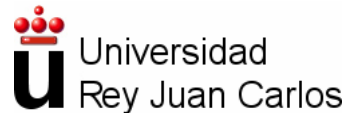


# *Teoría de Autómatas y Lenguajes Formales*

## *Capítulo 1: “Introducción”*

**Holger Billhardt**  
holger.billhardt@urjc.es



## Introducción

- Teoría de Autómatas y Lenguajes formales es un “repaso” a la informática teórica.
- La informática teórica:
  - se ha desarrollado en base a la confluencia de campos en apariencia muy distintos:
    - Investigación acerca de Fundamentos Matemáticos,
    - Teoría de Máquinas,
    - Lingüística, ...
  - Ciencia multidisciplinar que se apoya en que los mismos fenómenos pueden actuar y servir de fundamento en áreas totalmente desconectadas (aparentemente).

## Introducción

### ■ Pilares de la informática teórica:

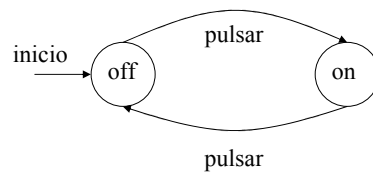
- Autómatas / máquinas secuenciales
- Lenguajes y gramáticas
- Máquinas abstractas y algoritmos

## Autómatas / Máquinas secuenciales

- Eslabón de la informática teórica que proviene de la Ingeniería Eléctrica.
- 1938 – Claude Elwood Shannon: “A symbolic Analysis of relay and switching circuits”
  - Aplicación de la lógica matemática a los circuitos combinatorios y secuenciales.
- Sus ideas desarrollaron la Teoría de los autómatas finitos y máquinas secuenciales
  - Un autómata es un dispositivo abstracto que es capaz de recibir información, cambiar de estado y transmitir información.

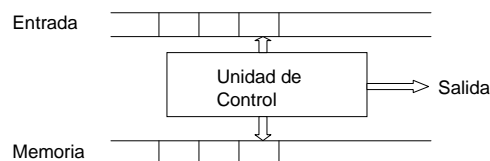
## Autómatas / Máquinas secuenciales

- Un autómata puede describir de forma formal el funcionamiento de un sistema
  - Ejemplo: interruptor



## Autómatas / Máquinas secuenciales

- Un autómata es un modelo abstracto de una computadora digital
  - Lee símbolos en la entrada
  - Produce símbolos en la salida
  - Tiene una unidad de control que puede estar en uno de sus posibles estados internos
  - Puede cambiar de los estados internos en función de la entrada
  - Puede tener algún tipo de memoria
- Autómatas transductores / generadores / aceptadores



## Lenguajes y Gramáticas

- Segundo eslabón: Lingüística (campo tradicionalmente considerado no científico).
- Años 50 – Avram Noam Chomsky
  - Teoría de las Gramáticas Transformacionales
    - Base de la Lingüística Matemática
    - Proporcionó una herramienta que no sólo podía aplicarse a los lenguajes naturales, sino que facilitaba el estudio y formalización de los lenguajes de ordenador que aparecían en aquella época.

## Lenguajes y Gramáticas

- Cualquier comunicación se realiza mediante cadenas de símbolos que corresponden a un lenguaje.
- Lenguajes son conjuntos de cadenas de símbolos (palabras, oraciones, textos o frases)
- El estudio de los lenguajes se reduce, básicamente, a:
  - **Sintaxis: (gramática)**
    - define las secuencias de símbolos que forman cadenas válidas de un lenguaje
    - Gramática: Descripción formalizada de las oraciones de un lenguaje. Una gramática genera o describe un lenguaje.
  - **Semántica:**
    - significado de las cadenas que componen un lenguaje

# Lenguajes y Gramáticas

## ■ Ejemplo 1:

- Semántica:
  - A es un número natural.
- Diferente sintaxis en diferentes lenguajes:
  - A is a natural number.
  - A : Natural;
  - 0100000100000001

# Lenguajes y Gramáticas

## ■ Ejemplo 2:

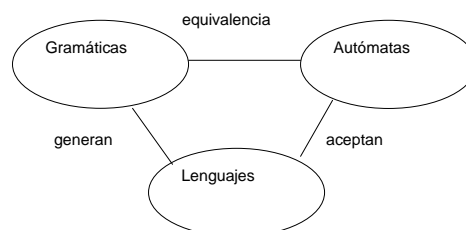
- Sintaxis:
  - `if_statement ::= if condition then sequence_of_statement {elsif condition then sequence_of_statements} [else sequence_of_statements] end if;`
- Semántica:
  - Si se cumple `<condition>` entonces haz lo que viene definido por `<sequence_of_statements>`. En caso contrario ...
- Cadena del lenguaje:
  - `if Line_Too_Short then raise Layout_Error; elsif Line_Full then New_Line; Put(Item); else Put(Item); end if;`

## Lenguajes y Gramáticas

- La Teoría de Lenguajes Formales resultó tener una relación inmediata y directa con la Teoría de Máquinas Abstractas.
  - Se establecieron correspondencias (isomorfismos) entre ellas.
  - Se puede describir el funcionamiento de sistemas mediante gramáticas → [lenguajes de programación](#)
- Chomsky clasificó de las gramáticas en diferentes tipos:
  - Lenguajes del mismo tipo tienen propiedades en común
  - Según el tipo de lenguaje, existen diferentes algoritmos que permiten comprobar la sintaxis de textos.

## Lenguajes y Gramáticas

- Relación entre autómatas, lenguajes y gramáticas:
  - Autómatas **aceptadores**: las entradas válidas corresponden a un lenguaje



## Máquinas abstractas y algoritmos

- La historia de la informática teórica se remonta a la década de los 30.
- 1931 - **Kurt Gödel**: “*On formally undecidable Propositions in Principia Mathematica and related systems*”
  - Revolución Matemática: “*Cualquier teoría matemática ha de ser incompleta. Siempre habrá en ella afirmaciones que no se podrán demostrar ni negar.*”

## Máquinas abstractas y algoritmos

- ¿Cómo se puede formalizar el concepto de realizar un cálculo?
  - 1937 – **Alan Mathison Turing**: “*On computable numbers with an application to the Entscheidungsproblem*”
  - Definición de la **Máquina de Turing como dispositivo matemático abstracto de cálculo que introduce el concepto de “algoritmo”**.
  - Origen “oficial” de la informática teórica.
  - Precursora abstracta de las máquinas de calcular automáticas.
  - La Máquina de Turing es un modelo abstracto de los ordenadores actuales.
  - Demuestra la **existencia de problemas irresolubles**, los que ninguna máquina de Turing (y ningún ordenador) puede resolver o calcular. (Teoría de la Computabilidad).

## Desarrollo de la Asignatura

- Conceptos básicos: Lenguajes Formales y Gramáticas
- Lenguajes regulares
- Autómatas Finitos
- Lenguajes Independientes del Contexto
- Autómatas a Pila
- Máquinas de Turing
- Computabilidad

## Notaciones que utilizaremos

- Básicamente operaciones sobre conjuntos.
- **Conjunto:**
  - $\{x \mid \dots\}$ , el conjunto de los  $x$  tales que ...
  - $\{x_1, x_2, x_3, x_4, x_5\}$ , el conjunto compuesto por los elementos  $x_1$  a  $x_5$
- **Pertenencia:**  $x \in C$ , el elemento  $x$  pertenece al conjunto  $C$
- **Inclusión:**  $C \subseteq C'$ , el conjunto  $C$  es un subconjunto del  $C'$
- **Cardinalidad:**  $|C|$ , el número de elementos del conjunto  $C$



## Notaciones que utilizaremos

- **Unión de conjuntos:**  $C \cup C'$ , la unión de los conjuntos  $C$  y  $C'$
- **Intersección:**  $C \cap C'$ , la intersección de los conjuntos  $C$  y  $C'$
- **Simplificación:**  $a...z$  ó  $a, ...,z$  ó  $x_1,...,x_n$  (todos los elementos entre  $x_1$  y  $x_n$ )
- **Aplicación entre conjuntos:**
  - $f: E_1 \times E_2 \times \dots \times E_n \rightarrow S_1 \times S_2 \times \dots \times S_m$   
La función (aplicación)  $f$  está definida entre los conjuntos  $E_i$  y  $S_j$ .

Desde el punto de vista computacional, se puede entender que  $f$  recibe de entrada a un elemento para cada conjunto  $E_i$  y genera una salida para cada conjunto  $S_j$
- Más notaciones se introducirán a lo largo del curso.