Lema del Bombeo Hemos visto qué son los lenguajes regulares (un lenguaje L para el cual existe un DFA A tal que $L = L(A)$), ejemplos de lenguajes regulares, pero todavía no vemos un ejemplo de un lenguaje que no sea regular. En estos apuntes veremos el Lema del Bombeo que nos permitirá discernir cuando un lenguaje es regular y cuando no.

Teorema 2.12 (Lema del Bombeo para lenguajes regulares)**.** *Sea L un lenguaje regular. Entonces, existe una constante n (que depende de L) tal que para toda palabra $w \in L$ con $|w| \geq n$, podemos separar w en tres palabras x, y, z de la siguiente forma, $w = xyz$, que cumplen:*

- $y \neq \epsilon$,
- $|xy| \leq n$,
- *Para todo entero $k \geq 0$, la palabra xy^kz pertenece a L.*

Proof. Supongamos que L es un lenguaje regular. Luego existe un DFA A tal que $L(A) = L$. Supongamos que A tiene n estados. Considere una palabra w de largo mayor o igual a n, digamos $w = a_1a_2 \cdots a_m$, donde $m \geq n$ y cada a_i es un símbolo del alfabeto de A. Para cada $i \in \{0, 1, 2, \ldots, n\}$ definimos $p_i := \hat{\delta}(q_0, a_1a_2 \cdots a_i)$ (donde $\hat{\delta}$ es la función de transición extendida de A, y q_0 es el estado inicial de A), es decir, p_i es el estado en que se encuentra A luego de procesar los primeros i símbilos de w. Notemos que $p_0 = q_0$.

Se definieron $m+1$ estados p_i. Ahora, dado que A tiene n estados y usando el Principio del Palomar, existen $i \neq j$ tal que $p_i = p_j$. Sin pérdida de generalidad supongamos que $0 \leq i < j \leq n$. Ahora, separemos w como sigue, $w = xyz$ tal que:

- $x = a_1a_2 \cdots a_i$,
- $y = a_{i+1}a_{i+1} \cdots a_j$,
- $z = a_{j+1}a_{j+2} \cdots a_m$.

De esta forma, cuando A procesa a x, éste termina en el estado p_i. Luego, al procesar a y, A pasa del estado p_i al estado p_i (ya que $p_j = p_i$). Finalmente al procesar a z, A pasa de p_i al estado en que termina A al procesar a w. Ahora, como $i < j$, entonces y no puede ser la palabra de largo 0. Por otro lado, por definición de x e y, tenemos que $|xy| \leq n$, ya que la repetición de estados ocurre al procesar las primeras n letras de w.

Ahora consideremos qué pasa cuando el autómata procesa una palabra xy^kz. El DFA A recibe la palabra xy^kz para algún $k \geq 0$; A pasa del estado inicial al estado p_i al procesar x; luego, A retorna al estado p_i k veces al procesar y^k; finalmente, A pasa del estado p_i al mismo estado en donde termina cuando procesa w al procesar z. Entonces podemos concluir que A acepta xy^kz si y sólo si A acepta w. Como partimos de la base que $w \in L$, entonces $xy^kz \in L$. □

La principal utilidad del Lema del Bombeo es ayudarnos a determinar cuando un lenguaje no es regular. Ahora veremos un par de ejemplos.

Considere el lenguaje L_{eq} que contiene todas las palabras con el mismo número de 0's y de 1's. Veremos que L_{eq} no es regular. Demostraremos esto por contradicción. Supongamos que L_{eq} es regular, por lo tanto el Lema del Bombeo se debe cumplir. Supongamos que n es la constante que existe, según el Lema del Bombeo, para L_{eq}. Tomemos entonces la palabra $w = 0^n 1^n$, consistente en la concatenación de n ceros seguidos de n unos. Esta palabra está en L_{eq}. Ahora, podemos separar w como la concatenación de tres palabras xyz tal que $|y| \neq 0$, y $|xy| \leq n$. Como $|xy| \leq n$, entonces la palabra xy consta solamente de 0's. Ahora, el Lema del Bombeo nos dice que $xz \in L_{eq}$, sin embargo xz tiene n unos y a lo más $n-1$ ceros ya que y tiene al menos un cero que no está en xz. Lo que no se condice con la definición de L_{eq}, por lo tanto tenemos una contradicción. De esta forma L_{eq} no puede ser regular.

Para el segundo ejemplo, consideremos el lenguaje L_{prm}, definido como sigue:

$$L_{prm} = \{1^p \colon p \text{ es un número primo}\}.$$

Usando el Lema del Bombeo veremos que L_{prm} no es regular. Lo haremos por contradicción, por lo tanto suponemos que sí es regular. De esta forma existe una constante n que satisface las condiciones del Lema del Bombeo. Consideremos un número primo $p \geq n+2$, y sea $w = 1^p$. Por el Lema del Bombeo podemos separar w en tres palabras de la forma $w = xyz$ tal que $y \neq \epsilon$ y $|xy| \leq n$. Sea $|y| = m > 0$, luego $|xz| = p - m$. Ahora, consideremos la palabra $xy^{p-m}z$, que por el Lema del Bombeo debe pertenecer a L_{prm}. Sin embargo,

$$|xy^{p-m}z| = |xz| + (p-m)|y| = p - m + (p-m)m = (p-m)(m+1).$$

Como $m > 0$, entonces $m+1 > 1$. Como $p \geq n+2$ y $m \leq n$, tenemos que $p - m > 1$. Luego, $(p-m)(m+1)$ no es un número primo y tenemos una contradicción. Lo que nos permite concluir que L_{prm} no puede ser regular.

Propiedades de los Lenguajes Regulares Ahora veremos que el conjunto de todos los lenguajes regulares es cerrado bajo ciertas operaciones. Consideraremos operaciones bien definidas para conjuntos como *unión*, *intersección* y *complemento*, además de operaciones que aplican a lenguajes, como son la *concatenación* y la *estrella de Kleene*.

Sea L y M dos lenguajes definidos sobre los alfabetos Σ_L y Σ_M, respectivamente. Entonces definimos las siguientes operaciones.

- Unión: $L \cup M := \{w \colon\ w \in L \vee w \in M\}$.
- Intersección: $L \cap M := \{w \colon\ w \in L \wedge w \in M\}$.
- Resta de conjuntos $L \setminus M := \{w \colon\ w \in L \wedge w \notin M\}$.
- Complemento: $\bar{L} := \{w \in \Sigma_L^* \colon\ w \notin L\} = \Sigma_L^* \setminus L$.
- Concatenación $LM := \{w = xy \colon\ x \in L \wedge y \in M\}$.

- Estrella de Kleene: $L^* := \bigcup_{n \in \mathbb{N} \cup \{0\}} L^n$, (donde $L^0 = \{\epsilon\}$)

Lema 2.13. *Sea L un lenguaje regular, entonces el lenguaje $\bar{L}$ es regular.*

Proof. Para demostrar que $\bar{L}$ es un lenguaje regular, tenemos que dar un DFA cuyo lenguaje sea $\bar{L}$. Como L es regular, existe un DFA $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L(A) = L$. Consideremos ahora el DFA $\bar{A} = (Q, \Sigma, \delta, q_0, Q \setminus F)$. Veremos que $L(\bar{A}) = \bar{L}$. Para esto primero consideremos una palabra $w \in \bar{L}$. Como w no pertenece a L cuando A procesa a w termina en un estado que no pertenece a F, luego termina en un estado que pertenece a $Q \setminus F$. Como $\bar{A}$ hace exactamente lo mismo que A, ya que la única diferencia entre estos dos DFA es el conjunto de estados de aceptación, entonces cuando $\bar{A}$ procesa a w termina en un estado en $Q \setminus F$. Luego $\bar{A}$ acepta a w, y de esta forma $w \in L(A)$.

Ahora, tomemos una palabra $w \in L(A)$. Entonces, al ser procesada por $\bar{A}$ (y por A) termina en un estado en $Q \setminus F$. Luego, A rechaza w. De esta forma $w \in \bar{L}$. □

Lema 2.14. *Sean L y M dos lenguajes regulares. Entonces $L \cup M$ es regular.*

Proof. Como L y M son dos lenguajes regulares, existen DFA cuyos lenguajes son L y M. Sea $A = (Q_A, \Sigma, \delta_A, q_0^A, F_A)$ un DFA que reconoce a L y $B = (Q_B, \Sigma, \delta_B, q_0^B, F_B)$ un DFA que reconoce a M.

A partir de estos dos DFA definimos el DFA $C = (Q, \Sigma, \delta, q_0, F)$ como sigue:

- $Q = Q_A \times Q_B$
- $\delta : Q_A \times Q_B \times \Sigma \to Q_A \times Q_B$, donde $\delta((q_A, q_B), a) = (\delta_A(q_A, a), \delta_B(q_B, a))$.
- $q_0 = (q_0^A, q_0^B)$
- $F = \{(q_A, q_B) : \ q_A \in F_A \lor q_B \in F_B\} = F_A \times Q_B \cup Q_A \times F_B$.

El DFA C *simula* una ejecución en paralelo de A y B, y llega a un estado de aceptación cuando uno de los dos autómatas llega a un estado de aceptación. Por lo tanto $L(C) = L \cup M$.

Sea $w \in L \cup M$. Entonces $w \in L$ o $w \in M$ (o ambas). Sin pérdida de generalidad, supongamos que $w \in L$. Entonces cuando A procesa a w termina en un estado en F_A. De esta forma, cuando C procesa a w termina en un estado en $F_A \times Q_B$. Luego, C acepta a w, o dicho de otra forma $w \in L(C)$.

Ahora, considere $w \in L(C)$. Luego, cuando C procesa a w termina en un estado de aceptación. Es decir, termina en un estado que pertenece a $F_A \times Q_B$ o $Q_A \times F_B$. Luego A o B acepta a w (o ambas). De esta forma $w \in L \cup M$. □

Corolario 6. *La intersección y la resta de lenguajes regulares da como resultado un lenguaje regular.*

Proof. Para ver que la intersección de dos lenguajes regulares da como resultado un lenguaje regular, basta con ver que

$$L \cap M = \overline{\bar{L} \cup \bar{M}},$$

y usar que la unión y el complemento de lenguajes regulares da como resultado un lenguaje regular.

Para ver que la resta de lenguajes regulares da como resultado un lenguaje regular basta con ver que $L \setminus M = L \cap \bar{M}$, y usar que el complemento y la intersección de lenguajes regulares da como resultado un lenguaje regular. $\square$