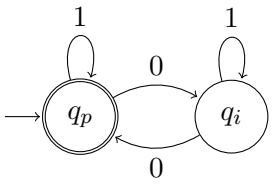
Autómatas Finitos y Gramáticas Regulares

Teoría de Lenguajes

1. Autómata finito determinístico (AFD)

Ejemplo: L_1 = Cadenas sobre $\Sigma = \{0, 1\}$ con un número par de ceros (ej 1b de la práctica 1)
Otra forma de escribirlo: $L_1 = \{\alpha \mid |\alpha|_0 \equiv 0 \mod 2\}$
El autómata A_1 reconoce el lenguaje L_1:



Podemos interpretar que q_p indica que la cantidad de ceros de la cadena leída hasta el momento es par, y q_i que la misma es impar.

1.1. Definición

Un AFD es una tupla de la forma:

$$A =< Q, \Sigma, \delta, q_0, F >$$

donde:

Q	conjunto de estados
Σ	alfabeto o conjunto de símbolos
$\delta : Q \times \Sigma \to Q$	función de transición
$q_0 \in Q$	estado inicial
$F \subseteq Q$	conjunto de estados finales

Ejemplo (continuación): La tupla que define a A_1 es entonces:

$$A_1 =< \overset{Q}{\{q_p, q_i\}}, \overset{\Sigma}{\{0, 1\}}, \delta, \overset{\text{estado inicial}}{q_p}, \overset{F}{\{q_p\}} >$$

donde $\delta : \{q_p, q_i\} \times \{0, 1\} \to \{q_p, q_i\}$ está dada por la siguiente tabla:

δ	0	1
q_p	q_i	q_p
q_i	q_p	q_i

1.2. Configuración instantánea

Para definir el lenguaje aceptado por un autómata finito, en la práctica usaremos una formalización distinta a la usada en la teórica, basada en (Aho, Ulman, Vol. 1, sección 2.2.3). El primer paso es definir *configuración instantánea*, que es una foto del proceso de reconocimiento de una cadena por el autómata en un instante dado. En un auómata finito tiene esta forma:

$$(q, \alpha) \in Q \times \Sigma^*$$

Donde q es el estado actual y α es lo que queda por consumir de la cadena de entrada.

1.3. Relación de transición entre configuraciones

La relación entre configuraciones $\vdash$ nos indica cuándo podemos pasar de una configuración instantánea a otra:

$$(q_i, a.\alpha) \vdash_A (q_j, \alpha) \Leftrightarrow q_j = \delta(q_i, a)$$

Es decir, podemos pasar del estado q_i al estado q_j consumiendo el símbolo a de la cadena de entrada si q_j es el valor de la función de transición aplicada a (q_i, a)

1.4. Pertenencia al lenguaje

Una cadena α pertenecerá al lenguaje del autómata si partiendo de la configuración (q_0, α) se puede llegar a consumir toda la cadena llegando a un estado final. Es decir, llegar a la configuración (q_f, λ), donde q_f es un estado final:

$$\alpha \in L(A) \Leftrightarrow \exists q_f \in F \mid (q_0, \alpha) \vdash_A^* (q_f, \lambda)$$

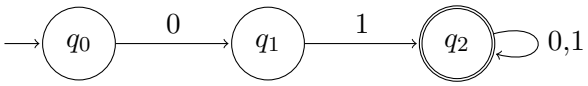
Recordar que $\vdash^*$ es la clasura reflexiva transitiva de la relación de transición $\vdash$, es decir, aplicar cero o más veces la relación. El caso "cero veces" aplica cuando la cadena de entrada es λ: la cadena vacía será aceptada si y sólo si el estado inicial es tambien un estado final.

Ejemplo (continuación): Seguimiento del proceso de reconocimiento de cadenas que pertenecen y no pertenecen al lenguaje:

$$(q_0, 010) \vdash_{A_1} (q_1, 10) \vdash_{A_1} (q_1, 0) \vdash_{A_1} (q_0, \lambda) \checkmark$$

$$(q_0, 101) \vdash_{A_1} (q_0, 01) \vdash_{A_1} (q_1, 1) \vdash_{A_1} (q_1, \lambda) \times$$

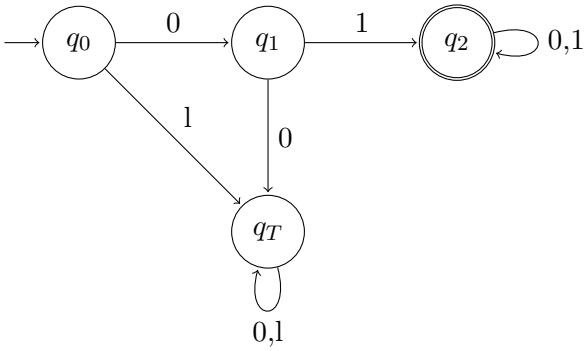
Ejemplo: $L_2 =$ cadenas que comienzan por 01.
También lo podemos escribir como $L_2 = 01\Sigma^*$.
Proponemos el autómata A_2:



Notar que la función de transición así está incompleta:

δ	0	1
q_0	q_1	?
q_1	?	q_2
q_2	q_2	q_2

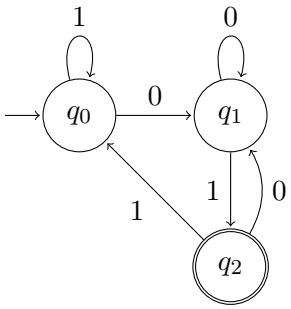
Para que el autómata quede bien definido, agregamos un estado de no aceptación, que solemos llamar *estado trampa*:



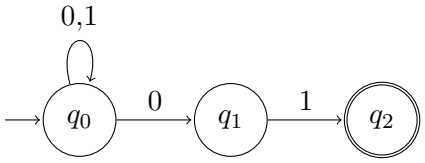
Ahora la función de transición sí está completa:

δ	0	1
q_0	q_1	q_T
q_1	q_T	q_2
q_2	q_2	q_2
q_T	q_T	q_T

Ejemplo: L_3 = Cadenas que terminen en 01.
También lo podemos escribir como $L_3 = \Sigma^*01$.
A_3:



Notar que aunque la definición es similar a la de L_2, el autómata se ve más complicado. Nos gustaría expresar algo como: "puede venir cualquier cadena y al final 01". Entonces proponemos A'_3:



Pero éste ya no es un AFD porque hay más de una opción para la combinación $(q_0, 0)$. Pasemos entonces a formalizar los autómatas finitos no determinísticos.

2. Autómata finito no determinístico con transiciones λ (AFND-λ)

2.1. Definición

Al igual que los AFD, un AFND-λ es una tupla de la forma:

$$A =< Q, \Sigma, \delta, q_0, F >$$

Pero cambia el tipo de la función de transición:

$$\delta : Q \times (\Sigma \cup \lambda) \rightarrow P(Q)$$

Hay dos cambios con respecto a los AFD: el primero es que en lugar de un solo estado, la función devuelve un conjunto de estados posibles a los que puede ir el autómata. El segundo es que además de transiciones que consumen un símbolo de la cadena de entrada, también puede haber transiciones por λ, es decir sin consumir ningún símbolo.

Ejemplo (continuación): La tupla que define a A'_3 es entonces:

$$A'_3 =< \{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\} >$$

donde δ está dada por la siguiente tabla:

δ	0	1
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$

Como se ve, ya no es necesario "completar" el autómata agregando un estado trampa como en el caso de los AFDs, porque la función de transición puede devolver un conjunto vacío para los casos en los que no hay ninguna transición para alguna combinación estado - símbolo de entrada.

2.2. Relación de transición entre configuraciones

Tenemos dos casos posibles: transición consumiendo un símbolo de entrada y transición λ:

$$(q_i, a.\alpha) \vdash_A (q_j, \alpha) \Leftrightarrow q_j \in \delta(q_i, a)$$

$$(q_i, \alpha) \vdash_A (q_j, \alpha) \Leftrightarrow q_j \in \delta(q_i, \lambda)$$

2.3. Pertenencia al lenguaje

La definición es idéntica que para AFD:

$$\alpha \in L(A) \Leftrightarrow \exists q_f \in F \mid (q_0, \alpha) \vdash_A^* (q_f, \lambda)$$

Una cadena va a estar en el lenguaje si existe alguna secuencia de configuraciones que lleve a un estado final consumiendo toda la cadena.

Ejemplo (continuación): Seguimientos de una cadena perteneciente a L_3.

$$(q_0, 101) \vdash_{A'_3} (q_0, 01) \vdash_{A'_3} (q_1, 1) \vdash_{A'_3} (q_2, \lambda) \checkmark$$

$$(q_0, 101) \vdash_{A'_3} (q_0, 01) \vdash_{A'_3} (q_0, 1) \vdash_{A'_3} (q_0, \lambda) \times$$

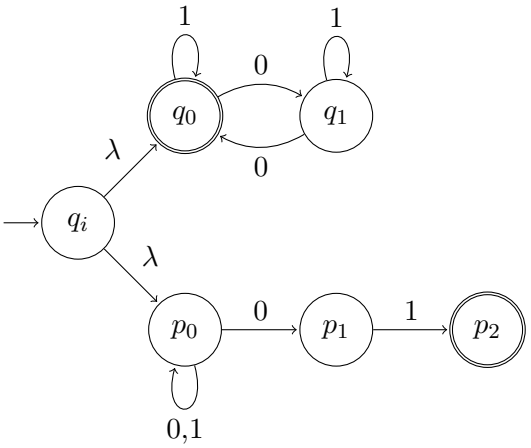
Notar que la cadena pertenece al lenguaje porque existe al menos una secuencia de configuraciones que lleva a un estado final consumiendo toda la cadena. No importa que otras secuencias no lleven a aceptar.

Ejemplo (continuación): Seguimientos de una cadena no perteneciente a L_3.

$$(q_0, 010) \vdash_{A'_3} (q_0, 10) \vdash_{A'_3} (q_0, 0) \vdash_{A'_3} (q_1, \lambda) \times$$
$$(q_0, 010) \vdash_{A'_3} (q_1, 10) \vdash_{A'_3} (q_2, 0) \times$$

La primera secuencia consume toda la cadena pero no llega a un estado final. La segunda secuencia llega a un estado final pero no consume toda la cadena. En ninguna secuencia posible para esta cadena se cumplen ambas condiciones, por lo que la cadena no pertenece al lenguaje de A_3.

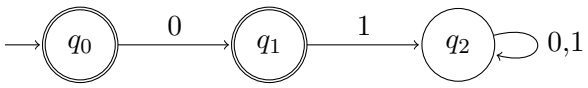
Ejemplo: $L_4 = L_1 \cup L_3$, cadenas que terminen en 01 o tengan cantidad par de 0s.

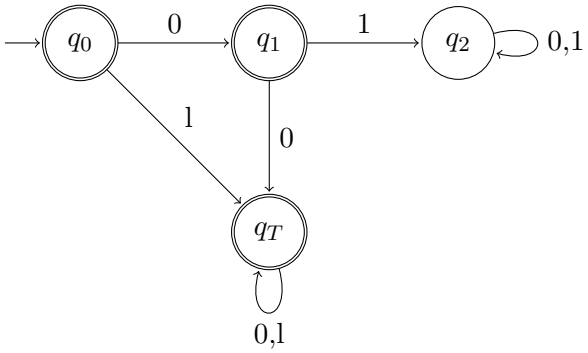


2.4. Complemento

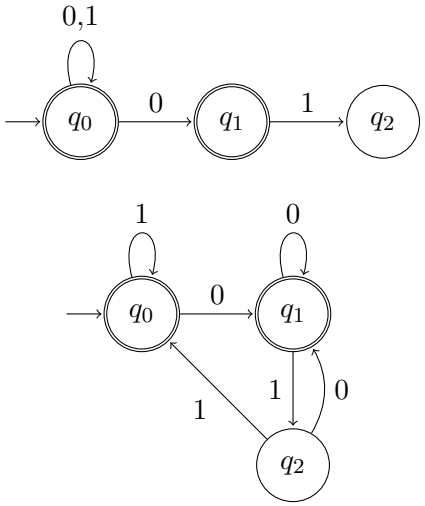
Dado un AFD COMPLETO, hacer $F' = Q \backslash F$.

Ejemplo: L_2^c = cadenas que NO comienzan por 01.





Ejemplo: L_3^c = cadenas que NO terminan en 01.

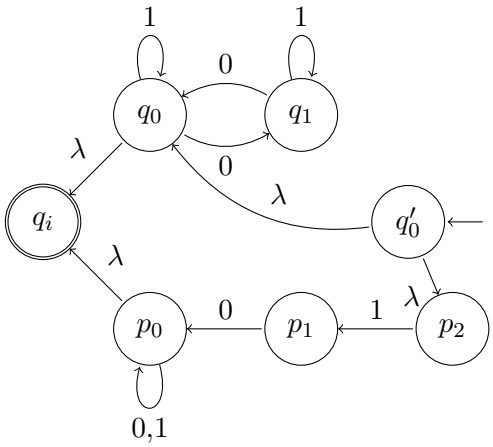


2.5. Reversa

Dado un AFND-λ obtener $A' =< Q', \Sigma, \delta', q_0', F' >$:

$$Q' = Q' \cup \{q_0'\}$$
$$\delta'(q_0', \lambda) = F$$
$$q_2 \in \delta'(q_1, a) \Leftrightarrow q_1 \in \delta(q_2, a)$$
$$F' = q_0$$

Ejemplo: L_4^r.



Resolución: Se ve que no hace falta un estado inicial nuevo porque había un solo estado final.