Tema 1: Introducción

Bibliografía

- Hopcroft, J. E., Motwani, R., y Ullman, J. D. “Introducción a la Teoría de Autómatas, Lenguajes y Computación”. Addison Wesley. 2002.
  - capítulo 1
- Sudkamp, Thomas A. “Languages and machines : an introduction to the theory of computer science”. Addison-Wesley Publishing Company, 1998.
  - capítulos 1 y 2

Un poco de historia

- Teoría de autómatas: estudio de “máquinas” o dispositivos abstractos con capacidad de computación
- Turing (1930): 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 (1969): separa los problemas que se pueden resolver eficientemente de los que no (intratables)

Utilidad de la teoría de autómatas

- Autómatas finitos
  – Software para el diseño y verificación del comportamiento de circuitos digitales
  – Analizador léxico de un compilador
  – Software para explorar texto buscando la aparición de 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 de la teoría de autómatas (II)

- Gramáticas
  - analizador sintáctico (parser)
- Expresiones regulares
  - patrones de cadenas
- Autómatas y complejidad
  - ¿Qué puede hacer un computador?: decidibilidad
  - ¿Qué puede hacer un computador eficientemente?: intratabilidad

Teoría de conjuntos

- $x \in X$, $x \notin X$
- $X = \{1, 2, 3\}$, $X = \{x \mid x \in N \text{ y } x \leq 3\}$
- $X \subseteq N$
- $X \cup Y = \{z \mid z \in X \text{ o } z \in Y\}$
- $X \cap Y = \{z \mid z \in X \text{ y } z \in Y\}$
- $X - Y = \{z \mid z \in X \text{ y } z \notin Y\}$
- Complemento de X respecto a U: $\overline{X} = U - X$
- Leyes de DeMorgan
  - $\overline{(X \cup Y)} = \overline{X} \cap \overline{Y}$
  - $\overline{(X \cap Y)} = \overline{X} \cup \overline{Y}$
- Cardinalidad: tamaño de un conjunto, card(X)
  - finito
  - infinito
    - contable o numerable: correspondencia 1 a 1 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: $\in$, 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
  - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$
  - $\Sigma^* = \Sigma^+ \cup \{\in\}$

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 $A_1 \ldots A_n$, entonces la palabra inversa de *x*, x^{-1}, se forma invirtiendo el orden de los símbolos en la palabra. $x^{-1} = A_n \ldots A_2 A_1$

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: $\varnothing$
  - Lenguaje que contiene únicamente la cadena vacía: $\{\in\}$
  - $L = \{w \mid w$ contiene el mismo número de ceros y unos$\}$
  - $L = \{0^n 1^n \mid n \geq 1\}$
  - $L = \{a^i b^j c^k \mid j = i+k$ e $i, j, k > 0\}$

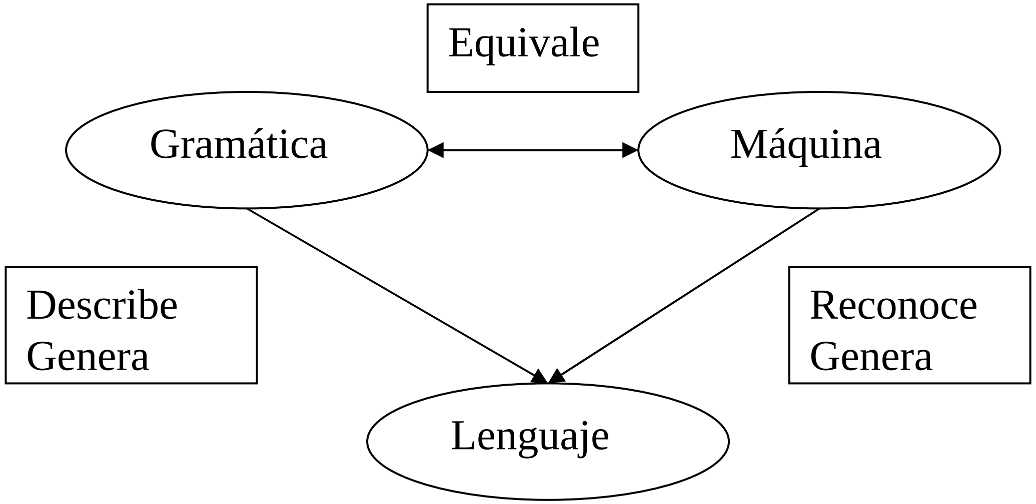
Operaciones con lenguajes

- **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:** $L_1 . L_2$, contendrá todas las palabras que se puedan formar por la concatenación de una palabra de L_1 *y* otra de L_2
- **Potencia:** la potencia i-ésima de un lenguaje es la concatenación i veces del lenguaje consigo mismo
  - Σ^*: cierre o clausura
  - Σ^+: clausura positiva
- **Reflexión:** está formada por la aplicación de la reflexión a cada una de las palabras del lenguaje. Se representa por L^-

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
- $G3 \subseteq G2 \subseteq G1 \subseteq G0$
- 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
Tipo 0: sin restricciones	Sin restricciones	Máquina de Turing (MT)
Tipo 1: sensible al contexto	Sensible al contexto	Autómata Linealmente Acotado (ALA)
Tipo 2: contexto libre o independiente del contexto	Contexto libre o independiente del contexto	Autómata con Pila (AP)
Tipo 3: regular	Regular	Autómata Finito (AF)