

# TeoriaComputacion\_1

October 30, 2019

Teoría de la Computación

Introducción

Prof. Wladimir Rodriguez

wladimir.rodriguez@outlook.com

Departamento de Computación

## 0.1 Objetivo General

- El estudio de los conceptos fundamentales de la teoría de autómatas y lenguajes formales. Conocer la jerarquía de modelos de máquinas computacionales y su funcionamiento, la jerarquía de las gramáticas formales y de los lenguajes correspondientes. Conocer la correspondencia entre máquinas, gramáticas y lenguajes.

## 1 Objetivos Específicos

- Conocer los conceptos fundamentales de la teoría de autómatas y lenguajes formales.
- Conocer la jerarquía de modelos de máquinas computacionales.
- Comprender el funcionamiento y estructura de cada una de las máquinas computacionales.
- Conocer y comprender la jerarquía de gramáticas formales y de los lenguajes correspondientes.
- Conocer la relación existente entre máquinas computacionales, gramáticas y lenguajes.

## 2 Programa

1. Introducción y Preliminares
2. Automatas de estados finitos
  1. Autómatas finitos.
  2. Expresiones regulares.
  3. Conjuntos regulares.
3. Gramáticas de Libre Contexto
  1. Definición y propiedades
  2. Arboles de derivación y formas normales
4. Lenguajes de Contexto Libre

1. Autómatas a Pila.
2. Propiedades de los Lenguajes de Libre Contexto.
5. Máquinas de Turing
  1. Modelo, Lenguajes y Funciones Computables
  2. Técnicas para la Construcción de Máquinas de Turing

### 3 Evaluación

- Tareas: Una tarea cada semana (25%)
- Tres exámenes parciales:
  - Primer Parcial: Temas 1 y 2 (Preliminares y autómatas finitos) (25%)
  - Segundo Parcial: Tema 3 (Gramáticas de libre contexto) y Tema 4 (Autómatas a Pilas) (25%)
  - Tercer Parcial: Tema 5 (Máquinas de Turing) (25%)

### 4 Recursos

- Repositorio en GitHub: <https://github.com/wladrod/TeoriaComputacion2019.git>
- Software JFLAP versión 7.1: <http://www.jflap.org/jflaptmp/>
- Jove: Jupyter Notebooks for Automata and Computability.  
<https://github.com/ganeshutah/Jove>

### 5 Repositorio en GitHub

### 6 JFLAP

JFLAP es un software para experimentar con temas de lenguajes formales que incluyen autómatas finitos no deterministas, autómatas a pila, Máquinas Turing.

### 7 Jove

Jove es la colección de notebooks de Jupyter creada por Ganesh Gopalakrishnan para enseñar Modelos de Computación.

### 8 Bibliografía

- Autómatas y Lenguajes. Ramón Brena, Tecnológico de Monterrey, 2003.
- Lenguajes, Gramáticas y Autómatas. Juan M. Cueva, Segunda Edición, Oviedo, 2001.
- Notas del Curso CCO-500 lenguajesFormales y Autómatas, José Palacios, Universidad Autónoma de Puebla. 2002.

- Teoría de Autómatas y Lenguajes Formales. Serafín Moral, Granada, 2001.
- Introduction to the Theory of Computation. Michael Sipser, Third Edition, 2013.
- Automata and Computability. Ganesh Gopalakrishnan, 2019.

## 9 Introducción

- Ciencias de la Computación:
  - Fundamentos de las Matemáticas (Teoría de la Computabilidad)
    - \* Trata del estudio de la computabilidad y de la complejidad.
- Teoría de Máquinas (Teoría de Autómatas)
  - Trata del estudio de máquinas de computación abstractas.
- Lingüística (Teoría de Lenguajes y las Gramáticas)
  - Trata del estudio y formalización de los lenguaje

## 10 Teoría de la Computación

- En 1928, el matemático alemán David Hillbert, presenta a sus colegas como reto demostrar tres proposiciones de gran generalidad sobre su ciencia:
  - ¿Son completas las matemáticas, en el sentido de que pueda probarse o no cada aseveración matemática?
  - ¿Son las matemáticas consistentes, en el sentido de que no pueda probarse simultáneamente una aseveración y su negación?
  - ¿Son las matemáticas decidibles, en el sentido de que exista un método definido que se pueda aplicar a cualquier aseveración matemática, y que determine que dicha aseveración es cierta?
- En 1928, el matemático Ackermann presenta también el problema de la decisión en un estudio sobre la lógica de primer orden.
- En 1930, el matemático checo Kurt Gödel prueba que las matemáticas no pueden ser completas y consistentes al mismo tiempo.
- En 1936, el matemático americano Alonzo Church responde negativamente en un artículo al tercer problema propuesto por Hillbert, el problema decisorio.
- En ese mismo año, el matemático inglés Alan Turing da una respuesta también negativa a esa tercera cuestión, sus resultados son más consistentes que los obtenidos por Church.
- La demostración de Turing se basa en principios completamente básicos y elementales.
- Turing enuncia el problema de decisión de la siguiente forma:
  - *Buscar un algoritmo para determinar si una conclusión particular puede derivarse de ciertas premisas con el uso de reglas de prueba*
- Da una noción precisa del concepto de algoritmo, como aquello que pueda ser realizado por una máquina abstracta  $\rightarrow$  Máquina de Turing.
- La máquina abstracta, Máquina de Turing se convierte en la herramienta de Turing para el estudio de la computabilidad.
- Estas máquinas tienen una capacidad muy especial, que es la capacidad de imitar cualquier otra máquina  $\rightarrow$  Máquina Universal de Turing.

- Demostrando que existen problemas para los que se puede construir una Máquina de Turing pero esta no da solución, no se detiene.
- De este modo, Alan Turing con su máquina universal capaz de realizar el trabajo de cualquier otra máquina, mediante la lectura de su descripción en una cinta, delinea el diseño de un computador.
- En 1969, S. Cook extiende el estudio de Turing. Cook separa aquellos problemas que pueden ser solucionados de aquellos que en principio pueden ser solucionados pero que en la práctica toman demasiados recursos.
- La Máquina de Turing ayuda a comprender que es lo que podemos esperar de nuestro software.

## 11 Ramas de la Computación

- En el campo de la computación se distinguen dos ramas, que estudian dos problemas:
  - *Computabilidad*, estudian aquellos problemas que pueden ser resueltos por medio de un computador.
  - *Complejidad*, estudian la complejidad de aquellos problemas que pueden ser resueltos por un computador.

## 12 Teoría de Máquinas o Autómatas

- En 1937, Claude Shannon presenta su tesis de licenciatura en el MIT, estableciendo el paralelismo entre la lógica de Boole y los circuitos de transmisión.
- Las ideas de esta tesis se desarrollan notablemente dando lugar a la formalización de la teoría de las máquinas secuenciales y de los autómatas finitos.
- Todo sistema que acepta señales del medio que le rodea y, como resultado, cambia de estado y transmite otras señales al medio, puede considerarse un autómata.
- En 1943, McCulloch-Pitts desarrolla unas máquinas simples, en cuanto su funcionamiento, que fueron conocidas como autómatas finitos de actividad nerviosa, para modelar el funcionamiento de una neurona biológica.
- En 1956, Moore publica el primer estudio riguroso sobre autómatas. Con anterioridad, debido al desarrollo de los primeros ordenadores, se habían estudiado diversos métodos para la síntesis de circuitos secuenciales (Huffman, 1954 y Mealy, 1955).
- En los años 60 es donde se realizan la mayor parte de trabajos sobre teoría de autómatas finitos
- La teoría de máquinas o autómatas, trata de la invención y estudio de autómatas, incluyendo el estudio de las capacidades y limitaciones en el proceso de computo, la manera en la que los sistemas reciben entradas, las procesan y producen salidas y las relaciones entre las teorías de la conducta y la operación y uso de mecanismos automáticos.

## 13 Campos de Aplicación de los Autómatas

- Teoría de la comunicación.

- Teoría de control.
- Lógica de los Circuitos Secuenciales.
- Reconocimiento de patrones.
- Computadoras.
- Lógica de los sistemas evolutivos y autoreproductivos.
- Traducción de lenguajes.
- Redes conmutadoras y codificadoras.

## 14 Lingüística

- En 1956, N. Chomsky comienza el estudio formal de las gramáticas (como generadoras de lenguajes).
- Formaliza matemáticamente los conceptos de gramática generativa o reglas para la construcción las frases de un lenguaje.
- Enuncia la teoría sobre el origen y la naturaleza de los lenguajes naturales.
- Estas herramientas fueron empleadas para la formalización de los lenguajes de ordenador, Teoría de los Lenguaje Formales.
- A finales de los años 50 se relacionan los autómatas con los lenguajes.

## 15 Utilidad de la Asignatura

- Las Máquinas de Turing son esenciales para el estudio de los límites de la computación:
  - ¿Qué puede hacer un computador? Problema de la decibilidad.
  - ¿Qué puede hacer un computador eficientemente? Problema de la tratabilidad.
- Los Autómatas son modelos útiles para muchos elementos hardware y software (Diseñar y chequear la conducta de circuitos digitales, Reconocimiento de patrones,...).
- Los lenguajes permiten la comunicación con la máquina, lo que puede hacer la máquina depende del poder descriptivo del lenguaje.

## 16 Máquinas Abstractas

- Es una máquina que no existe físicamente, pero que ejecuta instrucciones sobre una entrada y produce una salida, habitualmente se han empleado en el estudio de la computabilidad.

## 17 Ejemplos de Máquinas Abstractas

- Autómatas Finitos.
- Autómatas a Pila.
- Autómatas linealmente acotados.
- Máquina de Turing.

## 18 Autómata Finito

- Son sistemas que en todo momento se encuentran en un estado de entre un conjunto finito de estados.
- El propósito de un estado es recordar la historia del sistema.
- Puesto que el número de estados es finito, el sistema debe ser diseñado para recordar aquello que es importante y olvidar lo que no.
- La ventaja de tener un número finito de estados es que el sistema podrá ser implementado con un conjunto fijo de recursos.

## 19 Preliminares

## 20 Conjuntos

- **Definición:** Un conjunto es una colección no ordenada de objetos o elementos.
  - Los conjuntos se escriben con llaves  $\{\}$ .
  - Los elementos en el conjunto se escriben entre las llaves
- **Definition:**
  - Para cualquier conjunto  $S$ , “ $x \in S$ ” denota que  $x$  es un elemento del conjunto  $S$ .
  - Además, “ $y \notin S$ ” denota que  $y$  no es un elemento del conjunto  $S$ .

**Observación:** a menudo especificamos un conjunto utilizando la notación de conjunto, por ejemplo,

$$\{x | x \in \mathbb{R}, x^2 - 4 = 0\}$$

-  $\mathbb{R}$  denota el conjunto de números reales. -  $|$  Significa “tal que” - Coma significa “y”

## 21 Ejemplos de conjuntos

- El conjunto  $\{a, b, c\}$  tiene elementos  $a$ ,  $b$  y  $c$ .
- Los conjuntos  $\{a, b, c\}$  y  $\{b, c, b, a, a\}$  son iguales
  - El orden y la redundancia no importan en un conjunto
- El conjunto  $\{a\}$  tiene el elemento  $a$ .
  - $\{a\}$  y  $a$  son cosas diferentes.
  - $\{a\}$  es un conjunto con un elemento  $a$ .

- El conjunto  $\mathbb{Z}$  de **enteros** es

$$\{\dots, -2, -1, 0, 1, 2, \dots\}$$

- El conjunto  $\mathbb{Z}_+$  de enteros no negativos es

$$\{0, 1, 2, 3, \dots\}$$

- El conjunto de números pares es

$$\{0, 2, 4, 6, 8, 10, 12, \dots\}$$

, que también podemos escribir como

$$\{2n | n = 0, 1, 2, \dots\}$$

- El conjunto de números impares

$$\{1, 3, 5, 7, 9, 11, 13, \dots\}$$

también se puede escribir como

$$\{2n + 1 | n = 0, 1, 2, \dots\}$$

**Ejemplo:** si  $A$  es el conjunto  $\{2n | n = 0, 1, 2, \dots\}$ , entonces  $4 \in A$ , pero  $5 \notin A$ .

## 22 Alfabetos

Un alfabeto es un conjunto finito de unidades fundamentales (llamadas letras o símbolos).

**Observación:** normalmente denotamos un alfabeto con una letra griega mayúscula - por ejemplo,  $\Sigma$  o  $\Gamma$  (es decir, Sigma o Gamma)

**Ejemplos** - El alfabeto de las 26 letras romanas en minúscula es

$$\Sigma = \{a, b, c, \dots, z\}$$

- El alfabeto de las 26 letras romanas mayúsculas es

$$\Gamma = \{A, B, C, \dots, Z\}$$

- El alfabeto de los 10 números arábigos es

$$\Sigma = \{0, 1, 2, \dots, 9\}$$

- En esta clase usaremos a menudo los alfabetos

$$\Sigma = \{a, b\}$$

$$\Sigma = \{0, 1\}$$

## 23 Secuencias y cadenas

**Definición:** La secuencia de objetos es una lista de estos objetos en algún orden. - El orden y la redundancia importan en una secuencia, a diferencia de un conjunto. -  $a, b, c$  y  $b, c, b, a, a$  son secuencias diferentes. -  $\{a, b, c\}$  y  $\{b, c, b, a, a\}$  son el mismo conjunto.

**Definición:** una cadena sobre un alfabeto es una secuencia finita de símbolos del alfabeto (escritos sin comas o espacios entre los símbolos)

## 24 Ejemplos

- $x, trolulent, amziixen$  y  $qwertyuio$  cadenas sobre el alfabeto

$$\Sigma = \{a, b, c, \dots, z\}$$

- 0131 es una cadena sobre el alfabeto

$$\Sigma = \{0, 1, 2, \dots, 9\}$$

## 25 Longitud de una cadena

**Definición:** La longitud de una cadena  $w$  es el número de símbolos en  $w$ . - Algunas veces se denota la longitud de  $w$  con  $\text{longitud}(w)$  o  $|w|$ .

**Ejemplo:**  $\text{longitud}(\text{casa}) = |\text{casa}| = 4$

**Definición:** La cadena vacía o cadena nula, denotada por  $\epsilon$  (es decir, épsilon), es la cadena que no consta de símbolos, es decir,

$$|\epsilon| = 0$$

## 26 Estrella de Kleene

**Definición:** para un alfabeto dado  $\Sigma$ , deje que  $\Sigma^*$  denote el conjunto de todas las cadenas posibles (incluyendo  $\epsilon$ ) sobre  $\Sigma$ .

**Ejemplo:** si  $\Sigma = \{a, b\}$ , entonces

$$\Sigma^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, \dots\}$$

## 27 Ordenamiento de Cadenas

**Definición:** una lista de cadenas  $w_1, w_2, \dots$  sobre un alfabeto  $\Sigma$  está en orden de cadena (también llamado orden lexicográfico) si 1. las cadenas más cortas siempre aparecen antes que las cadenas más largas, y 2. las cadenas de la misma longitud aparecen en orden alfabético.

**Ejemplo:** si  $\Sigma = \{0, 1\}$ , el orden de las cadenas en  $\Sigma^*$  es

$$\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \dots$$

## 28 Concatenación

**Definición:** La concatenación de las cadenas  $x$  e  $y$  es la cadena  $xy$ .

**Ejemplos:**

- Si  $x = \text{luna}$  e  $y = \text{sol}$ , entonces  $xy = \text{lunasol}$  e  $yx = \text{solluna}$ .
- Si  $x = \epsilon$  e  $y = ab$ , entonces  $xy = ab = yx$ .
- Si  $x = \epsilon$  e  $y = \epsilon$ , entonces  $xy = \epsilon = yx$ ; es decir,  $\epsilon\epsilon = \epsilon$ .

**Definición:** para la cadena  $w$ , definimos  $w^n$  para  $n \geq 0$  inductivamente como

- $w^0 = \epsilon$ ;
- $w^n = w^{n-1}w$  para cualquier  $n \geq 1$ .



**Ejemplo:** si  $w = casa$ , entonces

$$w^0 = \epsilon, w^1 = w^0 w = \epsilon casaw^2 = w^1 w = casacasaw^3 = w^2 w = casacasacasa...$$

**Ejemplo:** también se puede aplicar esto a un solo símbolo

- $a^3 = aaa$
- $a^0 = \epsilon$

## 29 Subcadenas

**Definición:** Una subcadena de una cadena  $w$  es cualquier parte contigua de  $w$ . - es decir,  $y$  es una subcadena de  $w$  si existen cadenas  $x$  y  $z$  (una o ambas posiblemente vacío) tal que  $w = xyz$ .

**Ejemplos:**

- $y = 47$  es una subcadena de  $w = 472$  ya que permite que  $x =$  y  $z = 2$  da  $w = xyz$ .
- La cadena 472 tiene subcadenas , 4, 7, 2, 47, 72 y 472.
- 42 no es una subcadena de 472.

[ ]: