



## Computabilidad y Complejidad

### **Tema 1: Una introducción a la Teoría de la Computabilidad**

La **teoría de la computación** es una rama de la matemáticas y la computación que centra su interés en las limitaciones y capacidades fundamentales de las computadoras. Específicamente esta teoría busca modelos matemáticos que formalizan el concepto de *hacer un cómputo* (cuenta o cálculo) y la clasificación de problemas. Esta teoría explora los límites de la posibilidad de solucionar problemas mediante **algoritmos**. Gran parte de las ciencias computacionales están dedicadas a resolver problemas de forma algorítmica, de manera que el descubrimiento de **problemas irresolubles** es útil para no tratar de resolver algorítmicamente estos problemas, ahorrando así tiempo y esfuerzo.

Obviando los trabajos de algunos matemáticos ilustres (Leibniz, Pascal, etc.), podemos establecer que los fundamentos de la teoría de la computación se realizaron durante la primera mitad del siglo XX. Matemáticos como A. Church, A. Turing , K. Gödel, S. Kleene y E. Post (entre otros) crearon los fundamentos de la Teoría de la Computabilidad que, hoy en día, es una parte fundamental de la Informática Teórica (*Theoretical Computer Science*).

Además, esta teoría engloba (o puede describir) otras subramas de conocimiento fundamentales en la informática teórica: **la teoría de autómatas** (o máquinas formales), **la teoría de los lenguajes formales** y **la teoría de la complejidad computacional**.



David Hilbert

El **Programa de Hilbert**, formulado por el matemático alemán David Hilbert en la década de 1920, fue una solución propuesta ante la crisis fundacional de las matemáticas, en épocas en que en los primeros intentos por clarificar los fundamentos de la matemática contenían paradojas e inconsistencias. Como solución, Hilbert propuso basarse en todas las teorías existentes para formar un conjunto de axiomas finito y completo, y proveer prueba de que esos axiomas eran consistentes. El alemán propuso que la consistencia de sistemas más complicados, como el análisis real, podrían ser probados en términos de sistemas más simples. Últimamente, la consistencia de toda la matemática puede ser reducida a aritmética básica.

Los **teoremas de incompletitud de Gödel** son dos célebres teoremas de lógica matemática demostrados por Kurt Gödel en 1931. Ambos están relacionados con la existencia de **proposiciones indecidibles** en ciertas teorías aritméticas.

El **primer teorema de incompletitud** afirma que, bajo ciertas condiciones, ninguna teoría matemática formal capaz de describir los números naturales y la aritmética con suficiente expresividad, es a la vez consistente y completa. Es decir, si los axiomas de dicha teoría no se contradicen entre sí, entonces existen enunciados que no pueden probarse ni refutarse (usando sólo las reglas de deducción de dicha teoría). Las teorías aritméticas para las que el teorema es válido son básicamente aquellas en las que la deducción de teoremas puede realizarse mediante un algoritmo (y por tanto el conjunto de axiomas sea recursivamente enumerable).



Kurt Gödel

El **Entscheidungsproblem** (en castellano: problema de decisión) fue el reto en lógica simbólica de encontrar un algoritmo general que decidiera si una fórmula del cálculo de primer orden es un teorema. En 1928, David Hilbert y Wilhelm Ackermann propusieron la pregunta en su formulación anteriormente mencionada.



Alonzo Church

Antes de poder responder a esta pregunta, hubo que definir formalmente la noción de algoritmo. Esto fue realizado por Alonzo Church en 1936 con el concepto de "calculabilidad efectiva" basada en su **cálculo lambda** y por Alan Turing basándose en la **máquina de Turing**. La respuesta negativa al Entscheidungsproblem fue dada por Alonzo Church en 1936 e independientemente, muy poco tiempo después por Alan Turing, también en 1936. Church demostró que no existe algoritmo (definido según las funciones recursivas) que decida para dos expresiones del cálculo lambda si son equivalentes o no. Por otra parte, Turing redujo este problema al problema de la parada para las máquinas de Turing. Ambos trabajos se vieron influidos por trabajos anteriores de Kurt Gödel sobre el teorema de incompletitud, especialmente por el método de asignar números a las fórmulas lógicas para poder reducir la lógica a la aritmética.



Alan Turing

# Teoría de la computabilidad

## Teoría de la complejidad

Clases de complejidad.  
Clasificación de problemas.  
Relaciones entre problemas y recursos computacionales, etc.

## Teoría de autómatas (máquinas abstractas)

- Máquinas de Turing
- Autómatas de memoria limitada linealmente
- Autómatas de pila
- Autómatas finitos

Además, otros tipos de máquinas abstractas:  
máquinas de registros  
máquinas de contadores  
autómatas celulares  
autómatas de colas, etc.

## Teoría de lenguajes formales (gramáticas formales)

### La Jerarquía de Chomsky

- Gramáticas de tipo 0 (lenguajes recursivamente enumerables)
- Gramáticas de tipo 1 (lenguajes sensibles al contexto)
- Gramáticas de tipo 2 (lenguajes de contexto libre)
- Gramáticas de tipo 3 (lenguajes regulares)

### Además, otras jerarquías de lenguajes formales:

Gramáticas contextuales ó de Marcus.  
Gramáticas con derivación controlada  
Gramáticas matriciales, etc.

Otra perspectiva radicalmente distinta es la que se contempla en lo que hoy en día se conoce como **computación natural**. La computación natural forma parte de la **computación no convencional** y toma como fuente de inspiración los fenómenos naturales (atómicos, bioquímicos, evolutivos, etc.) y el tratamiento de la información que se realiza en los mismos.

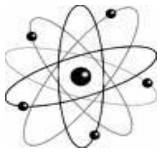
La computación natural re-define el concepto de algoritmo formulando nuevos modelos de computación equivalentes a la máquina de Turing.

Estos modelos pueden ser abstractos y la plausibilidad de su implementación real es materia de investigación en la actualidad. Fundamentalmente, se están investigando nuevos materiales (nano-bio-mecánicos) que cambien el paradigma actual de las computadoras electrónicas basadas en la tecnología del silicio hacia computadoras bioquímicas que tomen como sustrato las bio-macromoléculas (el ADN, el ARN, las proteínas, etc.)

Nacen nuevas disciplinas que integran la biología, la física y la computación: la biología sintética y la nanocomputación son dos ejemplos reales de estas nuevas disciplinas.

Además, los fenómenos naturales se estudian bajo otro prisma que integra la teoría de la información y la teoría de la computación. Áreas de conocimiento como la biología de sistemas y la cosmología integran hoy en día este nuevo paradigma.

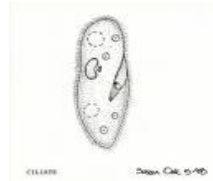
## Algunos modelos de la computación natural



**computación cuántica**



**computación biomolecular  
basada en ADN, ARN y proteínas**



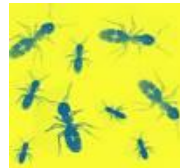
**computación en microorganismos:  
computación en ciliados**



**sistemas inmunoartificiales**



**computación celular: Sistemas P,  
Procesadores evolutivos, etc.**



**sistemas cooperativos naturales  
(inteligencia en enjambre)**



**computación multicelular : Sistemas  
de Tejidos P, Redes de Procesadores  
Evolutivos, Redes neuronales, etc.**



**sistemas cooperativos sociales  
y ecológicos**



**sistemas cromosómicos:  
Algoritmos genéticos, computación  
evolutiva**

## Conclusiones

- La teoría de la computabilidad propone la definición de algoritmo y establece los límites de los problemas que se pueden resolver de forma algorítmica.
- Propone nuevos modelos de cálculo que pueden tener una implicación real en las tecnologías informáticas futuras.
- Explica de forma exhaustiva las capacidades de la tecnología informática actual y da cuenta de sus limitaciones. La teoría de la computabilidad engloba a la Ingeniería Informática como un caso particular de la tecnología actual.
- Propone un marco matemático que explica otras disciplinas y áreas de estudio bajo un nuevo prisma formal.