Tema I: Introducción. Computación Natural

Planteamiento y resolución de de problemas.

Búsqueda de procedimientos sistemáticos.

Resolución *mecánica* de problemas:

- Transferencia de conocimiento.
- Apoyo a la resolución.

Algoritmo: cualquier método especial de resolución de un cierto tipo de problemas.

Abú Jáfar Mohammed ibn al–Khowarizmi: procedimientos mecánicos para el Álgebra (año 825 d.C.)

Primer algoritmo no trivial: máximo común divisor de dos números enteros (Euclides entre 400 y 300 a. C.).

Computabilidad

Existencia de problemas resolubles *mecánicamente*

A finales del siglo XVII Leibnitz formula

- Necesidad de disponer de un lenguaje universal (*lingua characteristica*)
- Necesidad de mecanizar cualquier tipo de razonamiento (*calculus ratiocinator*).

D. Hilbert (en 1928) formula tres cuestiones sobre las Matemáticas:

1. *¿Son completas?*
2. *¿Son consistentes?*
3. *¿Son decidibles?* (*Entscheidungsproblem*).

Respuestas negativas a las dos primeras cuestiones: teoremas de incompletitud de K. Gödel (1931).

- No es posible encontrar una axiomatización completa de las Matemáticas.

Gödel dejó sin responder la tercera cuestión.

Aparecen problemas de los que no se conocen soluciones *mecánicas*.

Cuestión 1: *Dado un problema, determinar si existe un procedimiento mecánico que lo resuelve*.

- Soluciones positivas *versus* soluciones negativas.

Modelo de computación: *sintaxis* y *semántica*.

Clase de *funciones computables* en el modelo.

Formalización del concepto de procedimiento mecánico

- *Funciones recursivas* (K. Gödel, 1931–1933).
- λ*–cálculo* (A. Church y S. Kleene, 1931).
- *Máquinas de Turing* (A. Turing, 1936).

Limitaciones de los modelos de computación

- Existencia de problemas indecidibles.
  - Lógica de primer orden (Church y Turing, 1936).
  - Problema de la parada (Turing, 1936).

La tesis de *Church–Turing*.

Equivalencia modelos de computación (Turing, 1936).

Complejidad Computacional

Aparición de los primeros ordenadores de propósito general (implementación práctica de ideas de J. von Neumann).

¿Qué problemas resuelven las *máquinas reales*?

Resolubilidad *mecánica práctica* de problemas.

- Cantidad de *recursos computacionales* necesarios (tiempo y espacio).
- Análisis comparativo de distintas soluciones.

Cuestión 2: *Dado un problema hallar el mejor algoritmo que lo resuelva.*

Búsqueda de Algoritmos óptimos

- Hallar una cota inferior de los recursos necesarios para ejecutar *cualquier* algoritmo que resuelva el problema.
- Hallar un algoritmo que resuelva el problema y que use una cantidad de recursos del orden de la cota.

Complejidad computacional inherente a un problema.

A veces, es *imposible* encontrar algoritmos óptimos

- (*teorema de aceleración de Blum*).

Clases de complejidad

Necesidad de analizar la complejidad de problemas de manera global:

- *Clases de complejidad*.

Ingredientes necesarios para definir una clase de complejidad:

- Un *modelo de computación*.
- Un *modo de computación*.
- Una *medida de complejidad*.
- Una *función* total computable (*cota superior* de recursos).

Las clases de complejidad **L**, **P** y **EXP**.

Resolubilidad a través de ordenadores reales.

- Tratabilidad e intratabilidad de problemas.

Algoritmo eficiente: la cantidad de recursos necesarios para su ejecución está acotada por un polinomio.

¿Por qué los polinomios para establecer la frontera?

- Es una clase de funciones estables por suma y producto.
- Tienen un crecimiento *moderado*.

La clase **P** de los problemas tratables.

Computación no determinista

Una configuración de la máquina puede tener varias configuraciones siguientes.

- Puede realizar computaciones distintas con un mismo dato de entrada.

La clase **NP**: *tratabilidad en modo no determinista*.

Se tiene que **P**$\subseteq$**NP**.

¿Es estricta la inclusión **P**$\subseteq$**NP**?

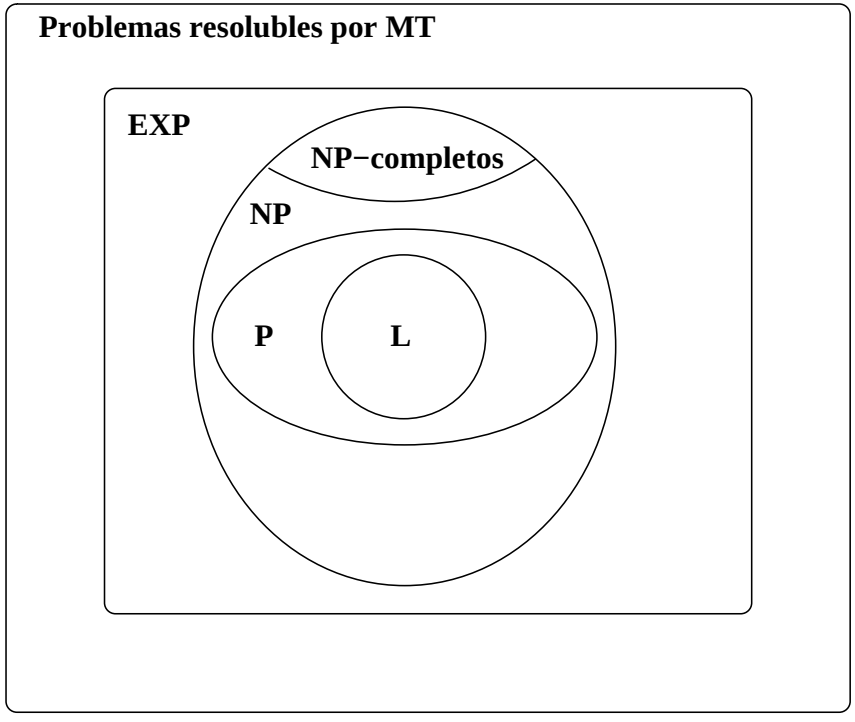
El problema **P** *versus* **NP**: determinar si **P** y **NP** coinciden.

- Premio del CMI: un millón de dólares.

Problemas NP–completos

Son los problemas más *difíciles* de la clase **NP**.

Problemas **NP**–completos: candidatos idóneos para atacar la cuestión $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$.



Necesidad de mejorar *cuantitativamente* la resolución mecánica de problemas **NP**–completos.

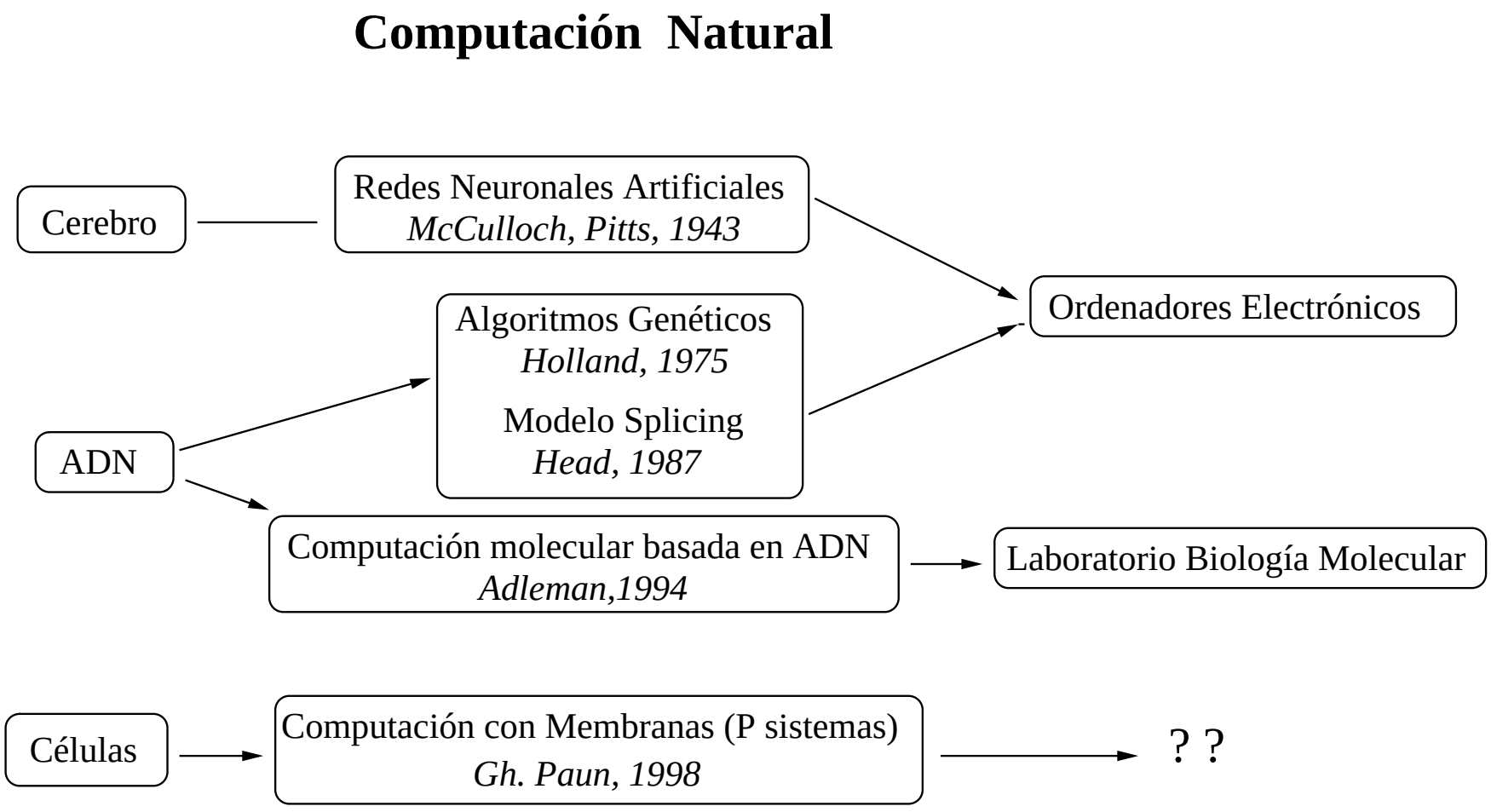
El problema **SAT** (S.A. Cook, 1971).

¿Cómo enfrentarnos a un problema computacionalmente difícil/duro?

- Preguntarnos en qué aspecto del problema radica la razón de la dificultad.
- Intentar buscar una solución aproximada más simple.
- Tener presente que algunos problemas sólo son difíciles en el caso peor.
- Considerar otros modelos alternativos, no convencionales.

La Naturaleza: Una alternativa

Computación Natural: disciplina que intenta capturar la forma en que la Naturaleza lleva *calculando* millones de años.



Computación molecular basada en ADN

- *Tratabilidad* de problemas:
  - Paralelismo.
  - Miniaturización.
- *Computación a nivel molecular* (R. Feynman, 1961).
- Limitaciones velocidad procesadores (R. Churchhouse, 1983).
- Analogía: procedimientos matemáticos y procesos biológicos.
  - L. Adleman materializó esta similitud (nov. 1994).

- Cromosomas: proteinas + **ADN**.
- **ADN** (J. Watson y F. Crick, 1951–1953)
  - Descifran la estructura.
  - Descubren el principio de complementariedad.
  - Demuestran que las moléculas de ADN codifican toda la información genética.
  - Justifican el uso de ciertas técnicas para su manipulación.
- Transistor (1958): manipulación electrónica silicio.
- L. Adleman (1994): manipulación bioquímica del carbono.
- Julio de 2000: interruptor a partir de una molécula.
  - Sustituye la luz por una reacción química.
  - Pueden disponer de más de mil procesadores en el espacio ocupado hoy día por un procesador.
  - Pueden aumentar la velocidad cien mil millones de veces.
  - Pueden reproducir cien ordenadores convencionales en el tamaño de un grano de sal fina.
- Simulación bioquímica de una MT (E. Shapiro, nov. 2001)

Computación Celular con Membranas

Modo en que la Naturaleza *calcula* a un *nivel celular*.

Célula: unidad fundamental de todo organismo vivo.

- Estructura compleja y, a la vez, muy organizada.
- Permite ejecución simultánea de reacciones químicas.

Existen dos tipos de células:

- *Procariotas*: carecen de un núcleo bien definido (propias de los organismos unicelulares).
- *Eucariotas*: poseen un núcleo rodeado por una doble membrana (específicas de animales y plantas).

Computación celular con Membranas:

- Introducida por Gh. Păun (octubre de 1998- febrero 2000).
- Modelo no determinista de tipo distribuido, paralelo y maximal.
- Inspirado en el funcionamiento de la célula como organismo vivo capaz de procesar y generar información.