xrightarrow{a_1} (p_1, p_i, 1) \xrightarrow{a_2} \dots \xrightarrow{a_i} (p_i, p_i, 1)$$

  con $(p_i, p_i, 1) \in F^{\circlearrowleft}$, además por construcción de $\mathcal{A}^{\circlearrowleft}$ sabemos que la transición $((p_n, p_i, 0), a_1, (p_1, p_i, 1))$ tiene $p_n \in F$, $\delta(q_0, a_1) = p_1$ y q_0 es estado inicial de $\mathcal{A}$ y que las transiciones con 0 en la tupla de estado es la lectura de u y las con 1 es la lectura de v. Por lo tanto, como se lee primero u y después v y $v \cdot u \in L$, $w \in L^{\circlearrowleft}$ y $L(\mathcal{A}^{\circlearrowleft}) \subseteq L^{\circlearrowleft}$.

- $L^{\circlearrowleft} \subseteq L(\mathcal{A}^{\circlearrowleft})$

  Sea $w \in L^{\circlearrowleft}$, por definición existen u y v tal que $w = u \cdot v$ y $v \cdot u$ es una palabra perteneciente a L. Como L es un lenguaje regular, existe un autómata $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ que define el mismo lenguaje con una ejecución de aceptación sobre $v \cdot u$ con $v = a_1 \dots a_i$ y $u = a_{i+1} \dots a_n$

  $$\rho : q_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_i} p_i \xrightarrow{a_{i+1}} \dots \xrightarrow{a_n} p_n$$

  con $p_n \in F$. Luego, como sabemos que $w = u \cdot v$ construimos una ejecución de w sobre $\mathcal{A}^{\circlearrowleft}$:

  $$\rho^{\circlearrowleft} : (p_i, p_i, 0) \xrightarrow{a_{i+1}} (p_{i+1}, p_i, 0) \xrightarrow{a_{i+2}} \dots \xrightarrow{a_n} (p_n, p_i, 0) \xrightarrow{a_1} (p_1, p_i, 1) \xrightarrow{a_2} \dots \xrightarrow{a_i} (p_i, p_i, 1)$$

  ya que primero se leen las transiciones de u y después las de v. A partir de esto podemos observar que $(p_i, p_i, 1)$ es un estado final de $\mathcal{A}^{\circlearrowleft}$ por construcción y ya que el autómata partió en el estado $(p_i, p_i, 0)$ y que todas las transiciones corresponden a las del autómata leyendo u, cambiando de autómata y después leyendo v. Por lo tanto $w \in \mathcal{A}^{\circlearrowleft}$ y $L^{\circlearrowleft} \subseteq L(\mathcal{A}^{\circlearrowleft})$.

Por lo anterior $L(\mathcal{A}^{\circlearrowleft}) = L^{\circlearrowleft}$ y $L^{\circlearrowleft}$ es un lenguaje regular.

Problema 2

De expresiones regulares para los siguientes lenguajes

- $L = \{w \in \{0,1\}^* \mid |w| \mod 2 \equiv 0 \}$
- $L = \{w \in \{0,1\}^* \mid w$ no contiene el *substring* $111 \}$
- $L = \{w \in \{0,1\}^* \mid w$ no contiene el *substring* $010\}$

Solución

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

  $$(00 + 01 + 10 + 11)^*$$

  Podemos dividir toda palabra de largo par en subpalabras de largo 2 concatenadas entre si, teniendo solamente 4 formas de armar cada una de estas subpalabras con el alfabeto dado. Por lo anterior, construimos la expresión tomando la unión de las 4 expresiones para cada subpalabra y luego la clausura de Kleene de esta unión para conseguir un largo arbitrario.

  Cabe señalar que la expresión dada acepta la palabra vacía, pues podemos considerarla como una palabra de largo 0, pero si se desea excluirla, bastaría con cambiar el $*$ por un $+$

- $L = \{w \in \{0,1\}^* \mid w$ no contiene el *substring* 111 $\}$

$$(0 + 10 + 110)^* \, 1?1?$$

Las palabras de este lenguaje contienen una cantidad arbitraria de 0s y si es que contienen algún 1 este debe venir seguido o de un 0 o de un 10, pues de lo contrario se formaría el patrón 111, por lo que realizamos la unión entre estas opciones y las repetimos arbitrariamente usando Kleene.

Finalmente, debemos considerar que las palabras del lenguaje también pueden terminar con uno o dos 1s, por lo que concatenamos las últimas expresiones a lo formado.

- $L = \{w \in \{0,1\}^* \mid w$ no contiene el *substring* $010\}$

$$(0^+1)?(1 + 10^+1)^*(10^+)? \, + \, 0^*$$

Las palabras del lengueje pueden contener una cantidad arbitraria de 0s y si encontramos un 1, este debe venir seguido de otro 1 para no formar el patrón. Además, debemos ser capaces de seguir agregando 1s o $0s$ arbirtrariamente, pero, nuevamente, entre cada grupo de 0s debe haber al menos dos 1s. Los dos primeros paréntesis representan esta idea y agregamos un ? al primero para poder comenzar la palabra con 0 o con 1.

La expresión anterior siempre terminan en, al menos, un 1, por lo que en el tercer paréntesis agregamos la posiblidad de terminar con una cantidad arbitraria de 0s (anteponiendo un 1 para no generar el patrón) y, finalmente, unimos la expresión formada con 0^* para poder generar palabras formadas solo con 0s.

Problema 3

a) Encuentra una expresión regular para cada uno de los siguientes lenguajes sobre el alfabeto $\Sigma = \{a, b, c\}$:

- Todos los strings que contienen exactamente dos a's.
- Todos los strings que contienen no más de 3 a's.
- Todos los strings que contienen al menos una ocurrencia de cada símbolo en Σ.

b) Encuentra una expresión regular para el lenguaje $L = \{uwu \mid u, w \in \{a, b\}^* \wedge |u| = 2\}$.

Solución

1. Tenemos que:

   - Una expresión regular para el lenguaje pedido es
   $$(b + c)^* a (b + c)^* a (b + c)^*$$
   - Podemos considerar 4 casos, strings que tengan desde 0 a's hasta strings que tengan 3 a's, es decir:
   $$R_1 = (b + c)^* \qquad R_3 = (b + c)^* a (b + c)^* a (b + c)^*$$
   $$R_2 = (b + c)^* a (b + c)^* \qquad R_4 = (b + c)^* a (b + c)^* a (b + c)^* a (b + c)^*$$
   Por lo tanto, una expresión regular para el lenguaje pedido es
   $$R_1 + R_2 + R_3 + R_4$$
   Otra opción es la expresión regular
   $$(b + c)^* a^? (b + c)^* a^? (b + c)^* a^? (b + c)$$
   - Debemos asegurarnos de que las letras a, b y c aparezcan al menos una vez, pero pueden aparecer en cualquier orden. Por ende, hay 6 posibles ordenes de aparición:

(a) abc
(b) acb
(c) bac
(d) bca
(e) cab
(f) cba

Luego, podemos también tener cualquier combinación de letras (de largo arbitrario) entre las letras de cada combinación. Por tanto, consideramos los siguientes 6 casos, con $R = (a + b + c)^*$:

$$R_1 = R \cdot a \cdot R \cdot b \cdot R \cdot c \cdot R \qquad R_2 = R \cdot a \cdot R \cdot c \cdot R \cdot b \cdot R$$
$$R_3 = R \cdot b \cdot R \cdot a \cdot R \cdot c \cdot R \qquad R_4 = R \cdot b \cdot R \cdot c \cdot R \cdot a \cdot R$$
$$R_5 = R \cdot c \cdot R \cdot a \cdot R \cdot b \cdot R \qquad R_6 = R \cdot c \cdot R \cdot b \cdot R \cdot a \cdot R$$

Así, una expresión regular para el lenguaje pedido es:

$$R_1 + R_2 + R_3 + R_4 + R_5 + R_6$$

2. Como u tiene largo $|u| = 2$, podemos enumerar todas sus posibilidades: aa, ab, ba y bb. Luego, para cada posibilidad de u, los strings de la forma uwu pueden ser expresados como

   (a) $aa(a + b)^*aa$
   (b) $ab(a + b)^*ab$
   (c) $ba(a + b)^*ba$
   (d) $bb(a + b)^*bb$

   Así, una expresión regular R tal que $L = \mathcal{L}(R)$ es

$$(aa(a + b)^*aa) + (ab(a + b)^*ab) + (ba(a + b)^*ba) + (bb(a + b)^*bb)$$

Problema 4 (propuesto)

Sea Σ un alfabeto cualquiera. Para un lenguaje $L \subseteq \Sigma^*$ se define:

$$L - 1 = \{w \in \Sigma^* \mid \exists a \in \Sigma.\ w \cdot a \in L\}$$

Demuestre que para todo lenguaje regular L, el lenguaje $L - 1$ es regular.

Solución

Sabemos que para cualquier lenguaje regular L, existe un autómata $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ tal que $\mathcal{L}(\mathcal{A}) = L$. Ahora, para demostrar que $L-1$ es regular, nos basta con definir $\mathcal{A}^{'} = (Q^{'}, \Sigma, \Delta^{'}, I^{'}, F^{'})$ tal que $\mathcal{L}(\mathcal{A}) = L-1$. Una construcción posible es:

- $Q^{'} = Q$
- $I^{'} = I$
- $\Delta^{'} = \Delta$
- $F^{'} = \{q \in Q \mid \exists a \in \Sigma. \exists p \in F. (q, a, p) \in \Delta\}$

Es decir, los estados finales del nuevo autómata son todos aquellos para los cuales existía una transición hacia un estado final en el autómata original.

$(\mathcal{L}(\mathcal{A}^{'}) \subseteq L - 1)$

Sea $w = a_1 a_2 a_3 \dots a_n \in \mathcal{L}(\mathcal{A}^{'})$ y sea ρ una ejecución sobre w de $\mathcal{L}(\mathcal{A}^{'})$:

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p_n$$

con $p_0 \in I^{'} y p_n \in F^{'}$. Ahora, como $p_n \in F^{'}$ entonces $\exists a \in \Sigma . \exists p \in F . (q, a, p) \in \Delta$ por definición de $\mathcal{A}^{'}$. Finalmente, si construimos $\rho^{'}$ tal que:

$$\rho^{'} : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} p_n \xrightarrow{a} p$$

Notamos que $\rho^{'}$ es una ejecución de aceptación de $\mathcal{A}$ sobre $w \cdot a \in L$. Por lo tanto:

$$w \in L - 1$$

$(L - 1 \subseteq \mathcal{L}(\mathcal{A}^{'}))$

Sea $w = b_1 b_2 b_3 \ldots b_n \in L - 1$. Por definición de $L - 1$ sabemos que $\exists a . w \cdot a \in L$ y por ende $w \cdot a \in \mathcal{L}(\mathcal{A})$ y por lo tanto existe:

$$\rho : p_0 \xrightarrow{b_1} p_1 \xrightarrow{b_2} \ldots \xrightarrow{b_n} p_n \xrightarrow{a} p$$

con $p_0 \in I y p \in F$. Esto implica que:

$$\rho^{'} : p_0 \xrightarrow{b_1} p_1 \xrightarrow{b_2} \ldots \xrightarrow{b_n} p_n$$

es una ejecución de aceptación sobre w de $\mathcal{A}^{'}$ ya que, por construcción, $p_n \in F$. Finalmente, como $(p_n, a, p) \in \Delta$ entonces $(p_n, a, p) \in \Delta^{'}$ y $p_n \in F^{'}$ ya que $p \in F$. Por lo tanto:

$$w \in \mathcal{L}(\mathcal{A}^{'})$$