2. Autómatas finitos

Los lenguajes regulares son aquellos que se pueden describir mediante un autómata finito.

2.2. Autómata finito determinista

Un autómata finito determinista consta de:

1. Q, un conjunto finito de estados.
2. Σ, un conjunto finito de símbolos de entrada.
3. δ, la función de transición tal que siendo $p, q \in Q$ y $a \in \Sigma$ entonces $\delta(q, a) = p$
4. q_0, un estado inicial perteneciente a Q.
5. Un conjunto de estados finales o de aceptación $F \subseteq Q$

• Diagrama de transiciones

1. Para cada estado de Q existe un nodo.
2. $\forall q \in Q$ y $\forall a \in \Sigma \longrightarrow \exists \delta(q, a) = p$
3. Existe una flecha dirigida al estado inicial q_0.
4. $\forall q \in F$ estará marcado por un doble círculo.

• Descripción de un autómata:

- Configuración o descripción instantánea:

  Un elemento de $Q \times \Sigma^*$: (q, u)

- Configuración inicial para $u \in \Sigma^*$: (q_0, u)

- ...

- Relación de cálculo entre dos configuraciones.

Sucesión de estados que desembocan en un estado final.

$\boxed{\text{Un modelo de computación es un conjunto de operaciones que nos permiten resolver un conjunto de problemas computacionales}}$

- **Tabla de transiciones**

1. Representación tabular convencional de una función.

Ejemplo:

	0	1
$\rightarrow q_0$	q_2	q_0
$* q_1$	q_1	q_1
q_2	q_2	q_1

2.2.4. Extensión a cadenas de la función de transición.

Un AFD define un lenguaje: el conjunto de todas las cadenas que dan lugar a una secuencia de transiciones desde el estado inicial hasta el estado de aceptación.

Definimos $\hat{\delta}$ la función de transición extendida como sigue:

Base: $\hat{\delta}(q, \varepsilon) = q$

Paso inductivo: Supongamos w la cadena formada por xa (siendo a el último símbolo de w).

$$\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$$

2.2.5. Definición del lenguaje de un AFD

$A = (Q, \Sigma, \delta, q_0, F)$

$$L(A) = \{ w \mid \hat{\delta}(q_0, w) \in F \}$$

2.3. Autómatas Finitos No deterministas

2.3.2. Definición

$$A = (Q, \Sigma, \delta, q_0, F)$$

dónde:

1. Q es un conjunto finito de estados
2. Σ es un conjunto finito de símbolos de entrada
3. q_0 es el estado inicial.
4. $f \in F \mid f \in Q$ y f sea un estado de aceptación.
5. δ la función de transición

$$\delta(q, w) = \mathbb{Q} \text{ (siendo } \mathbb{Q} \subseteq Q)$$

2.2.3. Función de transición extendida

Base: $\hat{\delta}(q, \varepsilon) = \{q\}$

Paso inductivo: Supongamos w con la forma xa siendo a el símbolo final de w.
Supongamos $\hat{\delta}(q, w) = \{p_1, p_2, \dots, p_k\}$.

Sea $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \dots, r_m\}$

Entonces $\hat{\delta}(q, w) = \{r_1, r_2, \dots, r_m\}$

2.3.4. El lenguaje de un AFN

Si $A = (Q, \Sigma, \delta, q_0, F)$ es un AFN

entonces:

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

2.3.5. Equivalencia de AFDs y AFNs.

Para demostrar que los AFD pueden hacer lo que los AFN realizamos una "construcción" conocida como construcción de subconjuntos, porque exige construir todos los subconjuntos del conjunto de estados del AFN.

La construcción de subconjuntos se inicia a partir de un AFN $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Su objetivo es la construcción de un AFD $D = (Q_D, \Sigma, \delta_D, q_0, F_D)$ tal que $L(N) = L(D)$, siendo:

- $Q_D = P(Q_N)$ (Por tanto si $\#Q_N = n$ estados $\rightarrow \#Q_D = 2^n$)

  Los estados inaccesibles de Q_D serán eliminados.

- $F_D = \{ S \subseteq Q_N$ tal que $S \cap F_N \neq \emptyset \}$

- Para cada conjunto $S \subseteq Q_N$ y para cada símbolo de entrada $a \in \Sigma$ entonces:

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$

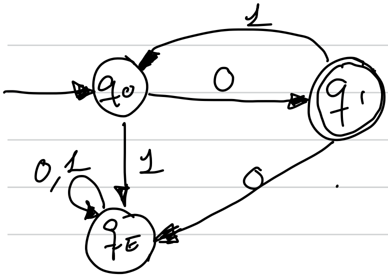
Teorema: Si $D = (Q_D, \Sigma, \delta_D, q_0, F_D)$ es el AFD construido a partir del AFN $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ mediante la construcción de subconjuntos, entonces $L(D) = L(N)$

Teorema: Un lenguaje L es aceptado por algún AFD $\longleftrightarrow$ L es aceptado por un AFN.

Ejercicio 3: Construir un autómata a partir de la siguiente gramática:

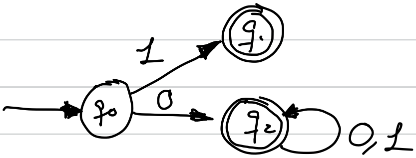
$$S \to S10 \; ; \; S \to 0$$

$$L = \{0(10)^n \mid n \geq 0\}$$



Ejercicio 2: AFND a partir de la siguiente gramática:

$$S \to 1 \mid 0B$$
$$B \to 1B \mid 0B$$
$$B \to \varepsilon$$

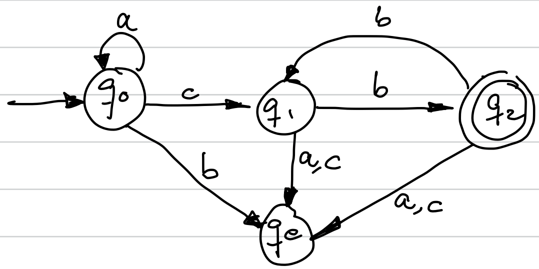


Ejercicio 11: AFD a partir de la siguiente gramática:

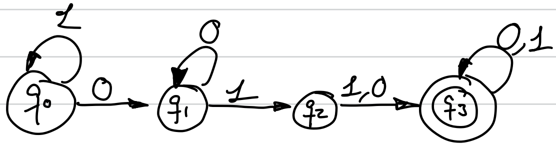
$$S \to AB \qquad A \to aA \qquad A \to c$$

$$B \to bBb \qquad B \to b$$

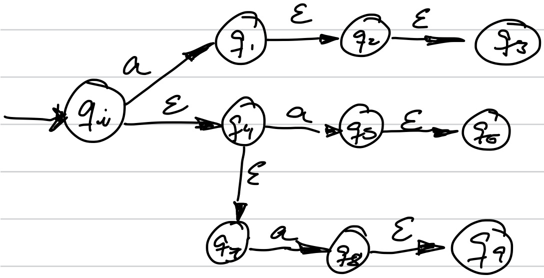
$$S \to a^n c B \longrightarrow a^n c b^m \qquad L(Ej_{11}) = \{a^n c b^m \mid n \geq 0 \text{ y } m \text{ impar}\}$$



Ejercicio 20a: AFD que contengan 011 o 010 como subcadenas

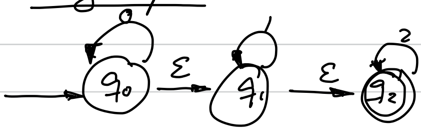


AFND con transiciones nulas



Después de introducir una 'a' llegamos a los estados $\{q_1, q_2, q_3, q_5, q_6, q_8, q_9\}$

Ejemplo:

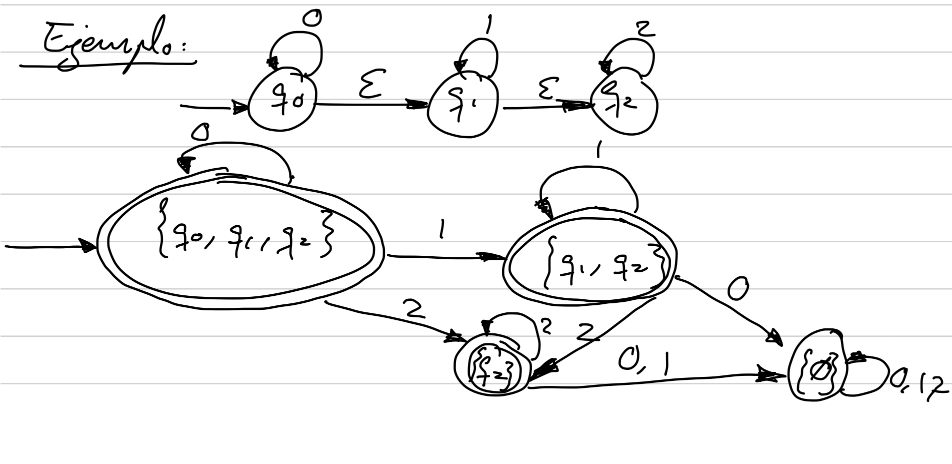


0112	$\{q_0, q_1, q_2\}$	012210	$\{q_0, q_1, q_2\}$
112	$\{q_1, q_2\}$	12210	$\{q_1, q_2\}$
12	$\{q_1, q_2\}$	2210	$\{q_2\}$
2	$\{q_2\}$	210	$\{q_2\}$
		10	$\{\emptyset\}$

$$L = \{ 0^i 1^j 2^k \mid i, j, k \geq 0 \}$$

Teorema: Dado un AFND con transiciones nulas existe otro equivalente sin transiciones nulas y viceversa.

Ejemplo:



Expresiones regulares

Expresión para núm. ceros par

$$1^*(01^*01^*)^*$$

Expresión 0110 como subcadena

$$(0+1)^*0110(0+1)^*$$

Expresión que únicamente tenga la subcadena 000 al principio de la palabra

$$000(1+10+100)^*$$

Homomorfismos

Sean A y B dos alfabetos. Si $\exists\, h: A^* \to B^*$ tq es un homomorfismo entonces si $L \subseteq A^*$ es regular...

Comprobar si dos autómatas finitos aceptan el mismo lenguaje. Basta con construir el autómata $(L(M_1)-L(M_2)) \cup (L(M_2)-L(M_1))$ y comprobar si quedan estados finales.

Un autómata es minimal si no existe otro con menos estados que él y acepte el mismo lenguaje.

Para identificar los estados indistinguibles usaremos la siguiente matriz.

b	(c,a) X					
c	X	X				
e		X	X			
f	X	X	X	(g,a) X		
g	X	~~(h,a) (g,a)~~	X	X	X	
h	X	(e,a)	X	X	X	(h,e) X
	a	b	c	e	f	g

$\begin{array}{c|cc} & 0 & 1\\ \hline h & g & c\\ a & b & f\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline h & g & c\\ b & g & c\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline h & g & c\\ e & h & f\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline g & g & e\\ f & c & g\end{array}$

$\begin{array}{c|cc} & 0 & 1\\ \hline h & g & c\\ f & c & g\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline h & g & c\\ g & g & e\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline g & g & e\\ a & b & f\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline f & c & g\\ a & b & f\end{array}$

$\begin{array}{c|cc} & 0 & 1\\ \hline g & g & e\\ b & g & c\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline g & g & e\\ c & a & c\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline g & g & e\\ e & h & f\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline f & c & g\\ b & g & c\end{array}$

$\begin{array}{c|cc} & 0 & 1\\ \hline f & c & g\\ c & a & c\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline f & c & g\\ e & h & f\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline e & h & f\\ a & b & f\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline e & h & f\\ b & g & c\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline e & h & f\\ c & a & c\end{array}$

$\begin{array}{c|cc} & 0 & 1\\ \hline c & a & c\\ a & b & f\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline c & a & c\\ b & g & c\end{array}$ $\begin{array}{c|cc} & 0 & 1\\ \hline b & g & c\\ a & b & f\end{array}$

Una vez terminado el algoritmo seleccionamos como estados $\{a,e\}$, $\{b,h\}$, c, f y g