4. Lenguajes Regulares.

Los autómatas finitos (AFD, AFND y AFND-λ) reconocen lenguajes regulares.
Las expresiones regulares describen los mismos lenguajes que reconocen los distintos tipos de autómatas: los lenguajes regulares.
Las gramáticas regulares generan lenguajes regulares.

¿Cómo demostrar que un Lenguaje es Regular?

Se puede demostrar que un Lenguaje es Regular si se encuentra un autómata finito (AFD, AFND o AFND-λ) que lo reconozca, o bien una expresion regular que lo describa o bien una gramática regular que lo genere.

Conversión de un Autómata Finito en una Gramática Regular.

Teorema (AF →GR).

Si L es un lenguaje aceptado por un autómata finito M, entonces existe una gramática regular G tal que $L = L(M) = L(G)$

Algoritmo:

Entrada: $M = \langle Q, \Sigma, q_0, F, \delta \rangle$

Salida: $G = (Q, \Sigma, q_0, P)$

1. El conjunto de símbolos no terminales de G es Q.
2. El conjunto de símbolos terminales de G es Σ.
3. El símbolo inicial de la gramática es el estado inicial del autómata.
4. Si existe $\delta(q, a) = p$, añadir $q \to ap$ al conjunto de producciones P.
5. Si $q_f \in F$, añadir $q_f \to \lambda$ al conjunto de producciones P.

Teorema (GR →AF).

Si L es un lenguaje generado por una gramática regular, entonces existe un autómata finito M, tal que $L = L(M) = L(G)$

Algoritmo:

Entrada: $G = (V, \Sigma, S, P)$

Salida: $M = \langle V \cup \{q_f\}, \Sigma, S, \{q_f\}, \delta \rangle$

1. El conjunto de estados de M es el conjunto de símbolos no terminales de G unión un estado q_f.
2. El conjunto de símbolos terminales de M es Σ.
3. El estado inicial del autómata es el símbolo inicial de la gramática.
4. El estado final del autómata es el conjunto formado por el estado q_f.
5. Si existe la regla $A \to aB$ en el conjunto de producciones P, agregar $\delta(A, a) = B$.
6. Si existe la regla $A \to a$ en el conjunto de producciones P, agregar $\delta(A, a) = \{q_f\}$.
7. Si existe la regla $A \to \lambda$ en el conjunto de producciones P, agregar $\delta(A, \lambda) = \{q_f\}$.

¿Cómo demostrar que un Lenguaje NO es Regular?.

Todo lenguaje ***finito*** es regular.
Si un lenguaje es ***infinito***, para demostrar que no es regular, se usa el **lema de bombeo**.

Lema del bombeo para Lenguajes Regulares.

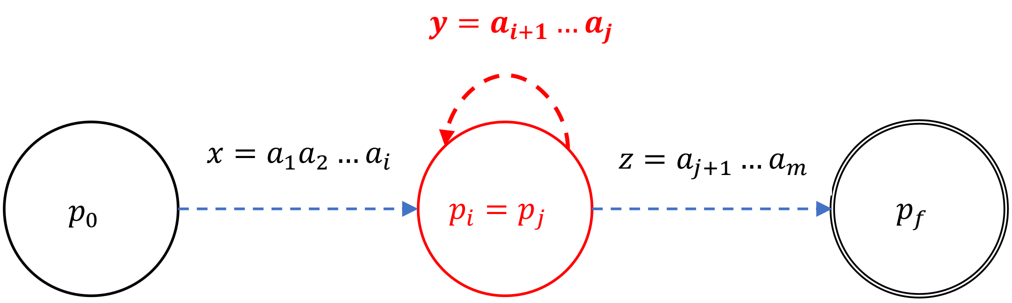
Si L es un **lenguaje regular infinito**
Entonces:
$\exists n / \forall \omega \in L, : |\omega| \geq n \Rightarrow$ podemos dividir ω en 3 cadenas, $\omega = xyz$ de modo que:
1. $y \neq \lambda$
2. $|xy| \leq n$
3. $\forall k \geq 0 : xy^k z \in L$

O sea: siempre es posible encontrar una cadena no vacía $y \neq \lambda$, no demasiado lejos del inicio de ω, que se puede "bombear". Es decir, que si se repite cualquier número de veces o se borra, la cadena sigue perteneciendo al lenguaje.

Demostración

Suponemos que L es regular, entonces $L = L(A)$ para algún AFD $A =< Q, \Sigma, q_0, \delta, F >$
Ese autómata podría tener, por ejemplo, n estados: $\#Q = n$.
Consideremos cualquier cadena ω de longitud n o mayor, por ejemplo:
$\omega = a_1 a_2 a_3 \dots a_m \;\; (m \geq n)$ con $a_i \in \Sigma \forall i$
Para $i = 0,1,2, \dots, n$ definimos el estado p_i como $\hat{\delta}(q_0, a_1 a_2 \dots a_i) = p_i$ (Estado donde se encuentra el autómata después de leer los primeros i símbolos de ω.
Nótese que $p_0 = q_0$.
No es posible que todos los $(n+1) p_i$ para $i = 0,1,2, \dots, n$ sean diferentes, pues sólo hay n estados distintos.
Por lo tanto, podemos determinar dos enteros distintos i y j, con $0 \leq i < j \leq n$, tales que $p_i = p_j$.
Podemos entonces descomponer $\omega = xyz$ como sigue:

1. $x = a_1 a_2 \dots a_i$
2. $y = a_{i+1} a_{i+2} \dots a_j$
3. $z = a_{j+1} \dots a_m$



Es decir, $\hat{\delta}(q_0, x) = p_i$, $\hat{\delta}(p_i, y) = p_i$ y z es el resto de la cadena ω.
Se observa que x puede estar vacía si $i = 0$.
También z puede estar vacía si $j = m = n$.
Pero y no puede estar vacía ya que $i < j$.
¿qué ocurre si el autómata recibe $xy^k z$ para distintos valores de k?

- Si $\boxed{k = 0}$, el autómata recibe la entrada $xy^0 z = xz = a_1 \dots a_i a_{j+1} \dots a_m$
  Como, $\hat{\delta}(q_0, xz) = \hat{\delta}(p_i, z) = p_f$, entonces $xy^0 z$ es aceptada por el autómata.
- Si $\boxed{k > 0}$, el autómata recibe la entrada:
  $xy^k z = xy \dots yz = a_1 \dots a_i a_{i+1} \dots a_j a_{i+1} \dots a_j \dots \; a_{i+1} \dots a_j \; a_{j+1} \dots a_m$
  Como, $\hat{\delta}(q_0, xy \dots yz) = \hat{\delta}(p_i, y \dots yz) = \dots = \hat{\delta}(p_i, \dots yz) = \hat{\delta}(p_i = p_j, z) = p_f$ entonces $xy^k z$ es aceptada por el autómata.
  Por lo tanto, $\forall k \geq 0 : xy^k z \in L(A)$.

Uso del Lema del Bombeo para demostrar que un Lenguaje no es Regular.

El lema permite demostrar por contradicción si un lenguaje dado NO es regular.
Asumiendo que sí lo fuera, debe cumplirse.

Por lo tanto, el desafío es contradecir el enunciado.

Como el enunciado dice:	Hay que encontrar
Existe un n, tal que <u>para toda palabra</u> de longitud mayor o igual que n,	***<u>UNA palabra</u>*** de longitud mayor o igual que n que no lo cumpla.
<u>existe una</u> partición $\omega = xyz$	Es decir, que pudiendo ser expresada como $\omega = xyz$, de ***<u>cualquier forma que esa partición</u>*** se pueda hacer,
1. $y \neq \lambda$ 2. $\lvert xy\rvert \leq n$ 3. $xy^kz \in L\ \forall k \geq 0$	1. $y = \lambda$ ***o bien***: 2. $\lvert xy\rvert > n$ ***o bien***: 3. $\exists k \geq 0 \mid xy^kz \notin L$

Ejemplo:

Demostrar que $L = \{a^m b^{2m} / m \in N, m \geq 0\}$

1. suponemos que L es regular
2. sea $n = N$ la constante del lema
3. seleccionamos la palabra $\omega = a^N b^{2N} \in L$, con $|\omega| = 3N \geq N$
4. descomponemos $\omega = xyz$ <u>de todas las formas posibles</u>, <u>tales que $|xy| \leq N$</u> y $y \neq \lambda$.
   Todas las descomposiciones tienen la forma $\omega = a^{|x|}a^{|y|}a^{N-|xy|}b^{2N}$
   Donde: $x = a^{|x|}$, $y = a^{|y|}$, $z = a^{N-|xy|}b^{2N}$
5. consideramos la palabra xy^0z de estas descomposiciones: $xy^0z = xz = a^{|x|}.a^{N-|xy|}.b^{2N} = a^{N-|y|}.b^{2N}$. Esta palabra o bien tiene $|y| = 0$ y sería entonces $y = \lambda$ o bien no pertenece al lenguaje. De cualquiera de esas dos formas contradice el lema de bombeo.
6. Por lo tanto, L no es regular.

Propiedades de los lenguajes regulares.

Unión.

Dados dos lenguajes regulares, su **UNIÓN** también es un lenguaje regular.

Si L y M son lenguajes regulares, entonces también lo es $L \cup M$.

Demostración

Dado que L y M son lenguajes regulares, pueden representarse mediante dos expresiones regulares r y s, es decir $L = L(r)$ y $M = L(s)$.
Entonces, $L(r) \cup L(s) = L(r+s) = L \cup M$, por definición del operador + de expresiones regulares.
Por lo tanto $L \cup M$ también es regular.

Concatenación.

Dados dos lenguajes regulares, su **CONCATENACIÓN** también es un lenguaje regular.

Si L y M son lenguajes regulares, entonces también lo es $L.M$.

Demostración

Similar a la de la unión, usando expresiones regulares.

Clausura.

Dado un lenguaje regular, su **CLAUSURA** también es un lenguaje regular.

Si L es un lenguaje regular, entonces también lo es L^*.

Demostración

Similar a la de la unión, usando expresiones regulares.

Complemento.

Dado un lenguaje regular, su **COMPLEMENTO** también es un lenguaje regular.

Si L es un lenguaje regular, entonces también lo es $\bar{L}$.

Demostración

Si L es un lenguaje regular, con el alfabeto Σ, entonces $\overline{L} = \Sigma^* - L$ y existe un AFD $A =< Q, \Sigma, \delta, q_0, F >$ tal que $L(A) = L$.
Entonces, existe $B =< Q, \Sigma, \delta, q_0, Q - F >$
Como todas las transiciones de A son las mismas que B, las cadenas siguen los mismos caminos, con la diferencia de que $\omega \in L(B) \Leftrightarrow \hat{\delta}(q_0, \omega) \in (Q - F)$, lo que equivale a decir que $\omega \in L(B) \Leftrightarrow \omega \notin L(A)$.
Entonces, al ser $L(B) = \overline{L}$, $\overline{L}$ es regular.

Intersección.

Dados dos lenguajes regulares, su **INTERSECCIÓN** también es un lenguaje regular.

Si L y M son lenguajes regulares, entonces también lo es $L \cap M$.

Dado que ya se demostró que tanto la unión como el complemento de lenguajes regulares es regular, la intersección debe ser regular por cumplirse $L \cap M = \overline{\overline{L} \cup \overline{M}}$.
Pero también existe una forma directa de obtener la intersección:

Demostración

Sea L un lenguaje regular, reconocido por el AFD $A_L =< Q_L, \Sigma, \delta_L, F_L >$ y M un lenguaje regular, reconocido por el AFD $A_M =< Q_M, \Sigma, \delta_M, F_M >$
Se define el autómata $A =< Q_L \times Q_M, \Sigma, \delta, (q_L, q_M), F_L \times F_M >$ (Donde $\times$ es el producto cartesiano) con la función de transición definida de la siguiente manera:

$$\delta\big((p, q), a\big) = (\delta_L(p, a), \delta_M(q, a))$$

Para ver que $L(A) = L(A_L) \cap L(A_M)$, $\forall\, \omega \in \Sigma^*: \omega \in L(A) \Leftrightarrow \hat{\delta}\big((q_L, q_M), \omega\big) \in F_L \times F_M$
Pero eso equivale a decir que $\hat{\delta}\big((q_L, q_M), \omega\big) = \Big(\hat{\delta}_L(q_L, \omega), \hat{\delta}_M(q_M, \omega)\Big) \in F_L \times F_M$
Cosa que ocurre si $\hat{\delta}_L(q_L, \omega) \in F_L \wedge \hat{\delta}_M(q_M, \omega) \in F_M$
Es decir: $\omega \in L(A_L) \wedge \omega \in L(A_M)$
Por lo tanto, $\omega \in L(A) \Leftrightarrow \omega \in L(A_L) \cap L(A_M)$, es decir A acepta $L \cap M$.

Diferencia.

Dados dos lenguajes regulares, su **DIFERENCIA** también es un lenguaje regular.

Si L y M son lenguajes regulares, entonces también lo es $L - M$.

Demostración

Si L y M son lenguajes regulares, $L - M$ también lo es ya que $L - M = L \cap \overline{M}$

Inverso o reverso.

Dado un lenguaje regular, su **INVERSO** también es regular.

Si L es un lenguaje regular, entonces también lo es L^R.

Demostración

La demostración se hace por inducción estructural sobre la expresión regular E, siendo $L = L(E)$ y encontrando una expresión regular para $L^R = \big(L(E)\big)^R$.