

1. Introducción a la Teoría de Autómatas y Lenguajes Formales

Araceli Sanchis de Miguel
Agapito Ledezma Espino
José A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber

Grado Ingeniería Informática
Teoría de Autómatas y Lenguajes Formales

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M. Alonso

(1)



Universidad
Carlos III de Madrid
www.uc3m.es



Objetivos

- Presentar la normativa, los contenidos y objetivos de la asignatura poniendo énfasis en las aplicaciones prácticas de la materia que se va a estudiar.
- Conocer la contextualización histórica de la Teoría de Autómatas y lenguajes formales. Desde los orígenes hasta los distintos campos de los que se ha nutrido esta área de conocimiento (Ingeniería, Lenguajes y Gramáticas, y Matemáticas y Computabilidad).
- Conocer el esquema básico que se seguirá a través de la jerarquía de Chomsky sobre los autómatas, gramáticas y lenguajes formales.
- Conocer otras máquinas abstractas relacionadas que se encuentran fuera de la jerarquía de Chomsky.
- Conocer los límites de las máquinas abstractas que se estudiarán y sus problemas de complejidad.

Índice

- El por qué de la Teoría de Autómatas
- Relación con otras Áreas de Conocimiento
- Máquinas, Lenguajes y Algoritmos

El por qué de la Teoría de Autómatas

Disciplinas de la Computación según la “*Educational Activities Board of IEEE*”:

- Computer Engineering , **Computer Science**, Information Systems, Information Technologies, Software Engineering

Computer Science:

- “A pesar de la enorme amplitud de la informática, existen conceptos y habilidades que son comunes a la informática en su conjunto.”
- “Todos los estudiantes de informática tienen que aprender a integrar la teoría y la práctica, a reconocer la importancia de la abstracción para apreciar el valor del buen diseño de ingeniería”

Fuente: Computing Curricula 2005. The Overview Report.

http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf



El por qué de la Teoría de Autómatas

- **Ciencias de la Computación:** cuerpo de conocimiento que se ocupa del estudio de los fundamentos teóricos de la información y la computación y de su implementación y aplicación en sistemas computacionales.
- Gibbs y Tucker (1986):
 - “No se debe entender que el objetivo de las Ciencias de la Computación sea la construcción de programas sino el estudio sistemático de los algoritmos y estructuras de datos, específicamente de sus propiedades formales”
 - Gibbs, N. E. and Tucker, A. B. 1986. A model curriculum for a liberal arts degree in computer science. Commun. ACM 29, 3 (Mar. 1986), 202-210. DOI= <http://doi.acm.org/10.1145/5666.5667>

El por qué de la Teoría de Autómatas

Primera inmersión en la “Teoría de la Computación”:

- Es anterior al invento del Computador (incluso del transistor)
- Propiedades **MATEMÁTICAS FUNDAMENTALES** de Software, Hardware y aplicaciones de los mismos. Responder a preguntas como:
 - ¿Cómo puede construirse un programa para resolver un problema?
 - ¿Resuelve el programa realmente el problema?
 - Cuánto se tarda en realizar un cómputo (complejidad temporal).
 - Cuanta memoria se necesita para realizar el computo (complejidad espacial).
 - Y el “modelo de computación” (Imperativo, POO, Programación. Lógica, etc.)
 - Qué se puede computar y qué NO se puede computar.



El por qué de la Teoría de Autómatas

Aplicación directa de conceptos propios de las Ciencias de la Computación:

- Videojuegos
 - Comportamiento de personajes
- Compiladores y Procesamiento de Lenguaje Natural
 - Análisis Léxico en lenguajes programación (compilador).
 - Búsqueda de cadenas o comparación de “patrones”
 - Diseño de nuevos lenguajes de programación o ampliación
- Implementación de Protocolos Robustos
 - Para clientes o usuarios
 - E.g. Sistemas de Seguridad
- Criptografía Moderna (sus protocolos)
- ...



El por qué de la Teoría de Autómatas

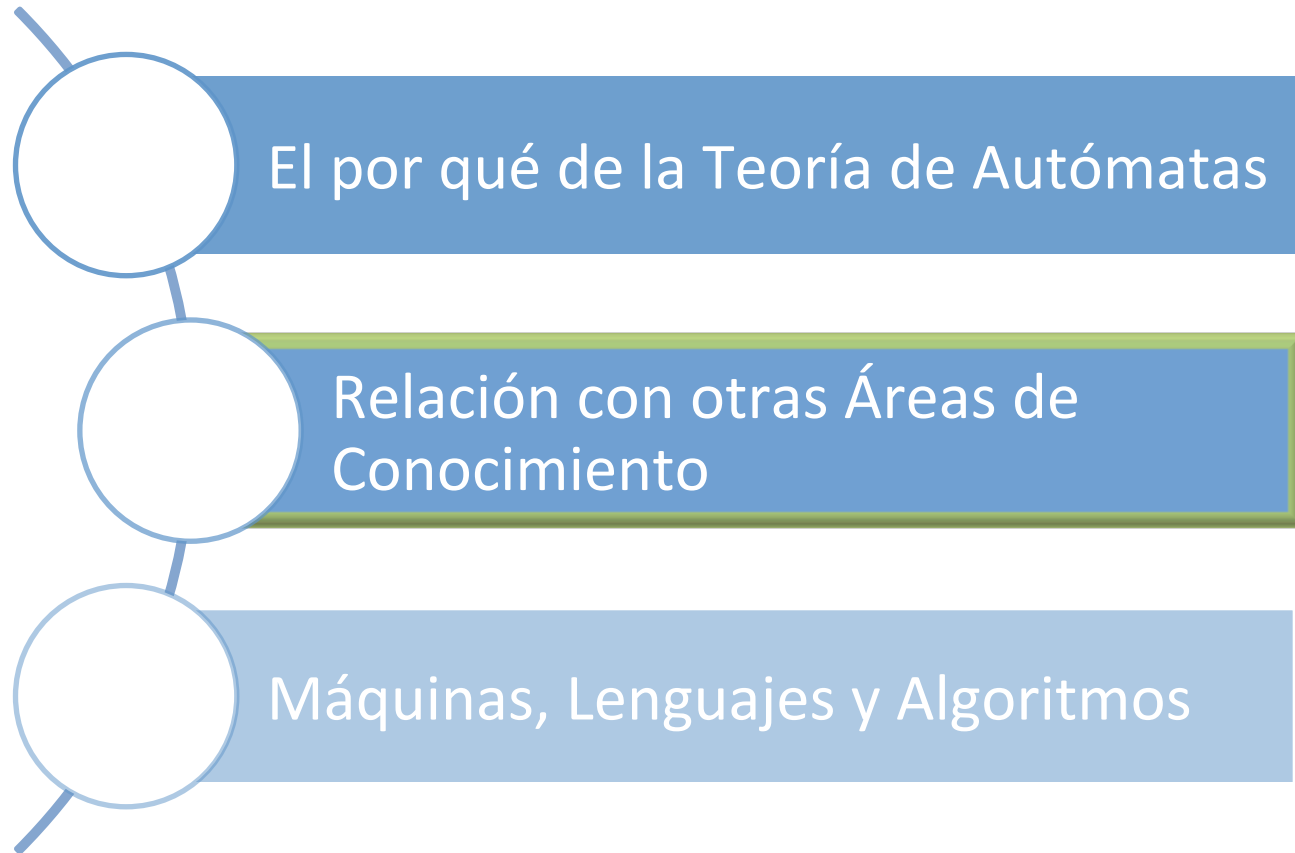
Aplicación directa de conceptos propios de las Ciencias de la Computación:

- ...
- Construcción de sistemas computacionales más elegantes y sencillos.
- Diseño (Maquina Secuencial --> Código)
- Diseño de estructuras y “parsing”: gramaticas (ej: XML)
 - Búsqueda de cadenas o comparación de “patrones”
- SW para diseñar y evaluar circuitos digitales.
- “Escanear” grandes cantidades de texto (web)
- SW para verificar sistemas que tiene un número finito de “estados”

El por qué de la Teoría de Autómatas

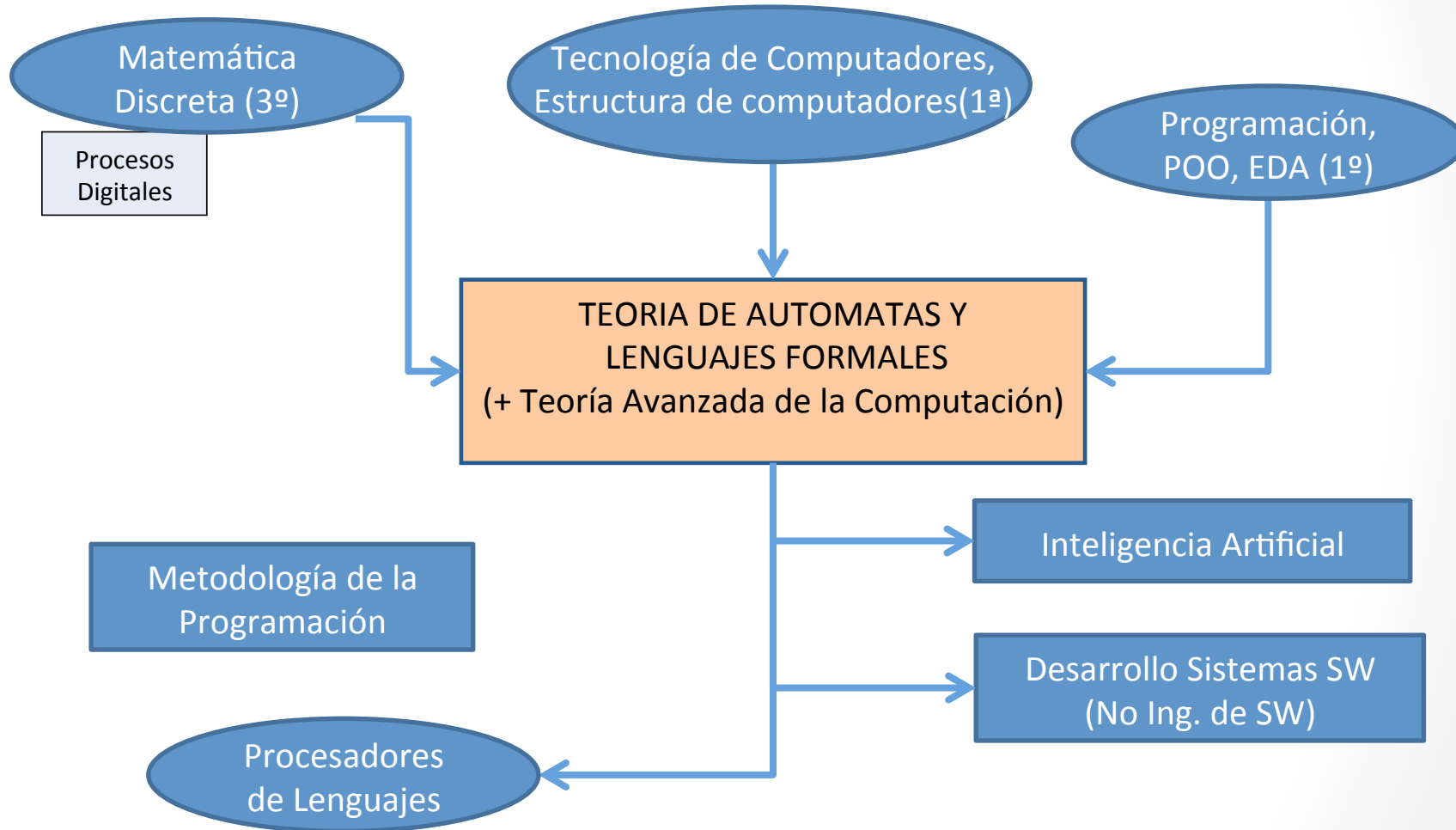
- Teoría de la Computación:
 - ¿Aburrida y arcaica? NO, es Comprensible e Interesante.
- Proporciona al Ingeniero:
 - Aspectos teóricos (permite innovación)
 - Autómatas,
 - Representación Estructural (Gramáticas)
 - Autómatas y Máquinas para establecer límites de la Computabilidad.
 - Aspectos prácticos (ingeniería)

Índice

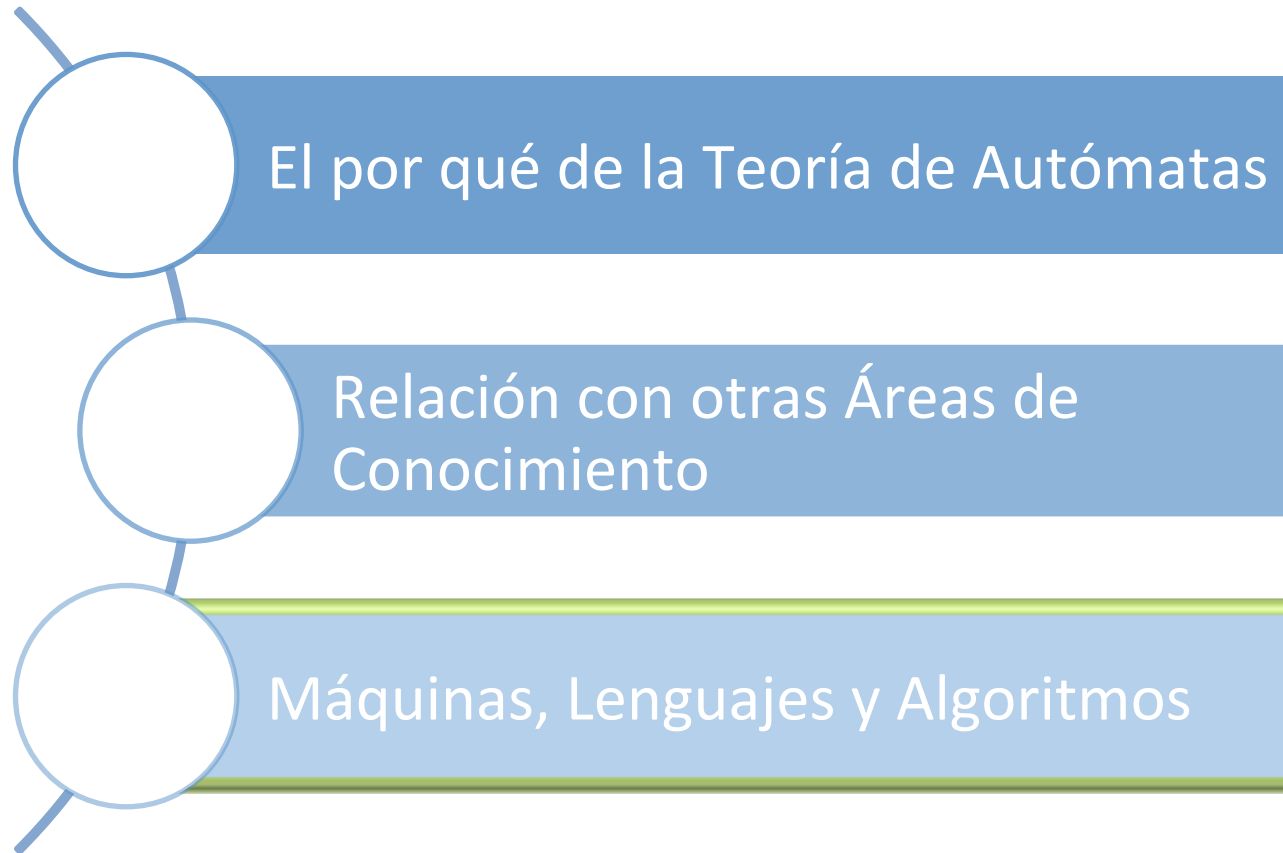


Relación con otras áreas.

Grado en Ingeniería Informática



Índice



Máquinas, Lenguajes y Algoritmos

Tres pilares sustentan la
Teoría de Lenguajes,
Gramáticas y Automatas

AUTÓMATAS (ingeniería)

- Leonardo Torres, 1915
- Shannon, 1938
- Mc Culloch-Pitts, 1943
- Moore, 1956

COMPUTABILIDAD (matemáticas)

- Hilbert, 1928
- Gödel, Kleene, Post y Turing, ≈1930
- Church, 1936
- Rabin, 1960
- Cobhan, 1964
- Cook, 1972
- Aho, Hopcroft, Ullman, 1974

LENGUAJES y GRAMÁTICAS (lingüística)

- Panini, entre el 400 y 200 AC
- Chomsky, 1967
- Backus, ≈1960
- Kleene, 1951
- Hirst, Tennant y Carbonell, 1981

Máquinas, Lenguajes y Algoritmos



Máquinas o
Autómatas

- Aplicación en campos muy diversos
- Manejan conceptos como “control”, “acción”, “memoria”
- Los objetos son controlados o recordados con símbolos, palabras o frases de algún tipo.

- Máquina de Moore y máquina de Mealy
- Circuitos combinatorios
- Autómatas Probabilísticos (incertidumbre en las transiciones)
- McCulloch-Pitts (1943) describieron los cálculos lógicos inmersos en un dispositivo denominado neurona artificial.
 - Redes de Neuronas Artificiales
- Autómatas Celulares (J.H. Conway, el juego de la vida).



Máquina de Turing Universal, Jim Wiked.

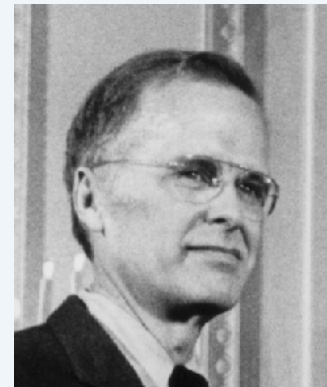
Máquinas, Lenguajes y Algoritmos

Lenguajes y Gramáticas

- Origen en la lingüística
- Noam Chomsky
 - Jerarquía de Chomsky (1956)
- “Backus normal form”
(para gramática de ALGOL)
 - Lenguajes de Programación
 - Lenguajes Naturales
 - Sistemas de Comandos

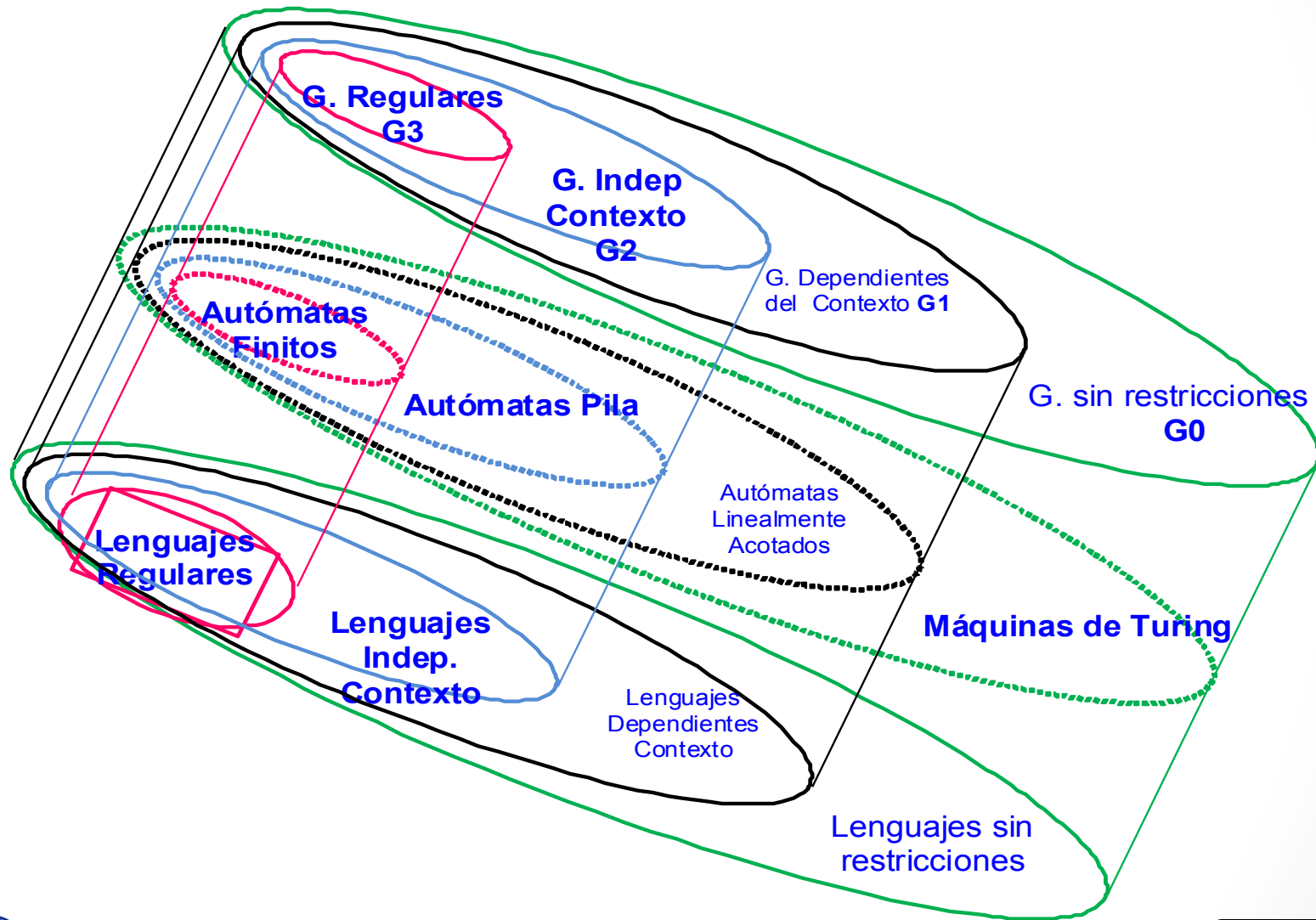


Noam Chomsky
(1928 -)

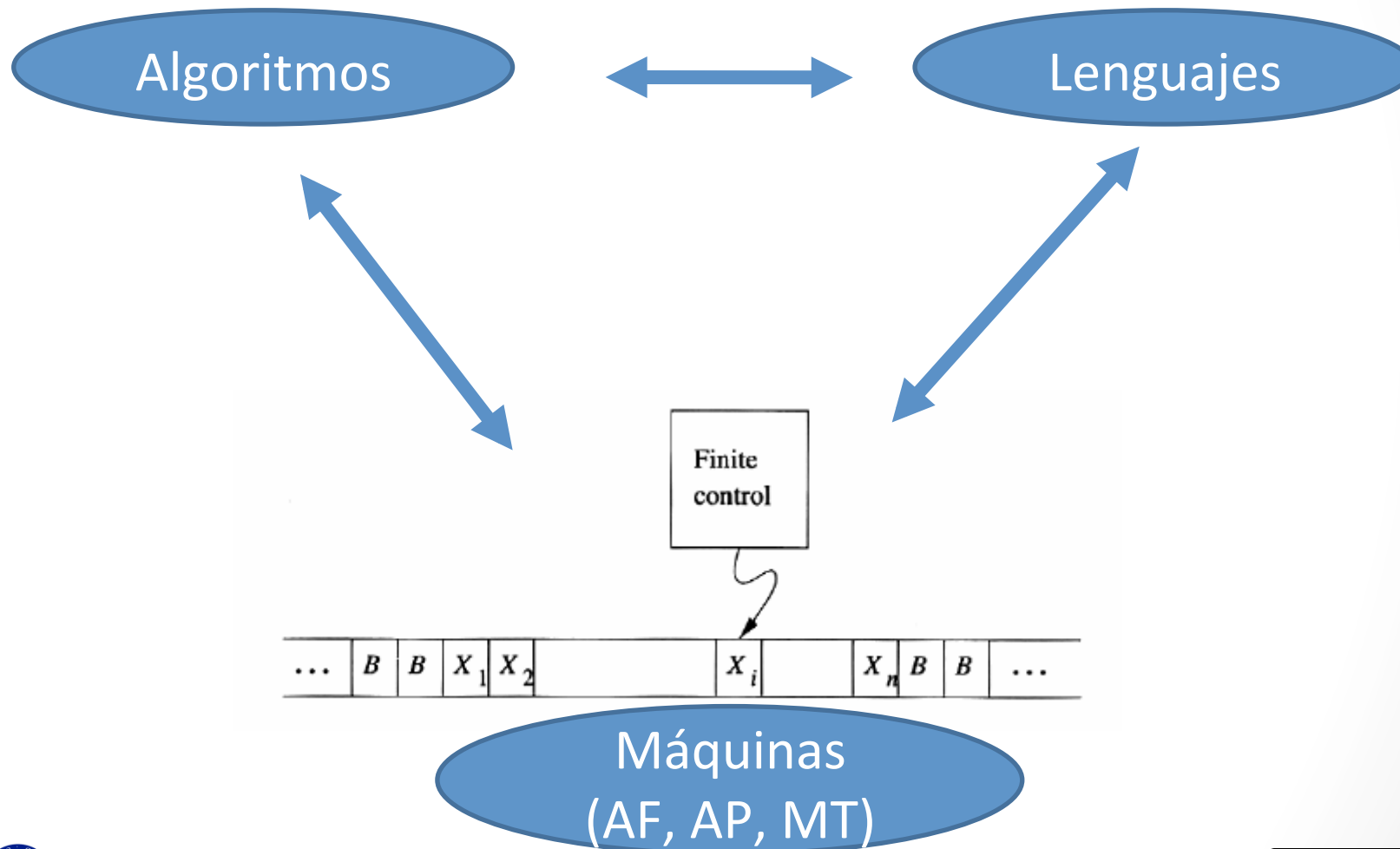


John Backus
(1924 - 2007)

Máquinas, Lenguajes y Algoritmos



Máquinas, Lenguajes y Algoritmos



Bibliografía

- **Referencias básicas :**

1. J. E. Hopcroft, R. Motwani, J. D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Ed. Pearson Addison Wesley , 2008
Capítulo 1. Introducción a lo Autómatas
2. E. Alfonseca Cubero, M. Alfonseca Moreno, R. Moriyón Salomón. *Teoría de Autómatas y Lenguajes Formales*. Ed. McGraw-Hill, 2007
Capítulo 1. Máquinas, Lenguajes y Problemas.

- **Referencias complementarias:**

1. P. Isasi, P. Martínez, D. Borrajo. *Lenguajes, Gramáticas y Autómatas: Un enfoque práctico*. Ed. Addison-Wesley, 1997
Capítulo 2. Lenguajes y Gramáticas Formales
2. D. M Kelley. *Teoría de autómatas y lenguajes formales*. Prentice-Hall, 1995
Capítulo 2. Lenguajes Regulares.
3. R. Penrose. *La Nueva Mente del Emperador*. DeBolsillo, 2011
Capítulo 1. ¿Puede tener mente un computador?
Capítulo 2. Algoritmos y máquinas de Turing
4. R. Penrose. *Las sombras de la mente: hacia una comprensión científica de la consciencia*. Mondadori. 1996
5. D.R. Hofstadter. *Gödel, Escher, Bach : un eterno y grácil bucle*. Tusquets, 1998

1. Introducción a la Teoría de Autómatas y Lenguajes Formales

Araceli Sanchis de Miguel
Agapito Ledezma Espino
José A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber

Grado Ingeniería Informática
Teoría de Autómatas y Lenguajes Formales

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M. Alonso

20



Universidad
Carlos III de Madrid
www.uc3m.es

