

## 1 Ayudantía 2: Autómatas Finitos

### 1.1 DFA

DFA (Deterministic Finite Automata).

Un DFA se define por  $M = (Q, \Sigma, \delta, q_0, F)$  donde:

- $Q$ : conjunto de estados.
- $\Sigma$ : alfabeto.
- $\delta$ : función de transición tal que  $\delta : Q \times \Sigma \rightarrow Q$ .
- $q_0$ : estado inicial tal que  $q_0 \in Q$ .
- $F$ : conjunto de estados finales donde  $F \subseteq Q$ .

Lenguaje aceptado. DFA:

- $\hat{\delta}(q_i, \epsilon) = q_i$
- $\hat{\delta}(q, \sigma a) = \delta(\hat{\delta}(q, \sigma), a)$
- $\sigma \in \Sigma^*$  ;  $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$
- $\mathcal{L}(M) = \{\sigma \in \Sigma^* \mid \hat{\delta}(q_0, \sigma) \in F\}$
- $\mathcal{L}(M) = \{\alpha \in \Sigma^* \mid q_0 \alpha \vdash_M^* \alpha q_F \wedge q_F \in F\}$

Descripciones instantáneas y relaciones de transición:

- $\alpha q \beta$ :  $\alpha$  string ya se consumió,  $q$  es el estado actual y nos falta  $\beta$  string por leer.
- $\alpha a q_i \beta \rightarrow \alpha q_i a \beta \vdash_M \alpha a q_j \beta$  ; donde  $\delta(q_i, a) = q_j$  y  $M$  el autómata.

Nota:  $\vdash_M^*$  es dar 0 más pasos y  $\vdash_M^+$  es dar 1 o más pasos del autómata.

Un ejemplo gráfico de un DFA:

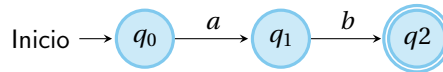


Figura 1: Para este DFA, se tiene que  $a, b \in \Sigma$

### 1.2 NFA

NFA: nondeterministic finite automata. Se define por  $M = (Q, \Sigma, \delta, q_0, F)$ .

Cada parámetro que compone  $M$  es lo mismo para un DFA, sólo que presenta esta diferencia:

$$\delta : Q \times \{\Sigma \cup \{\epsilon\}\} \rightarrow 2^Q$$

Nótese que  $2^Q$  es el conjunto de potencia de  $Q$ . Por ejemplo, si  $Q = \{1, 2, 3\}$ , su conjunto de potencia es:

$$2^Q = \{\{\epsilon\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

Ejemplo de porqué una función de transición de un NFA da como salida  $2^Q$ :

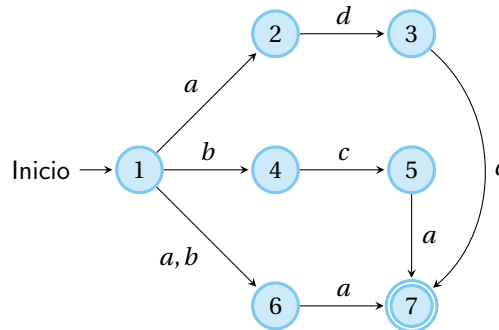


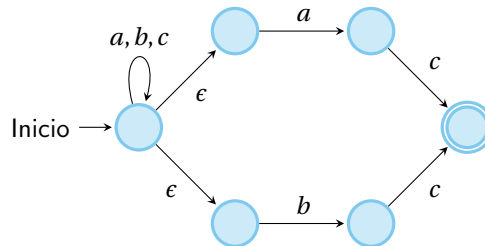
Figura 2: Ejemplo de un NFA

La Figura 2 es un NFA, porque la función de transición  $\delta(1, a) = \{2, 6\} \in 2^Q$ , es decir, el **estado siguiente no está determinado**.

Al igual que los DFA, el lenguaje aceptado por  $M$  se define por:

$$\mathcal{L}(M) = \left\{ \alpha \in \Sigma^* / q_0 \alpha \vdash_M^* \alpha q_f \wedge q_f \in F \right\}$$

Otro ejemplo de NFA:



Si el alfabeto del NFA anterior es  $\Sigma \cup \{\epsilon\} = \{a, b, c\} \cup \{\epsilon\}$ . ¿Que expresión regular representa?

**Respuesta:** representa la siguiente expresión regular:

$$(a|b|c)^* (a|b)c$$

### Ejemplo

Sea  $\Sigma = \{a, b, c\}$  y  $\alpha \in \Sigma^*$ . Encuentre el DFA y NFA de  $aac$

**Solución:**

La expresión regular de  $aac$  es:

$$a(a|b|c)^* c \tag{1.1}$$

Por lo tanto, el NFA que describe a (1.1) es:

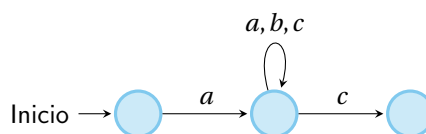


Figura 3: NFA candidato para (1.1)

Por otro lado, el DFA para (1.1) es:

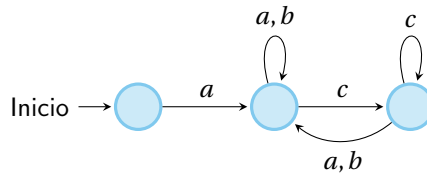


Figura 4: DFA candidato para (1.1)

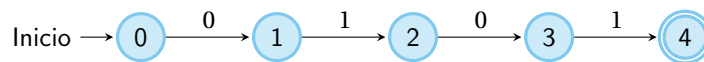
### 1.3 Preguntas

1. Dado el alfabeto  $\Sigma = \{0,1\}$ , construya un DFA que lea:

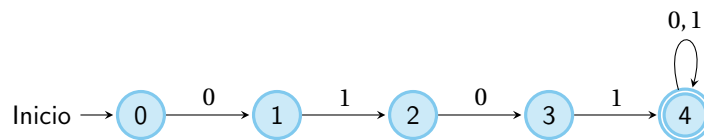
$$L = \{w \in \Sigma^* \mid w \text{ contiene } 0101\}$$

**Solución:**

La expresión regular que denota a  $L$ , es  $(0|1)^* 0101 (0|1)^*$ . Luego, comenzamos construyendo el DFA por el caso más simple: la palabra 0101:



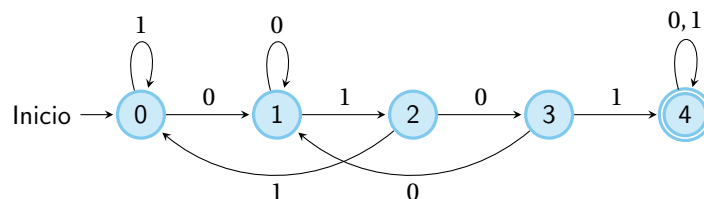
Ahora consideramos el caso  $0101(0|1)^*$ . Para ello, sólo basta con agregar un loop en el estado 4 consumiendo 0 o 1:



Por último, agregamos el resto para considerar todos los casos posibles. Para ello considere:

- Del estado 0 puedo consumir muchos 1 y luego empezar con la secuencia.
- En el estado 1 si vuelvo a consumir un 0, no rompo con la secuencia, así que podemos colocar un loop ahí.
- En el estado 2 si leo un 0, procedo a 3. De lo contrario la secuencia se rompe y deberemos empezar desde el inicio.
- Del estado 3 si leo un 1 terminamos. De lo contrario volvemos a romper la secuencia. Si volvemos al estado 1 podemos no haber problemas.

Por lo tanto, el DFA final es:



2. Sea  $\Sigma = \{0, 1\}$ . Construya un DFA sobre  $\Sigma$  que acepte números binarios múltiplos de 5.

**Solución:**

Para dar solución a este ejercicio observe el siguiente patrón:

$$2^0 = 1$$

$$2^1 = 2$$

$$2^3 = 4$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

Luego, aplicamos la operación mód  $(x, 5)$  a cada uno de los resultados anteriores:

$$\text{mód}(1, 5) = 1$$

$$\text{mód}(2, 5) = 2$$

$$\text{mód}(4, 5) = 4$$

$$\text{mód}(8, 5) = 3$$

$$\text{mód}(16, 5) = 1$$

$$\text{mód}(32, 5) = 2$$

$$\text{mód}(64, 5) = 4$$

Como puede ver el patrón se repite.

Sea  $x_{(2)} = y_{(10)}$ <sup>1</sup>. Considere las siguientes propiedades de los números binarios:

- Si hacemos un corrimiento a la izquierda agregando un 0, obtenemos  $2y$ :

$$x0_{(2)} = 2y_{(10)}$$

- Si hacemos un corrimiento a la izquierda agregando un 1, obtenemos  $2y + 1$ :

$$x1_{(2)} = (2y + 1)_{(10)}$$

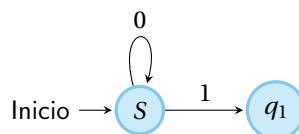
Sea  $M = (Q, \Sigma, \delta, q_0, Q)$  el DFA que acepta números binarios múltiplos de 5.

Cada estado  $q \in Q$  representa un valor del módulo y cada arco es el valor del bit agregado al momento de hacer un corrimiento.

Inicialmente no tenemos nada, así que podemos hacer dos cosas:

- Agregar 0 como locos y quedar donde mismo.
- Agregar 1 para obtener 1, cuyo módulo es 1.

Entonces:

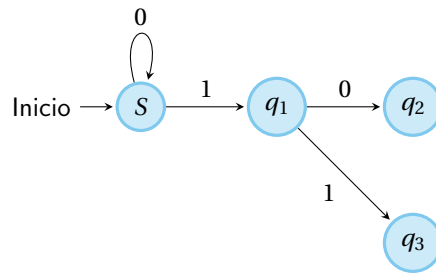


Cuando nos encontramos en  $q_1$ , tenemos un número  $x_{(2)}$  tal que su módulo es 1. Luego:

- Si le agregamos un 0, en decimal lo multiplicamos por 2, obteniendo número cuyo módulo será 2.
- Si le agregamos un 1, quedará con módulo 3 (recuerde que le sumamos 1).

<sup>1</sup>Los subíndices (2) y (10) son números binarios y decimales respectivamente.

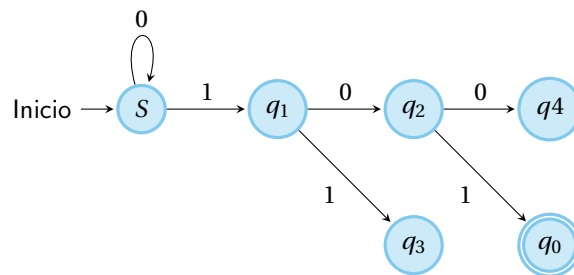
Entonces:



El estado  $q_2$  corresponde a un número con modulo 2. Entonces:

- Si agregamos un 0 al hacer el corrimiento, obtenemos un número cuyo modulo es 4.
- Si agregamos un 1 al hacer el corrimiento, obtenemos un número de módulo 0.

De acuerdo a lo anterior, se tiene que:



Si continuamos girando la manivela, obtenemos:

