Autómatas y Lenguajes Formales 2019-II
Facultad de Ciencias UNAM*
Nota de Clase 4

Favio E. Miranda Perea   A. Liliana Reyes Cabello   Lourdes González Huesca

12 de marzo de 2019

1. Autómatas No-Deterministas

El determinismo de un autómata, deseable desde el punto de vista teórico, puede provocar complicaciones en la práctica. Veamos las diferencias entre estos conceptos:

- **Determinismo:** dado un estado q y un símbolo a existe una única transición $\delta(q, a) = p$, es decir δ es una función total.
- **No-determinismo:** no hay una transición única al leer un símbolo a en un estado dado q.

El no-determinismo se traduce en que hay más de una transición al leer un símbolo, es decir, $\delta(q, a)$ deja de ser función. O bien no hay transición, es decir, $\delta(q, a)$ no está definida (δ se vuelve función parcial). Sin embargo la máquina funciona únicamente al leer un símbolo.

Veamos cómo se puede agregar el no-determinismo a la definición de autómata que vimos anteriormente:

Definición 1. *Un autómata finito* ***no*** *determinista (AFN) es una quintupla* $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ *donde*

- Q *es un conjunto finito de estados.*
- Σ *es el alfabeto de entrada.*
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ *es la función de transición.*
- $q_0 \in Q$ *es el estado inicial.*
- $F \subseteq Q$ *es el conjunto de estados finales.*

Obsérvese que la imagen de δ es ahora un elemento de $\mathcal{P}(Q)$, es decir es un subconjunto de estados de Q. Además $\delta(q, a) = \{q_1, q_2, \ldots, q_n\}$ indica que al leer el símbolo a en el estado q la máquina puede pasar a cualquiera de los estados $q_1, \ldots, q_n$. Si $\delta(q, a) = \varnothing$ entonces no hay transición posible desde el estado q al leer a, es decir, la máquina está bloqueada.

Veamos como las nociones vistas para los AFD se modifican:

*Material elaborado en el marco del proyecto PAPIME PE102117

Definición 2. *Sea* $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ *un AFN. La función de transición* δ *se extiende a cadenas mediante una función* $\delta^\star : Q \times \Sigma^\star \to \mathcal{P}(Q)$ *definida recursivamente como sigue:*

- $\delta^\star(q, \varepsilon) = q$
- $\delta^\star(q, wa) = \bigcup_{p \in \delta^\star(q,w)} \delta(p, a)$
- *Alternativamente:* $\delta^\star(q, aw) = \bigcup_{p \in \delta(q,a)} \delta^\star(p, w)$

El lenguaje de aceptación se define mediante $\delta^\star$ como sigue:

$$L(M) = \{w \in \Sigma^\star \mid \delta^\star(q_0, w) \cap F \neq \varnothing\}$$

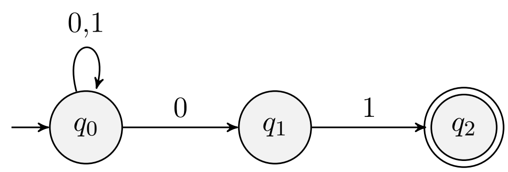
Es decir, $w \in L(M)$ si y sólo si existe al menos un cómputo o un procesamiento de w que conduce a un estado final al iniciar la máquina en q_0.

[START] Ejemplos

Ejemplos de AFN:

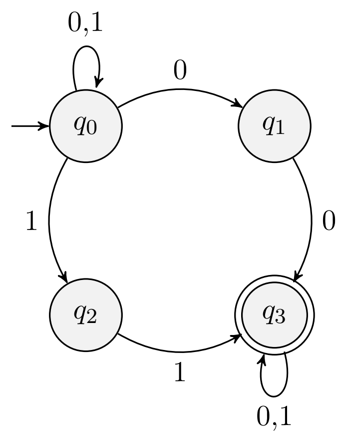
1. Sea un $M = \langle \{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\} \rangle$ un AFN que acepta todas las cadenas que terminan en `01`.

δ	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	$\varnothing$	$\{q_2\}$
$*q_2$	$\varnothing$	$\varnothing$



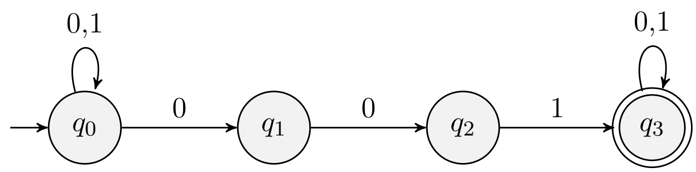
2. Sea un $M = \langle \{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_3\} \rangle$ un AFN acepta todas las cadenas que contienen `00` o `11` como subcadena.

δ	0	1
q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$
q_1	$\{q_3\}$	$\varnothing$
q_2	$\varnothing$	$\{q_3\}$
q_3	$\{q_3\}$	$\{q_3\}$



3. Sea un $M = \langle \{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_3\} \rangle$ un AFN que acepta todas las cadenas que contienen la subcadena `011`.

δ	0	1
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\varnothing$
q_2	$\varnothing$	$\{q_3\}$
q_3	$\{q_3\}$	$\{q_3\}$



[END] Ejemplos

[START] Ejecución

Ejemplos de ejecución de $\delta^\star$ para el AFN del inciso 1:

- Cadena aceptada, con la entrada `00101`.

  1. $\delta^\star(q_0, \varepsilon) = \{q_0\}$
  2. $\delta^\star(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
  3. $\delta^\star(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \varnothing = \{q_0, q_1\}$
  4. $\delta^\star(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$
  5. $\delta^\star(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \varnothing = \{q_0, q_1\}$
  6. $\delta^\star(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

  En la línea (1) se aplica la regla básica. Se obtiene la línea (2) aplicando δ al estado q_0, que está en el conjunto anterior, y se obtiene $\{q_0, q_1\}$ como resultado. En la línea (3) se obtiene de tomar la unión sobre los dos estados en el conjunto anterior de lo que se obtuvo cuando les aplicamos δ con la entrada `0`. Que es, $\delta(q_0, 0) = \{q_0, q_1\}$, mientra que $\delta(q_1, 0) = \varnothing$. Para la línea (4), tomamos la unión de $\delta(q_0, 1) = \{q_0\}$ y $\delta(q_1, 1) = \{q_2\}$. Las líneas (5) y (6) son similares a las líneas (3) y (4).

- Cadena no aceptada, con la entrada `110`.

  1. $\delta^\star(q_0, \varepsilon) = \{q_0\}$
  2. $\delta^\star(q_0, 1) = \delta(q_0, 1) = \{q_0\}$
  3. $\delta^\star(q_0, 11) = \delta(q_0, 1) = \{q_0\}$
  4. $\delta^\star(q_0, 110) = \delta(q_0, 0) = \{q_0, q_1\}$

  Como $\{q_0, q_1\} \cap \{q_2\} = \varnothing$, entonces no es aceptado.

Ejemplos de ejecución de $\delta^\star$ para el AFN del inciso 3:

- Cadena aceptada, con la entrada `10010`.

  1. $\delta^\star(q_0, \varepsilon) = \{q_0\}$
  2. $\delta^\star(q_0, 1) = \delta(q_0, 1) = \{q_0\}$

3. $\delta^\star(q_0, 10) = \delta(q_0, 0) = \{q_0, q_1\}$
4. $\delta^\star(q_0, 100) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$
5. $\delta^\star(q_0, 1001) = \delta(q_0, 1) \cup \delta(q_2, 1) = \{q_0\} \cup \{q_3\} = \{q_0, q_3\}$
6. $\delta^\star(q_0, 10010) = \delta(q_0, 0) \cup \delta(q_3, 0) = \{q_0\} \cup \{q_3\} = \{q_0, q_3\}$

Por lo que la entrada es aceptada, ya que al procesarla nos lleva a un estado final que es q_3, porque $\delta^\star(q_0, 10010) \cap \{q_3\} \neq \varnothing$.

- Cadena no aceptada, con la entrada `0101`.

1. $\delta^\star(q_0, \varepsilon) = \{q_0\}$
2. $\delta^\star(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
3. $\delta^\star(q_0, 01) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \varnothing = \{q_0\}$
4. $\delta^\star(q_0, 010) = \delta(q_0, 0) = \{q_0, q_1\}$
5. $\delta^\star(q_0, 0101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \varnothing = \{q_0\}$

[END] Ejecución

1.1. Eliminación del no-determinismo

Es claro que todo AFD es un AFN, con la particularidad de que $\delta(p, a)$ consta de un único estado. El recíproco también es cierto, dando como resultado la equivalencia entre ambos modelos de reconocimiento de cadenas.

La idea para transformar un AFN en un AFD es considerar a cada conjunto de estados $\delta(p, a)$ del AFN como un único estado del nuevo AFD. A este método se conoce como la *construcción de subconjuntos*, cuyos detalles técnicos damos a continuación.

Definición 3. *Dado un AFN $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ definimos un AFD $M^D = \langle Q^D, \Sigma, \delta_D, q_0^D, F^D\rangle$ como sigue:*

- $Q^D =_{def} \mathcal{P}(Q)$
- $\delta_D(S, a) =_{def} \bigcup_{q \in S} \delta(q, a)$
- $q_0^d =_{def} \{q_0\}$
- $F^d =_{def} \{S \subseteq Q \mid S \cap F \neq \varnothing\}$

A continuación mostramos que ambos autómatas son equivalentes.

Lema 1. *Sea M un autómata finito no determinista. Se cumple lo siguiente*

$$\forall w \in \Sigma^\star(\delta^\star(q_0, w) = \delta_D^\star(\{q_0\}, w))$$

Demostración. Inducción sobre w. □

Proposición 1. *Sea M un autómata finito no determinista. Se cumple que $L(M) = L(M^D)$.*

Demostración. La demostración se sigue del lema anterior. □

[START] Ejemplos

Ejemplos de transformación de AFN a AFD

- Tomemos el primer ejemplo del AFN, que es el que acepta todas las cadenas que terminan en `01`.

  - Q^D es el conjunto de subconjuntos de Q, es decir, el conjunto potencia de Q. Notemos que si Q tiene n estados, entonces Q^D tiene 2^n estados. Suele pasar que no todos estos estados son accesibles desde el estado inicial de Q^D. Los estados inaccesibles pueden ser "desechados', y así el número de estados de M^D puede ser mucho menor que 2^n.
  - F^D es el conjunto de subconjuntos S de Q tal que $S \cap F \neq \varnothing$. Es decir, F^D es todos los conjuntos de los estados de M que incluye al menos un esrado de aceptación de M.
  - Para cada conjunto $S \subseteq Q$, y para cada símbolo a como entrada en Σ,

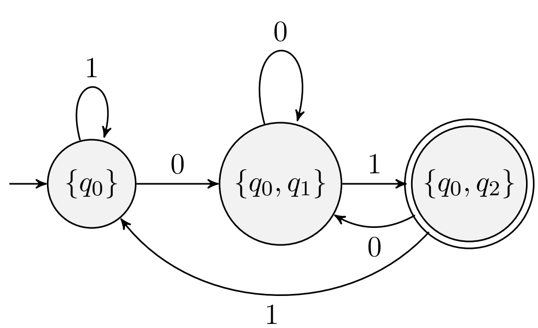
$$\delta_D(S,a) = \bigcup_{q \in S} \delta(q,a)$$

    Esto es, calcular $\delta_D(S,a)$ observamos a todos los estados $q \in S$, miremos qué estados de M van de q con la entrada a, y tomemos la unión de todos esos estados.

	0	1
$\varnothing$	$\varnothing$	$\varnothing$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\varnothing$	$\{q_2\}$
$*\{q_2\}$	$\varnothing$	$\varnothing$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\circ\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_2\}$	$\varnothing$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

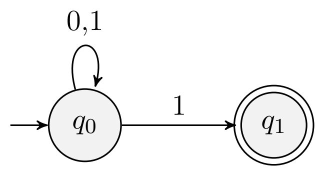
    Observemos que la construcción de los subconjuntos produce $2^3 = 8$ estados, correspondientes a los tres estados. También que el estado $\{q_0\}$ es el inicial y los estados con el $*$ a la derecha son estados no accesibles.

Con la tabla anterior, sabemos qué estados son accesible y sus transiciones. A continuación se mostrará el AFN completado. Notemos que solo tiene tres estados, lo cual por conincidencia, es exactamente el mismo número de estados del AFN. Sin embargo, el AFD tiene seis transiciones.



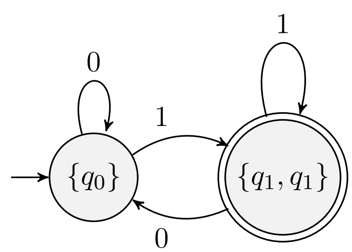
- Consideremos un $M = \langle \{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_1\} \rangle$ un AFN que acepta todas las cadenas que terminan con `1`.

δ	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	$\varnothing$	$\varnothing$



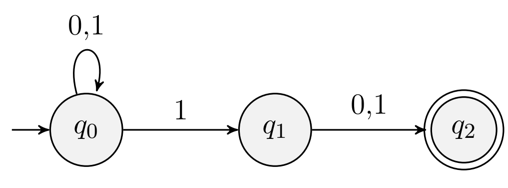
Donde, siguiendo la definición para eliminar en no-determinismo se tiene el siguiente AFN:

	0	1
$\rightarrow \{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\circ\{q_0, q_1\}$	$\{q_0\}$	$\{q_0, q_1\}$



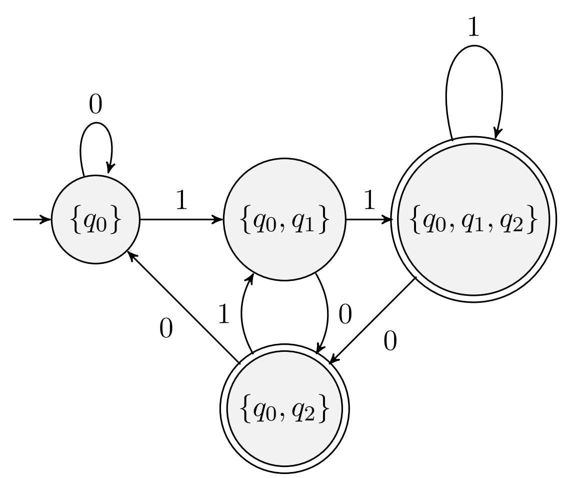
- Consideremos un $M = \langle \{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\} \rangle$ un AFN que acepta todas las cadenas que terminan con el penúltimo símbolo que siempre sea `1`.

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



Donde, siguiendo la definición para eliminar en no-determinismo se tiene el siguiente AFN:

	0	1
$\rightarrow \{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\circ\{q_0, q_1\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\circ\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$



[END] Ejemplos

2. AFN con ε-transiciones

Otra de las máquinas que son útiles para procesar cadenas son las que permiten procesar cadenas vacías, no sólo al tener como estado final a q_0 sino que permiten procesar (sub)cadenas vacías en una parte intermedia. De esta forma se definen los autómatas con transiciones etiquetadas con la cadena $\boldsymbol{\varepsilon}$:

Definición 4. *Un autómata finito **no** determinista con $\boldsymbol{\varepsilon}$-transiciones (AFNε) es una quintupla $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ donde*

- *Q es un conjunto finito de estados.*
- *Σ es el alfabeto de entrada.*
- *$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ es la función de transición.*
- *$q_0 \in Q$ es el estado inicial.*
- *$F \subseteq Q$ es el conjunto de estados finales.*

De la definición de $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ se observa que la única diferencia está en el dominio de δ. Es decir que las transiciones de la forma $\delta(q, \varepsilon) = a$ están permitidas e indican que la máquina puede cambiar de estado *sin* leer ningún símbolo. Algunas observaciones importantes acerca de los AFNε son:

- Se permiten múltiples cómputos para una cadena de entrada.
- Pueden existir cómputos bloqueados.
- A diferencia de los AFD y AFN simples, pueden existir cómputos infinitos, es decir, surge la **no-terminación**.
- La presencia de $\boldsymbol{\varepsilon}$-transiciones permite mayor libertad y modularidad en el diseño.

Para extender la definición de transiciones a procesamiento de cadenas es necesario introducir un concepto previo que considera los estados a los que la máquina puede llegar al incluir las cadenas vacías.

Definición 5. *Dado un estado q, definimos la $\boldsymbol{\varepsilon}$-cerradura de q como el conjunto de estados alcanzables desde q mediante cero o más $\boldsymbol{\varepsilon}$-transiciones. Es decir*

$$Cl_{\boldsymbol{\varepsilon}}(q) = \{s \in Q \mid \exists p_1, \ldots, p_n \ \text{con}\ p_1 = q,\ p_n = s,\ p_i \in \delta(p_{i-1}, \boldsymbol{\varepsilon})\}$$

Recursivamente:

- *$q \in Cl_{\boldsymbol{\varepsilon}}(q)$*
- *Si $r \in Cl_{\boldsymbol{\varepsilon}}(q)$ y $\delta(r, \boldsymbol{\varepsilon}) = s$ entonces $s \in Cl_{\boldsymbol{\varepsilon}}(q)$*

La definición anterior puede darse mediante reglas de inferencia como sigue:

$$\frac{}{q \in Cl_{\boldsymbol{\varepsilon}}(q)}(Cl1) \qquad \frac{r \in Cl_{\boldsymbol{\varepsilon}}(q) \quad \delta(r,\boldsymbol{\varepsilon}) = s}{s \in Cl_{\boldsymbol{\varepsilon}}(q)}(Cl2)$$

La operación de ε-cerradura se extiende a conjuntos de estados como sigue:

$$Cl_{\boldsymbol{\varepsilon}}(S) = \bigcup_{q \in S} Cl_{\boldsymbol{\varepsilon}}(q)$$

Con la $\boldsymbol{\varepsilon}$-cerradura de estados se puede definir la función de transición extendida $\delta^\star$:

Definición 6. *Sea $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ un AFN$\boldsymbol{\varepsilon}$. La función de transición δ se extiende a cadenas mediante una función*

$$\delta^\star : Q \times (\Sigma^\star \cup \{\boldsymbol{\varepsilon}\}) \to \mathcal{P}(Q)$$

definida recursivamente como sigue:

- $\delta^\star(q, \boldsymbol{\varepsilon}) = Cl_{\boldsymbol{\varepsilon}}(q)$
- $\delta^\star(q, w\sigma) = Cl_{\boldsymbol{\varepsilon}}\Big(\bigcup_{p \in \delta^\star(q,w)} \delta(p,\sigma)\Big)$

2.1. Eliminación de ε-transiciones

Estudiaremos el proceso de eliminación de transiciones $\boldsymbol{\varepsilon}$. Esta eliminación implica que un AFN es también un AFN$\boldsymbol{\varepsilon}$. Es decir que cualquier AFN$\boldsymbol{\varepsilon}$ es inmediatamente un AFN con la particularidad de que no existen $\boldsymbol{\varepsilon}$-transiciones.

Definición 7. *Dado un AFN$\boldsymbol{\varepsilon}$, $M = \langle Q, \Sigma, \delta, q_0, F\rangle$, definimos el AFN asociado a M, denotado M_N, como*

$$M_N = \langle Q, \Sigma, \delta_N, q_0, F_N\rangle,$$

donde:

- $\delta_N(q,\sigma) =_{def} \delta^\star(q,\sigma)$
- $F_N =_{def} F \cup \{q_0\}$ *si* $Cl_{\boldsymbol{\varepsilon}}(q_0) \cap F \neq \varnothing$ *o bien* $F_N =_{def} F$ *en caso contrario.*

La equivalencia entre M y M_N se bosqueja a continuación

Lema 2. *Sean M un AFN$\boldsymbol{\varepsilon}$, $q \in Q$ y $w, v \in \Sigma^\star$. Se cumple que*

$$\delta^\star(q, wv) = \bigcup_{p \in \delta^\star(q,w)} \delta^\star(p, v)$$

Demostración. Inducción sobre v □

Lema 3. *Sean M un AFN$\boldsymbol{\varepsilon}$, M_N el AFN asociado a M, $q \in Q$ y $w \in \Sigma^\star$. Si $w \neq \boldsymbol{\varepsilon}$ entonces*

$$\delta_N^\star(q, w) = \delta^\star(q, w)$$

Demostración. Inducción sobre w. El lema anterior será de utilidad. □

Proposición 2. *Sea M un AFN$\boldsymbol{\varepsilon}$. Los autómatas M y M_N son equivalentes, es decir $L(M) = L(M_N)$*

Demostración. Es consecuencia directa del lema anterior y de la definición de F_N. □

3. Equivalencias

Nuevamente surgen preguntas como la equivalencia entre autómatas deterministas, no deterministas y con transiciones ε.

En la subsección 1.1 se estudió la forma de eliminar el no-determinismo de un autómata, creando un AFD. Es decir AFN $\Rightarrow$ AFD.

En la subsección pasada se revisó la forma de eliminar ε-transiciones y de esta forma obtener un AFN. Así también se observó que un AFN es exactamente un AFNε ya que incluye el no-determinismo pero sin transiciones de la cadena vacía.

Esta equivalencia, AFN $\Leftrightarrow$ AFNε, cierra el ciclo de equivalencias de autómatas finitos. Cualquier tipo de autómata finito puede convertirse en un AFD y viceversa, es decir:

$$\text{AFD} \Leftrightarrow \text{AFN} \Leftrightarrow \text{AFN}\varepsilon$$

Nuestra siguiente meta es probar el Teorema de Kleene, el cual es uno de los resultados más importantes en la Teoría de la Computación pues asegura la equivalencia entre dos de nuestros tres conceptos fundamentales: los autómatas finitos y los lenguajes regulares.