Autómatas y Lenguajes Formales
Nota 06. Lema del bombeo, teorema de Myhill-Nerode y propiedades de cerradura*

Noé Salomón Hernández S.

1. El lema del bombeo

Sea $L \subseteq \Sigma^*$ un lenguaje regular, entonces existe $n \geq 1$ (la cual depende de L) tal que para toda $w \in L$, $|w| \geq n$, existen x, y y z con $w = xyz$, tales que

a) $y \neq \varepsilon$

b) $|xy| \leq n$

c) $\forall k \geq 0, \quad xy^k z \in L$

Demostración. Se demostrará más adelante. ⊣

Esta es una propiedad de los lenguajes regulares que puede ser usada en su forma contrapositiva para demostrar que un lenguaje dado **no** es regular. Dicha forma contrapositiva se aplica imaginando que se tiene la siguiente interacción en contra de un adversario.

a) El *adversario* piensa en un valor para $n \geq 1$. Pero no sabemos el valor en concreto de n.

b) Elegimos una cadena $w \in L$ con $|w| \geq n$. La elección de w está en términos del parámetro desconocido n.

c) El *adversario* piensa en una descomposición $w = xyz$ con $y \neq \varepsilon$, y $|xy| \leq n$. Pero no nos informa quienes son x, y y z.

d) Ganamos al encontrar $k \geq 0$ tal que $xy^k z \notin L$. Esto lo hacemos sin conocer a x, y y z.

Si logramos ganar el juego anterior, entonces habremos demostrado que L no es regular. Es importante seguir el orden de los pasos en la interacción. También nótese la falta de información acerca de n, x, y y z por lo que ganar este juego es complicado, pero aún así podemos ganar como en el siguiente ejemplo:

*Esta nota se basa en el libro: D. C. Kozen. *Automata and Computability*, Springer-Verlag, Inc., New York, NY, 1997 y en las notas del prof. Rajeev Motwani, aquí las encuentran.

Ejemplo 1.1 El lenguaje $L_{Eq} = \{w \in \{0,1\}^* \mid w \text{ tiene mismo número de 0s y 1s}\}$ no es regular.

Apliquemos la interacción que resulta de la forma contrapositiva del lema del bombeo.

a) El *adversario* piensa en un valor para $n \geq 1$.

b) Nosotros tomamos $w = 0^n 1^n$. Claramente $w \in L_{Eq}$ y $|w| = 2n \geq n$.

c) El *adversario* piensa en una descomposición $w = xyz$ con $y \neq \varepsilon$, y $|xy| \leq n$. Vemos que como $|xy| \leq n$, debe pasar que x y y están constituidos únicamente por 0s. Sean $|x| = i$, $|y| = j > 0$ y $|xy| = i + j \leq n$. De modo que $w = \underbrace{0^i}_{x}\underbrace{0^j}_{y}\underbrace{0^{n-(i+j)}1^n}_{z}$

d) Escogemos $k = 0$. Así

$$\begin{aligned} xy^k z &= xy^0 z \\ &= xz \\ &= 0^i\, 0^{n-(i+j)} 1^n \\ &= 0^{n-j} 1^n \end{aligned}$$

Como $j > 0$ (pues $y \neq \varepsilon$), $xy^k z$ tiene menos 0s que 1s. Por lo tanto, $xy^k z \notin L_{Eq}$. Hemos ganado el juego, esto quiere decir que L_{Eq} no es regular.

Los incisos a) y c) son ajenos a nosotros, ya que representan el turno del adversario. Nosotros actuamos en los incisos b) y d) de manera creativa e ingeniosa para ganar el juego.

Ejemplo 1.2 El lenguaje $L_p = \{a^p \mid p \text{ es un número primo}\}$ no es regular.

Siguiedo la interacción con el adversario que resulta de la forma contrapositiva del lema del bombeo tenemos:

a) El *adversario* piensa en un valor para $n \geq 1$. Desconocemos el valor concreto de n.

b) Elegimos q un número primo tal que $q \geq n + 2$. Como hay un número infinito de números primos, podemos asegurar que $q \geq n + 2$ siempre existe para cualquier n que haya escogido el adversario. Así, tomamos la cadena $w = a^q$. Claramente $w \in L_p$ y $|w| = q \geq n$.

c) El *adversario* piensa en una descomposición $w = xyz$ con $y \neq \varepsilon$, y $|xy| \leq n$. Sean $|x| = i$, $|y| = j > 0$ y $|xy| = i + j \leq n$. Por lo que $|z| = q - i - j$, así $w = \underbrace{a^i}_{x}\underbrace{a^j}_{y}\underbrace{a^{q-i-j}}_{z}$

d) Escogemos $k = q - j$. Así

$$\begin{aligned} |xy^k z| &= |xy^{q-j} z| \\ &= i + j(q-j) + (q-i-j) \\ &= j(q-j) + (q-j) \\ &= (q-j)(j+1) \end{aligned}$$

Veremos que la longitud de la cadena $xy^k z$ es el producto de factores no triviales, es decir, factores que son al menos 2.

- Como $j > 0$, tenemos $j + 1 > 1$. De manera que $\boxed{j+1 \geq 2}$.

- Sabemos que $q \geq n+2$ y también que $j \leq n$, de esto último se sigue que $-j \geq -n$. Sumando $q \geq n+2$ y $-j \geq -n$, llegamos a $\boxed{q - j \geq 2}$.

Luego, la longitud de la cadena xy^kz tiene dos factores no triviales y no puede ser primo. Por lo tanto, $xy^kz \notin L_p$. Hemos ganado el juego, esto quiere decir que L_p no es regular.

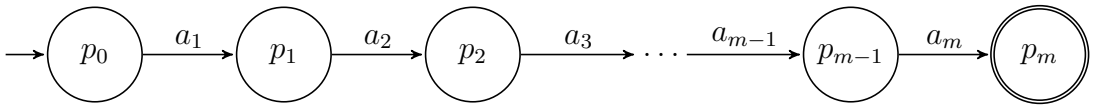
A continuación se presenta la demostración del lema del bombeo.

Demostración. Sea L un lenguaje regular y $M = (Q, \Sigma, \delta, q_0, F)$ el AFD mínimo para L. Así que tomamos $n = |Q|$, como q_0 siempre forma parte de Q, podemos garantizar que $n = |Q| \geq 1$. Ahora, consideramos $w \in L$ con $|w| = m \geq n$. Como $w \in L(M)$, $\widehat{\delta}(q_0, w)$ en M define la trayectoria de ejecución para $w = a_1 a_2 \dots a_m$.

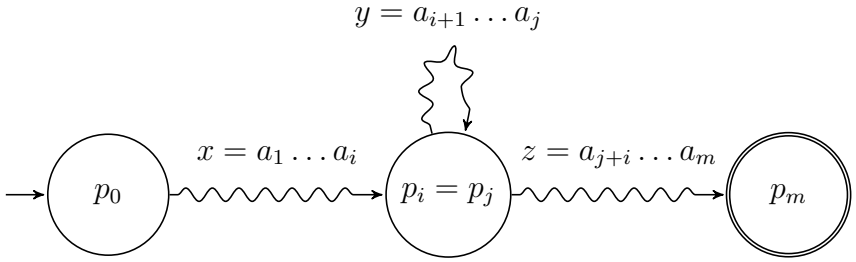
Denotamos a p_ℓ como el estado que se alcanza al procesar la subcadena $a_1 a_2 \dots a_\ell$ en M con $0 \leq \ell \leq m$, es decir,

$$p_\ell = \widehat{\delta}(q_0, a_1 a_2 \dots a_\ell).$$

Al definir $p_0 = q_0$, tenemos abajo la trayectoria de ejecución dada por $\widehat{\delta}(q_0, w)$,



De este modo, con la cadena w el AFD visita $m+1$ estados. Como $m+1 > n$ y cada p_ℓ pertenece a Q, el cual únicamente tiene n estados, se sigue por *el principio de las casillas* que $p_0, p_1, \dots, p_{m-1}, p_m$ **no** son todos distintos. Sean $0 \leq i < j \leq n$ índices tales que $p_i = p_j$. Por lo que el siguiente ciclo está presente en la trayectoria de ejecución,



Observemos que,

a) $|y| = j - i > 0$, pues $i < j$.

b) $|xy| = j \leq n$, ya que elegimos $i < j \leq n$.

c) $\forall k \geq 0,\ xy^kz \in L(M)$, ya que el ciclo de arriba se puede repetir k veces, culminando la ejecución en el estado final p_m. Esto se conoce como el ciclo del bombeo. ⊣

2. Teorema de Myhill-Nerode

Definición 2.1 Sea $R \subseteq \Sigma^*$ un lenguaje regular. Una relación Myhill-Nerode para R es una relación de equivalencia $\equiv$ sobre Σ^* que satisface las siguientes tres propiedades:

(I) $\equiv$ es de *congruencia derecha*: para toda $x, y \in \Sigma^*$ y $a \in \Sigma$,

$$x \equiv y \Rightarrow xa \equiv ya;$$

(II) $\equiv$ *redefine* R: para toda $x, y \in \Sigma^*$,

$$x \equiv y \Rightarrow (x \in R \Leftrightarrow y \in R);$$

(III) $\equiv$ es de *índice finito*, es decir, $\equiv$ tiene un número finito de clases de equivalencia.

Definición 2.2 Sea $R \subseteq \Sigma^*$ un lenguaje regular o no, y $x, y \in \Sigma^*$. Definimos una relación de equivalencia $\equiv_R$ sobre Σ^* en términos de R como sigue

$$x \equiv_R y \overset{\text{def}}{\Longleftrightarrow} \forall z \in \Sigma^*,\ (xz \in R \Leftrightarrow yz \in R).$$

En otras palabras, dos cadenas son equivalentes bajo $\equiv_R$ si, al concatenar cualquier cadena a la derecha de ambas, se obtienen dos cadenas tales que ambas están en R, o bien, ambas no están en R.

Teorema 2.3 (Myhill-Nerode) *Sea $R \subseteq \Sigma^*$. Las siguientes afirmaciones son equivalentes:*

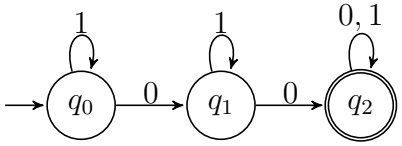
(a) R es regular;

(b) existe una relación de Myhill-Nerode para R;

(c) la relación $\equiv_R$ es de índice finito, es decir, $\equiv_R$ tiene un número finito de clases de equivalencia.

Demostración. Se omite pero puede ser encontrada en el libro *Automata and Computability* por D. C. Kozen. ⊣

Ejemplo 2.1 Encuentre las clases de equivalencia inducias por $\equiv_L$ para el lenguaje $L = L(M)$, donde M es el autómata siguiente.



De manera que $L = \{w \in \{0,1\}^* \mid w \text{ tiene al menos dos 0's}\}$. Las clases de equivalencia para $\equiv_L$ son:

- $[\varepsilon] = \{1^*\}$, en palabras esta es la clase de equivalencia de cadenas binarias sin 0's. En esta clase de equivalencia están ε, 1, 1111, ... ¿Qué $z \in \{0,1\}^*$ le ponemos concatenar a la derecha de las cadenas en esta clase de equivalencia para que resulten elementos de L? Únicamente, cadenas z de la forma $1^*01^*0(0+1)^*$.

- $[0] = \{1^*01^*\}$, en palabras esta es la clase de equivalencia de cadenas binarias con un 0. En esta clase de equivalencia están 0, 10, 111011, ... ¿Qué $z \in \{0,1\}^*$ le ponemos concatenar a la derecha de las cadenas en esta clase de equivalencia para que resulten elementos de L? Únicamente, cadenas z de la forma $1^*0(0+1)^*$.
- $[00] = \{1^*01^*0(0+1)^*\}$, en palabras esta es la clase de equivalencia de cadenas binarias con dos o más 0's. En esta clase de equivalencia están 00, 1010, 0010, ... ¿Qué $z \in \{0,1\}^*$ le ponemos concatenar a la derecha de las cadenas en esta clase de equivalencia para que resulten elementos de L? Cualquier z de la forma $(0+1)^*$.

Las clases de equivalencia son mutuamente excluyentes. Por ejemplo, $1111 \not\equiv_L 111011$ ya que al concatenar ambas cadenas con 0 a la derecha tenemos que $11110 \notin L$, mientras que $1110110 \in L$.

2.1. Una aplicación

El teorema de Myhill-Nerode puede usarse para determinar si un lenguaje R es regular o no, al encontrar el número de clases de equivalencia de $\equiv_R$.

Ejemplo 2.2 Tomemos el lenguaje

$$A = \{a^n b^n \mid n \geq 0\}$$

Si $k \neq m$, entonces $a^k \not\equiv_A a^m$, ya que al tomar b^k se tiene que $a^k b^k \in A$ pero $a^m b^k \notin A$. Por lo tanto, hay una infinidad de clases de equivalencia de $\equiv_A$, al menos una para cada a^k, $k \geq 0$. Por el teorema de Myhill-Nerode, A no es regular.

De hecho, uno puede mostrar que las clases de equivalencia de $\equiv_A$ son:

$$\begin{aligned} G_k &= \{a^k\}, \quad k \geq 0, \\ H_k &= \{a^{n+k} b^n \mid n \geq 1\}, \quad k \geq 0, \\ E &= \Sigma^* - \bigcup_{k \geq 0} G_k \cup H_k = \Sigma^* - \{a^m b^n \mid 0 \leq n \leq m\}. \end{aligned}$$

Para cadenas en G_k, únicamente los elementos de $\{a^n b^{n+k} \mid n \geq 0\}$ pueden ser concatenados para obtener cadenas de A; para las cadenas de H_k, únicamente las cadenas de la forma b^k pueden ser concatenados para obtener cadenas de A; y no hay cadena que pueda ser concatenada a elementos de E de manera que resulte una cadena de A.

Ejemplo 2.3 Tomemos el lenguaje

$$L_{Sq} = \{a^n \mid n \text{ es un cuadrado perfecto}\}$$

Supongamos que tenemos dos naturales i y j, tales que $i \neq j$. Sin pérdida de generalidad supongamos $0 \leq j < i$. Queremos ver que $a^i \not\equiv_{L_{Sq}} a^j$, por lo que hay que encontrar un entero k de modo que $a^i a^k \in L_{Sq}$ y $a^j a^k \notin L_{Sq}$. Sea $k = (i+1)^2 - i = i^2 + i + 1$. Entonces $i + k = \not{i} + (i+1)^2 - \not{i} = (i+1)^2$ pero $j + k = i^2 + i + j + 1$, así $i^2 < j + k < (i+1)^2$ ya que $j < i$. Como $j + k$ cae entre i^2 y $(i+1)^2$, $j + k$ no puede ser un cuadrado perfecto. Eso indica que $a^i a^k \in L_{Sq}$ y $a^j a^k \notin L_{sq}$. Como lo anterior ocurre para cualesquiera i y j distintos, hay una infinidad de clases de equivalencia de $\equiv_{L_{Sq}}$. Por el teorema de Myhill-Nerode, L_{Sq} no es regular.

Ejemplo 2.4 Tomemos al lenguaje

$$L_D = \{w \in \{a,b\}^* \mid n_a(w) < 2\, n_b(w)\}$$

donde la función $n_\sigma(w)$ calcula el número de veces que el símbolo $\sigma \in \{a,b\}$ figura en la cadena w. Sean k y m dos naturales distintos con $k < m$, así $k+1 \leq m$ y

$$2(k+1) \leq 2m \qquad (\clubsuit)$$

Ahora $a^{2k} \not\equiv_{L_D} a^{2m}$, puesto que al considerar b^{k+1} se tiene que

- $a^{2k}b^{k+1} \in L_D$ porque $n_a(a^{2k}b^{k+1}) = 2k < 2(k+1) = 2n_b(a^{2k}b^{k+1})$.
- Pero, $a^{2m}b^{k+1} \notin L_D$ porque $n_a(a^{2m}b^{k+1}) = 2m \geq 2(k+1) = 2n_b(a^{2k}b^{k+1})$ como dice la ecuación $\clubsuit$.

Efectivamente, $a^{2k} \not\equiv_{L_D} a^{2m}$ para cualesquiera k y m distintos, debido a que uno debe ser mayor que el otro. Por lo tanto, hay una infinidad de clases de equivalencia. Por el teorema de Myhill-Nerode, L_D no es regular.

3. Propiedades de cerradura

Teorema 3.1 *Si L_1 y L_2 son lenguajes regulares, entonces también lo son $L_1 \cup L_2$, $L_1 \cdot L_2$ y L_1^*.*

Demostración. Al ser L_1 y L_2 lenguajes regulares, entonces deben haber expresiones regulares R_1 y R_2 tales que,

$$L_1 = L(R_1),$$
$$L_2 = L(R_2).$$

Entonces,

- $L_1 \cup L_2 = L(R_1) \cup L(R_2) = L(R_1 + R_2)$,
- $L_1 \cdot L_2 = L(R_1) \cdot L(R_2) = L(R_1 \cdot R_2)$,
- $L_1^* = L(R_1)^* = L(R_1^*)$.

Por lo tanto, $L_1 \cup L_2$, $L_1 \cdot L_2$ y L_1^* deben ser regulares ya que son generados por una expresión regular.

Así, la clase de lenguajes regulares es cerrada bajo las operaciones de unión, concatenación y estrella de Kleene, es decir, no se pueden generar lenguajes fuera de esta clase al aplicar dichas operaciones a los lenguajes que ya forman parte de los lenguajes regulares. ⊣

3.1. Complemento

Sea Σ un alfabeto. El complemento del lenguaje L es $\overline{L} = \Sigma^* - L$.

Teorema 3.2 *Los lenguajes regulares son cerrados bajo la operación de complemento.*

Demostración. Consideremos el lenguaje regular L que es reconocido por el AFD $M = (Q, \Sigma, \delta, q_0, F)$. Construimos ahora el AFD $\overline{M} = (Q, \Sigma, \delta, q_0, Q - F)$. Claramente, $L(\overline{M}) = \overline{L(M)} = \overline{L}$. Por lo tanto, $\overline{L}$ es regular al tener un AFD que lo reconoce. ⊣

3.2. Intersección

Teorema 3.3 *Sean L_1 y L_2 lenguajes regulares, entonces $L_1 \cap L_2$ es regular.*

Demostración. Por las leyes de De Morgan conocemos que $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. Por resultados previos sabemos que los lenguajes regulares son cerrados bajo la unión y el complemento. Por lo tanto, $L_1 \cap L_2$ es regular. ⊣

3.3. Reversa

La reversa de una cadena se define como se muestra a continuación:

$$\begin{aligned} \varepsilon^{\mathcal{R}} &= \varepsilon, \\ w^{\mathcal{R}} &= (a_1 a_2 \dots a_{n-1} a_n)^{\mathcal{R}} \\ &= a_n a_{n-1} \dots a_2 a_1. \end{aligned}$$

De manera que la reversa de una lenguaje L es $L^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in L\}$.

Teorema 3.4 *Los lenguajes regulares son cerrados bajo la operación de reversa.*

Demostración. Vamos a extender la definición de reversa a expresiones regulares inductivamente,

$$\textit{Base} \begin{cases} \varepsilon^{\mathcal{R}} = \varepsilon \\ \varnothing^{\mathcal{R}} = \varnothing \\ a^{\mathcal{R}} = a \qquad \forall a \in \Sigma \end{cases}$$

Para el paso inductivo suponemos que E_1 y E_2 son expresiones regulares. Así,

$$\textit{Paso inductivo} \begin{cases} (E_1 + E_2)^{\mathcal{R}} = E_1^{\mathcal{R}} + E_2^{\mathcal{R}} \\ (E_1 \cdot E_2)^{\mathcal{R}} = E_2^{\mathcal{R}} \cdot E_1^{\mathcal{R}} \\ (E_1^*)^{\mathcal{R}} = (E_1^{\mathcal{R}})^* \end{cases}$$

Se deja como ejercicio al lector que verifique por inducción que $L(E^{\mathcal{R}}) = L(E)^{\mathcal{R}}$. ⊣

Aplicando como ejemplo la definición anterior a la expresión regular $E = abc + bc^*a$ tenemos que $E^{\mathcal{R}} = cba + ac^*b$.

4. Lenguajes no regulares

A continuación se da una lista de lenguajes no regulares, i.e., que no son aceptados por autómatas finitos.

- $\{w \in \{0,1\}^* \mid w$ tiene mismo número de 0s y 1s$\}$
- $\{a^n b^n \mid n \geq 0\}$; similarmente, $\{0^n 1^n \mid n \geq 0\}$
- $\{a^n \mid n$ es un cuadrado perfecto$\}$

- $\{w \in \{a,b\}^* \mid n_a(w) < 2\, n_b(w)\}$
- $\{a^n \mid n \text{ es número primo}\}$
- $\{a^{n!} \mid n \geq 0\}$
- $\{a^n b^m \mid n \geq m\}$
- $\{ww \mid w \in \{0,1\}^*\}$
- $\{0^i 1^j \mid i > j\}$
- $\{w \in \{0,1\}^* \mid w^{\mathcal{R}} = w$, es decir, w es un palíndromo$\}$