

Teoría de Autómatas e Linguaxes Formais: Introducción



Senén Barro Ameneiro, CiTIUS

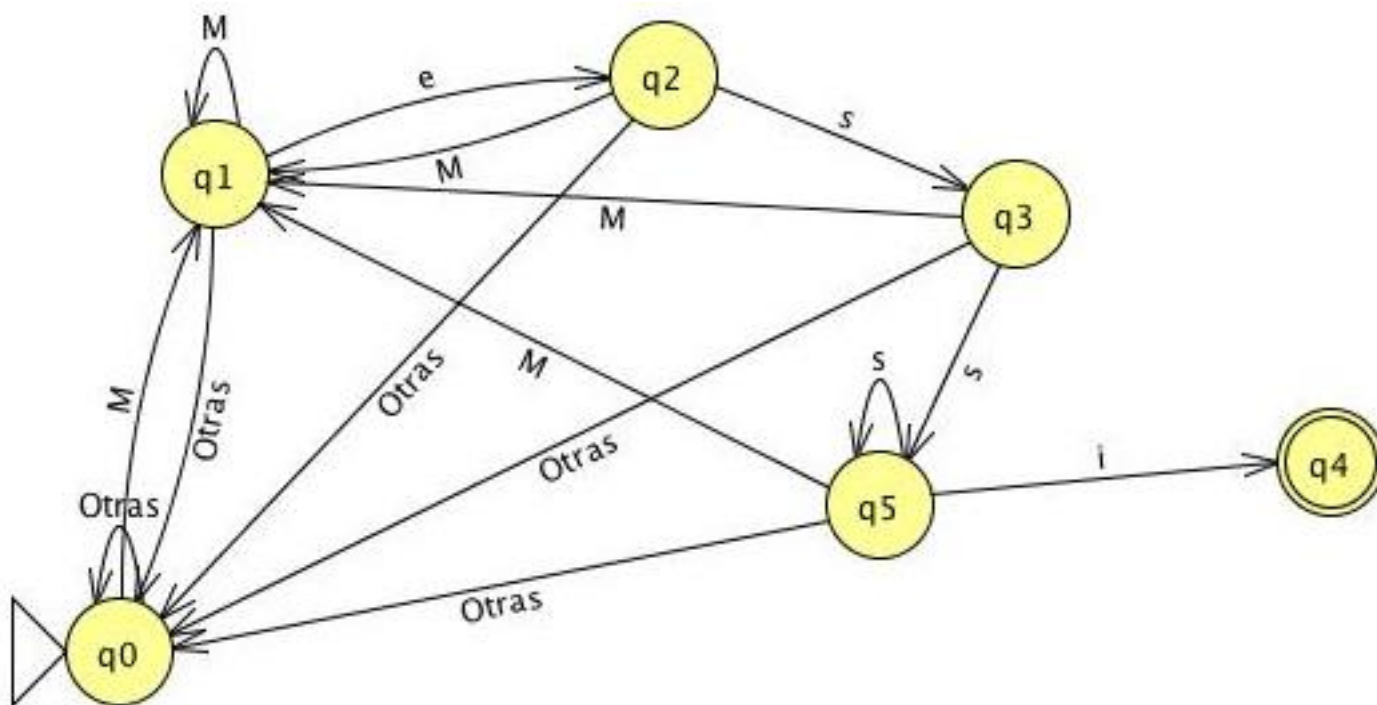
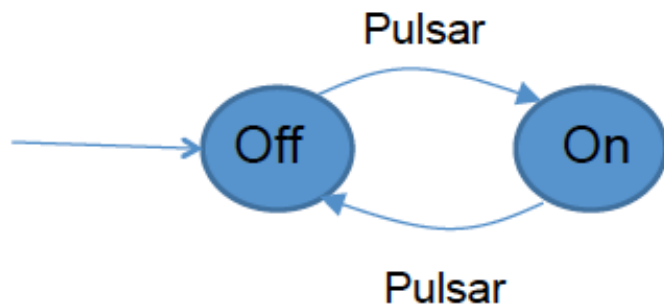
@SenenBarro

Material elaborado fundamentalmente polo
profesor Manuel Mucientes Molina

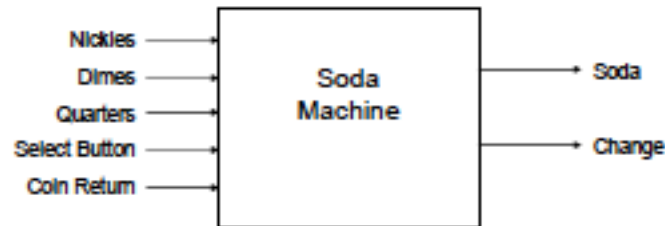
Bibliografía

- J.E. Hopcroft, R. Motwani, y J.D. Ullman. "Teoría de Autómatas, Lenguajes y Computación". Addison Wesley. 2008.
 - capítulo 1
- P. Linz. "An Introduction to Formal Languages and Automata". Jones and Bartlett Publishers, Inc. 2001.
 - capítulo 1

Ejemplos de autómatas MUY simples



Ejemplo: máquina expendedora



Current state	Nickle	Dime	Quarter	Select button	Coin return button
0¢	5¢ ¹	10¢ ¹	25¢ ¹	0¢	0¢ ⁴
5¢	10¢ ¹	15¢ ¹	30¢ ¹	5¢	0¢ ⁴
10¢	15¢ ¹	20¢ ¹	35¢ ¹	10¢	0¢ ⁴
15¢	20¢ ¹	25¢ ¹	40¢ ¹	15¢	0¢ ⁴
20¢	25¢ ¹	30¢ ¹	45¢ ¹	20¢	0¢ ⁴
25¢	25¢ ²	25¢ ²	25¢ ²	0¢ ³	0¢ ⁴
30¢	30¢ ²	30¢ ²	30¢ ²	0¢ ³	0¢ ⁴
35¢	35¢ ²	35¢ ²	35¢ ²	0¢ ³	0¢ ⁴
40¢	40¢ ²	40¢ ²	40¢ ²	0¢ ³	0¢ ⁴
45¢	45¢ ²	45¢ ²	45¢ ²	0¢ ³	0¢ ⁴

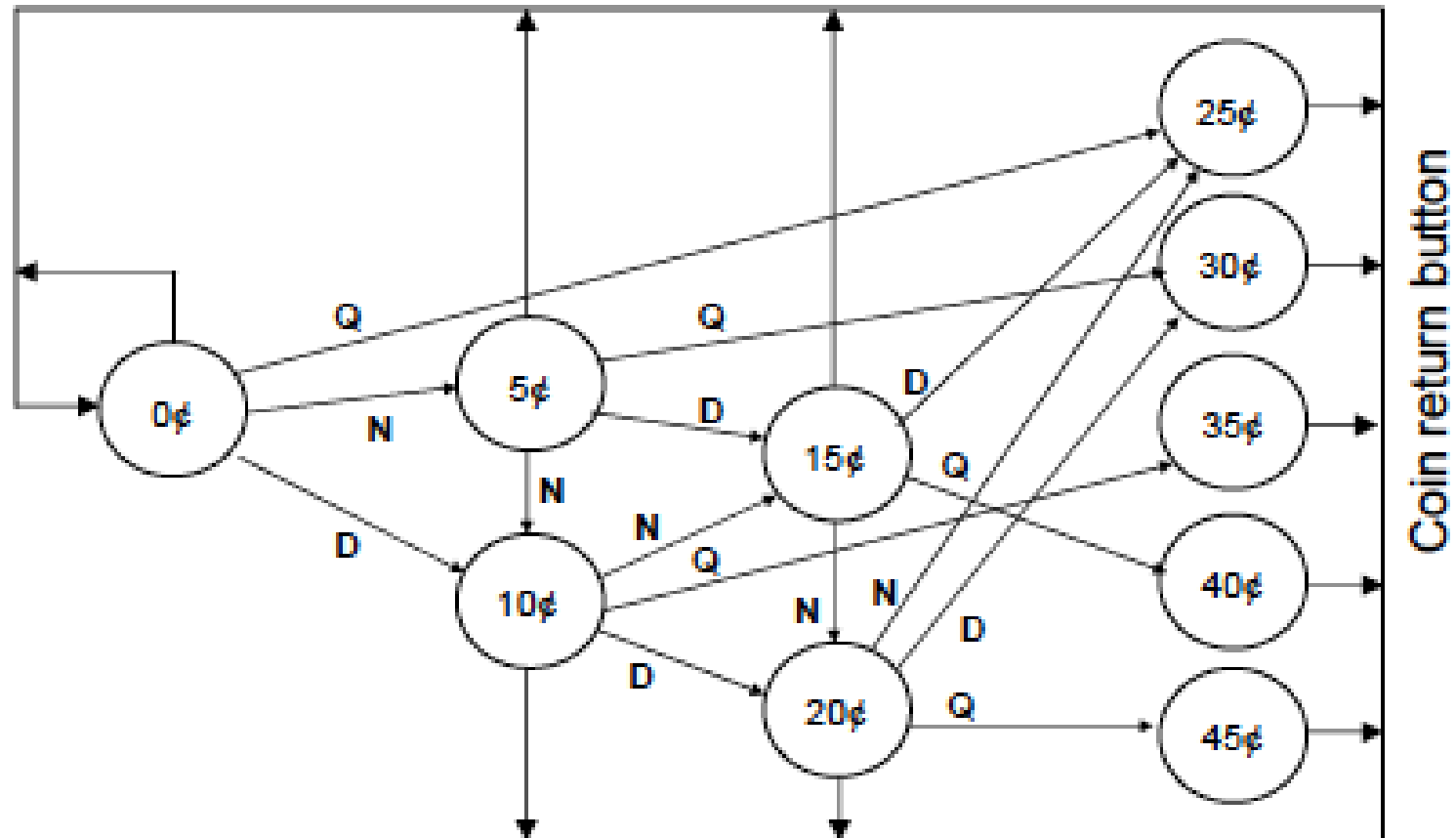
¹ – Accepts inserted money

² – Any money inserted is returned

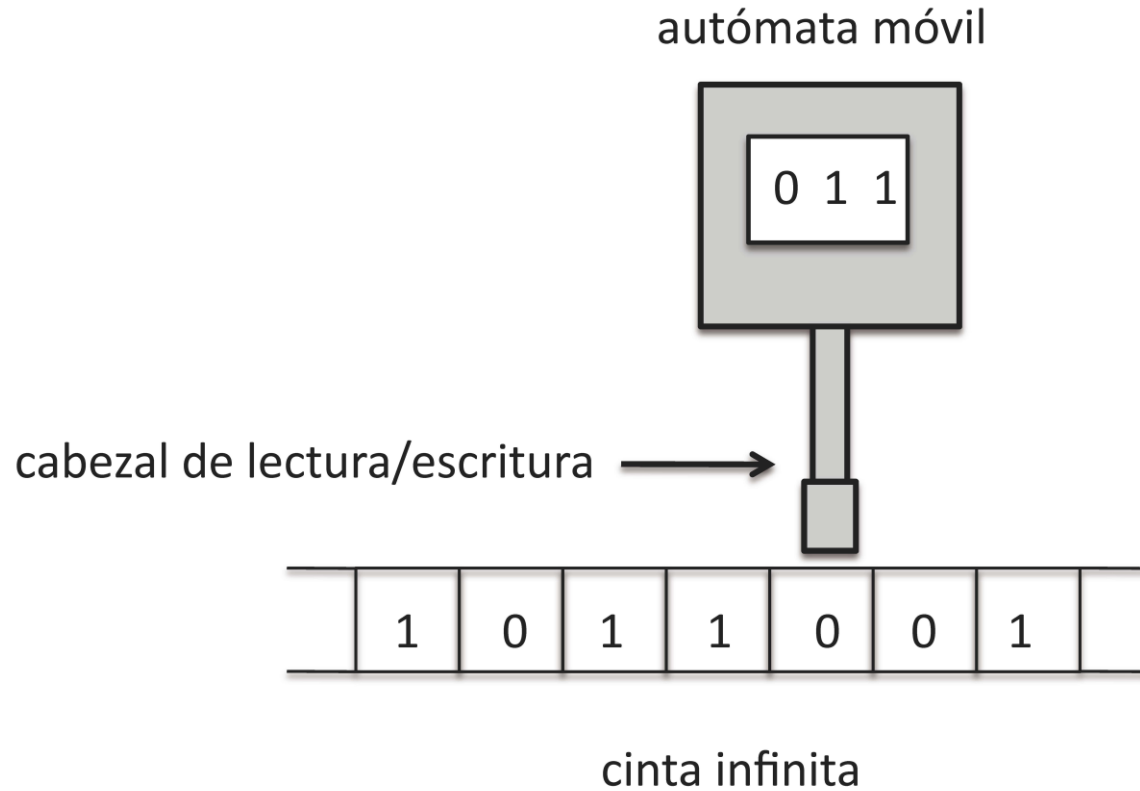
³ – Soda delivered and change returned

⁴ – All inserted money is returned

Ejemplo: máquina expendedora



Máquina de Turing



¿Qué crees que puede hacer?

Algo de historia

- Teoría de autómatas: estudio de “máquinas” o dispositivos abstractos con capacidad de computación
- Turing (1936): estudio de una máquina abstracta
 - determinar la frontera entre lo que se puede y no se puede hacer con un computador
 - máquina de Turing
- 1940, 1950: autómatas finitos
- Finales de los 50: estudio de las gramáticas formales (Chomsky)
- Cook (1971): separa los problemas que se pueden resolver eficientemente de los que no (intratables)

Utilidad

Autómatas [de número de estados] finito

- Software para el diseño y verificación del comportamiento de circuitos digitales
- Analizador léxico de un compilador
- Software para explorar textos buscando la aparición de ciertos patrones
- Software para comprobar el funcionamiento de un sistema con un número finito de estados diferentes (protocolos de comunicación, protocolos de intercambio seguro de información...)

Utilidad

- Gramáticas
 - Descripción de analizadores sintácticos (*parser*)
- Expresiones regulares
 - Especificación de patrones de cadenas
 - Diseño del software de verificación del formato del texto en formularios web
- Autómatas y complejidad
 - ¿Qué puede hacer una computadora? Decidibilidad y computabilidad
 - ¿Qué puede hacer una computadora eficientemente? Complejidad

¿Recuerdas la teoría de conjuntos?

- $x \in X, x \notin X$
- $X = \{1, 2, 3\}, X = \{x \mid x \in \mathbb{N} \text{ y } x \leq 3\}$
- $X \subseteq \mathbb{N}$
- $X \cup Y = \{z \mid z \in X \text{ o } z \in Y\}$
- $X \cap Y = \{z \mid z \in X \text{ y } z \in Y\}$
- $X - Y = \{z \mid z \in X \text{ y } z \notin Y\}$
- Complemento de X respecto a U : $\overline{X} = U - X$
- Leyes de DeMorgan: $\overline{(X \cup Y)} = \overline{X} \cap \overline{Y} \quad \overline{(X \cap Y)} = \overline{X} \cup \overline{Y}$
- Cardinalidad: tamaño de un conjunto, $\text{card}(X)$
 - finito
 - infinito
 - contable o numerable: correspondencia uno a uno con los números naturales
 - incontable

Alfabetos y palabras

- Alfabeto: conjunto finito no vacío de símbolos, Σ
- Cadena o palabra: secuencia finita de símbolos pertenecientes a un alfabeto
 - cadena vacía: λ , ϵ
 - Longitud de la cadena: $|w|$
- Potencias de un alfabeto: conjunto de todas las cadenas de una cierta longitud que se pueden formar con un alfabeto
 - Σ^k
 - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$ (clausura positiva)
 - $\Sigma^* = \Sigma^+ \cup \{\lambda\}$ (cierre o clausura)

Operaciones con palabras

- **Concatenación de cadenas:** sean x e y dos cadenas. xy es la concatenación de x e y
- **Potencia:** la potencia i -ésima de una palabra x , x^i , se forma por la concatenación i veces de x
- **Reflexión:** si la palabra x está formada por los símbolos $A_1 \dots A_n$, entonces la palabra inversa de x , x^{-1} , se forma invirtiendo el orden de los símbolos en la palabra. $x^{-1} = A_n \dots A_2 A_1$

Lenguajes

- Dado un alfabeto Σ , cualquier $L \subseteq \Sigma^*$ será un lenguaje de Σ
- El alfabeto sobre el que se define el lenguaje debe ser finito, aunque el lenguaje puede tener un número infinito de cadenas
- Ejemplos:
 - Lenguaje vacío: \emptyset
 - Lenguaje que contiene únicamente la cadena vacía: $\{\lambda\}$
 - $L = \{w \mid w \text{ contiene el mismo número de ceros y unos}\}$
 - $L = \{0^n 1^n \mid n \geq 1\}$
 - $L = \{a^i b^j c^k \mid j = i+k \text{ e } i, j, k > 0\}$

Operaciones con lenguajes

- **Unión:** $L_1 \cup L_2$, contendrá todas las palabras que pertenezcan a cualquiera de ellos
- **Intersección:** $L_1 \cap L_2$, contendrá todas las palabras que pertenezcan a ambos
- **Resta:** $L_1 - L_2$, contendrá todas las palabras que pertenezcan a L_1 y no a L_2
- **Concatenación:** $L_1 \cdot L_2$, contendrá todas las palabras que se puedan formar por la concatenación de una palabra de L_1 y otra de L_2

Operaciones con lenguajes

- **Potencia:** la potencia i -ésima de un lenguaje es la concatenación i veces del lenguaje consigo mismo
- **Reflexión:** está formada por la aplicación de la reflexión a cada una de las palabras del lenguaje. Se representa por L^r

Gramáticas y autómatas

- Una **gramática** establece la estructura de un lenguaje, es decir, las sentencias que lo forman, proporcionando las formas válidas en que se pueden combinar los símbolos del alfabeto
- Chomsky: clasificación de las gramáticas
 - G0 o de Tipo 0: gramáticas sin restricciones
 - G1 o de Tipo 1: gramáticas sensibles al contexto
 - G2 o de Tipo 2: gramáticas independientes del contexto
 - G3 o de Tipo 3: gramáticas regulares
- $G3 \subseteq G2 \subseteq G1 \subseteq G0$

Gramáticas



Noam Chomsky
(Filadelfia, 7 de
diciembre de
1928), lingüista,
filósofo, politólogo
y activista
estadounidense

“En solo cinco años, de 1953 a 1957, un licenciado de la Universidad de Pensilvania, un estudiante de doctorado de poco más de 20 años, se había apoderado de todo un ámbito de estudio, la lingüística, transformándola de arriba abajo, endureciendo esa presunta ciencia social tan esponjosa y convirtiéndola en una ciencia de verdad, una ciencia dura, a la que puso su nombre: Noam Chomsky”.

Tom Wolfe, 2018

Tom Wolfe



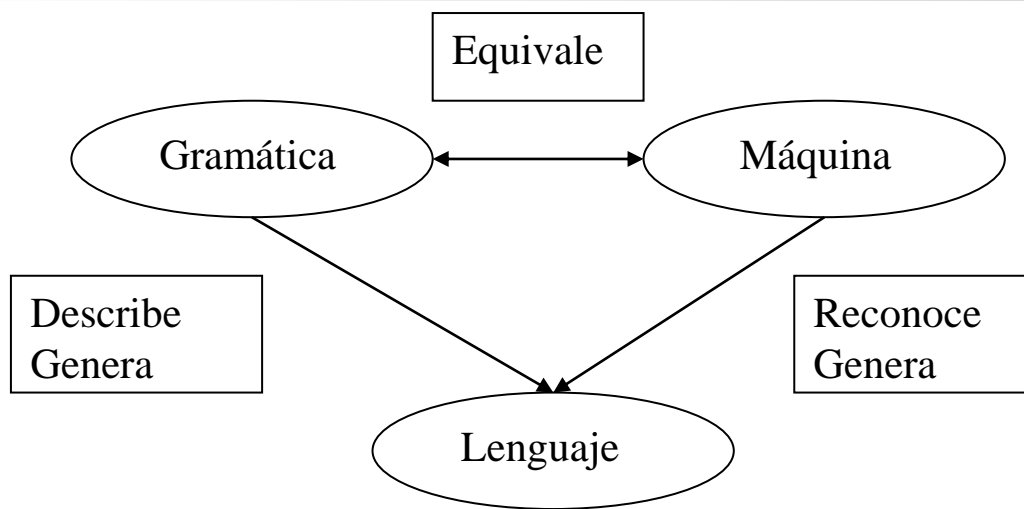
***El reino
del lenguaje***


ANAGRAMA
Colección Argumentos

Gramáticas y autómatas

- Cada gramática es capaz de generar un tipo de lenguaje
 - El lenguaje L será de tipo “ i ” ($i=0,1,2,3$) si existe una gramática de tipo “ i ” capaz de generar o describir dicho lenguaje
- **Máquina abstracta o autómata:** dispositivo teórico capaz de recibir y transmitir información. Dada una cadena de símbolos presentados a su entrada, produce una cadena de símbolos a la salida, en función de dichas entradas y los estados internos por los que transita la máquina.

Lenguajes, gramáticas y autómatas



Gramática	Lenguaje	Máquina / Complejidad
Tipo 0: sin restricciones	Recursivamente enumerable	Máquina de Turing (MT) / Indecible
Tipo 1: sensible al contexto	Sensible al contexto	Autómata Linealmente Acotado (ALA) / Exponencial
Tipo 2: contexto libre o independiente del contexto	Contexto libre o independiente del contexto	Autómata con Pila (AP) / Polinómica
Tipo 3: regular	Regular	Autómata Finito (AF) / Lineal