

# TEMA I

## Introducción

Mario de J. Pérez Jiménez

Grupo de investigación en Computación Natural

Dpto. Ciencias de la Computación e Inteligencia Artificial

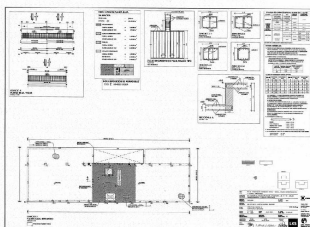
Universidad de Sevilla

Simulación y análisis computacional en Biología de Sistemas  
Máster Universitario en Lógica, Computación e Inteligencia Artificial

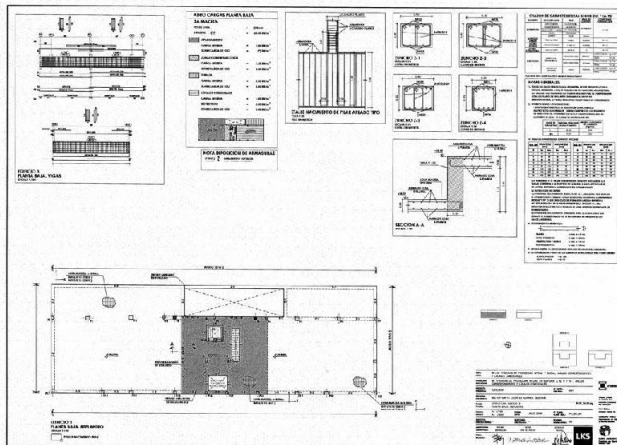
Curso 2012-13



# Computación y Máquinas



# Modelo de computación



## Máquina del modelo



# Modelos de Computación

Consta básicamente de:

- ▶ Una sintaxis formal que especifica los **procesos** que son considerados **mecánicos** en el modelo.
- ▶ Una semántica formal que especifica cómo se ejecutan los **procesos mecánicos** del modelo.

Todo modelo de computación tiene asociado:

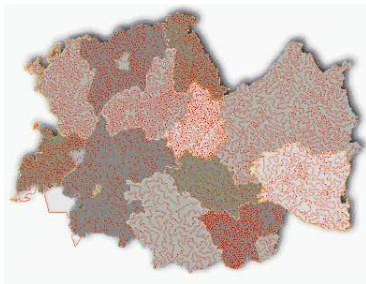
- ▶ El conjunto de todos los problemas que son resolubles por procesos mecánicos del modelo.

Para algunos modelos de computación es posible diseñar dispositivos físicos (**máquinas**) que son capaces de ejecutar de manera autónoma los procesos mecánicos del modelo.

# Resolución mecánica de problemas (I)

*Un problema concreto de la vida real:*

- ▶ Visitar 3150 ciudades, coste  $c_{ij}$  entre dos ciudades.
- ▶ Hallar un circuito por las 3150 ciudades de coste mínimo.



Un problema abstracto: TSP

*Dado un grafo no dirigido con pesos en sus aristas, determinar un circuito hamiltoniano de peso mínimo*

# Resolución mecánica de problemas (II)

## *Otro problema concreto de la vida real:*

- ▶ Determinas los mecanismos moleculares que rigen el comportamiento de una colonia concreta de bacterias en simbiosis.

Un problema abstracto:

- ▶ Describir un **marco formal** que **simule** el comportamiento de una colonia de bacterias del tipo considerado.

# Resolución mecánica de problemas (III)

- ▶ Se *modeliza* a través de un *problema abstracto*.
- ▶ Se diseña una *solución mecánica* del problema abstracto.
- ▶ Se *implementa* dicha solución mediante un programa.
- ▶ Se *ejecuta* el programa sobre una máquina electrónica para los datos específicos del *problema concreto*.

# Problemas tratables

Complejidad computacional de problemas abstractos:

- ▶ Tamaño de un problema concreto.
- ▶ Fijado un recurso computacional, toda solución mecánica de un problema abstracto tiene asociada una función (tamaño = variable independiente).

Resolubilidad algorítmica en términos prácticos: *tratabilidad*.

**Polinomial = Bueno**

**Exponencial = Malo**

	10	20	30	40	50
$n$	.00001 s	.00002 s	.00003 s	.00004 s	.00005 s
$n^2$	.00001 s	.00004 s	.00009 s	.00016 s	.00025 s
$n^3$	.00001 s	.00008 s	.027 s	.064 s	.125 s
$n^5$	.1 s	3.2 s	24.3 s	1.7 m	5.2 m
$2^n$	.001 s	1.0 s	17.9 m	12.7 d	35.7 a
$3^n$	.059 s	58 m	6.5 a	3855 sig	$2 \cdot 10^8$ sig

Problemas tratables: los *más simples*.



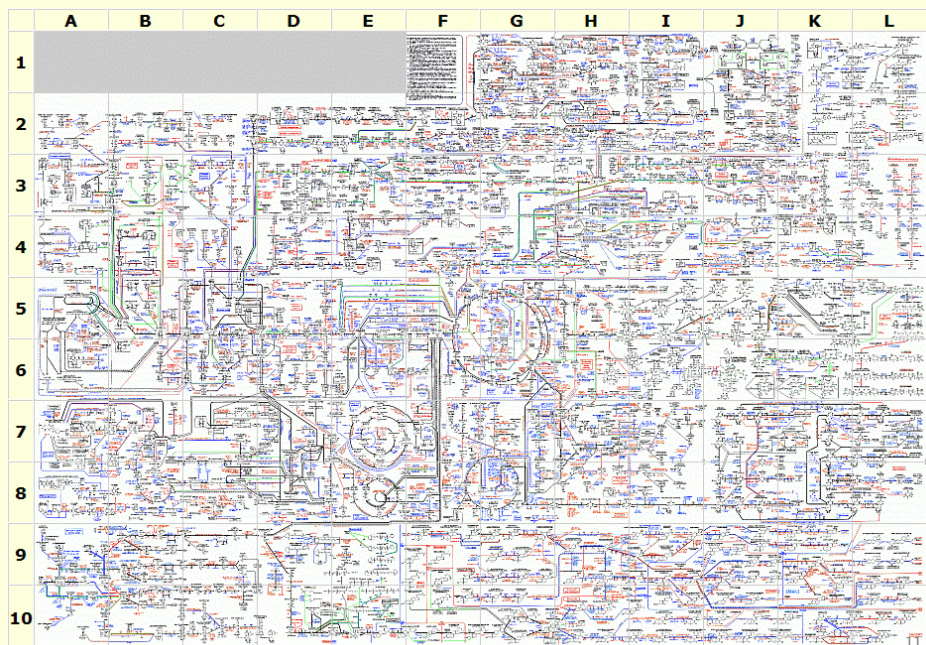
Desgraciadamente, existen muchos problemas abstractos tales que:

- ▶ **NINGÚN** programa **CONOCIDO** que lo resuelve proporciona soluciones para entradas de *tamaño grande*.

Problemas **presuntamente intratables** (**NP**-completos).

- ▶ La mayoría de sus instancias no podrían ser resueltas.

# Un problema presuntamente intratable



# La conjetura $P \neq NP$

¿ $P \neq NP$ ?

Si  $P = NP$ :

- ▶ Consecuencias funestas para la criptografía.
- ▶ Obtención de **pruebas de teoremas** de *longitud razonable*

Si  $P \neq NP$ :

- ▶ Reforzamiento de la **seguridad** de la **criptografía moderna**.
- ▶ Existencia de problemas de complejidad intermedia.

# Limitaciones de las máquinas electrónicas

Máquinas: dispositivos finitos

Recursos: *espacio* (memoria) y en *tiempo*.

- ▶ Espacio: miniaturización (*R. Feymann, 1959*).
- ▶ Tiempo: velocidad de cálculo de procesadores (*R. Churchhouse, 1983*).

*Consecuencia:*

- ▶ Existen problemas de la vida real que **nunca** podrán ser resueltos por ordenadores electrónicos (**a menos que ...**)

# Computación Natural (I)

- ▶ Estudiar modelos y técnicas computacionales inspiradas en la Naturaleza.
- ▶ Comprender el mundo en términos de **procesamiento de la información**.
  - ▶ *La información es más importante que la materia y que la energía* (Zuse, Fredkin 1960s).
  - ▶ *El universo: dispositivo computacional que está actualizando constantemente sus reglas de cálculo* (tesis de Zuse-Fredkin).
- ▶ La Naturaleza como fuente de inspiración.
- ▶ La Naturaleza como sustrato de implementación.
- ▶ La Naturaleza como computación.

# Computación Natural (II)

## La Naturaleza como fuente de inspiración:

- ★ *Cellular Automata* (K. Zuse, S. Ulam, J. von Neumann , 1940s).
- ★ *Neural networks* (W. McCulloch y W. Pitts , 1943).
- ★ *Genetics Algorithms* (J. Holland, 1975), *Evolutionary Computing* (T. Baeck, D. Fogel, Z. Michalewicz, 1997), *Genetic Programming*(J. Koza, 1992).
- ★ *Artificial Life* (Lindenmayer, 1968, Ch. Langton, 1986.).
- ★ *Artificial Immune Systems* (J.D. Farmer, N. Packard, A. Perelson, 1986).
- ★ *Swarm intelligence* (G. Beni, J. Wang, 1989).
- ★ *Membrane Computing*(Gh. Păun, 1998–2000).

# Computación Natural (III)

## La Naturaleza como sustrato de implementación:

- ★ *Molecular Computing* (L. Adleman, 1994).

## La Naturaleza como computación:

- ★ *Computational Systems Biology* (H. Kitano, 2002).
- ★ *Sinthetic Biology* (S.A. Benner, A.M. Sismour, 2005).
- ★ *Computation with living cells* (A. Ehrenfeucht, T. Harju, I. Petre, D.M. Prescott, G. Rozenberg, 2003).

# Computación *in vitro*

- ▶ Modelos moleculares basados en ADN.
- ▶ Experimento de Adleman (1994).
- ▶ Programas moleculares.

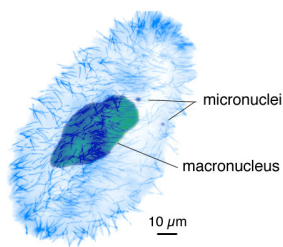
Un problema **presuntamente intratable**: *dado un grafo no dirigido  $G$ , determinar si existe una coloración válida de  $G$  con tres colores (3-COL).*

Programa molecular que lo resuelve:

```
Entrada:  $T$  (contiene todas las posibles coloraciones de  $G$ )
Para  $i \leftarrow 1$  hasta  $p$  hacer
   $T_1 \leftarrow +(T, e_i^1(c_1)); T_1^* \leftarrow -(T, e_i^1(c_1))$ 
   $T_2 \leftarrow +(T_1^*, e_i^1(c_2)); T_3 \leftarrow -(T_1^*, e_i^1(c_2))$ 
  Para  $j \leftarrow 1$  hasta 3 hacer
     $T'_j \leftarrow -(T_j, e_i^2(c_j))$ 
   $T \leftarrow T'_1 \cup T'_2$ 
   $T \leftarrow T \cup T'_3$ 
Detectar( $T$ )
```

# Computación **in vivo** (I)

## Gene assembly in ciliates (2003)



Naturaleza computacional del ensamblaje de genes en ciliados:

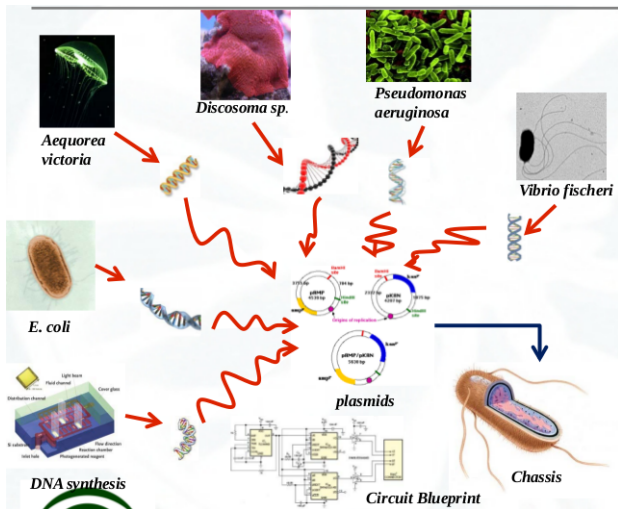
- ▶ Marco formal para razonar sobre el ensamblaje biológico.
- ▶ Desarrollo de aplicaciones software para la simulación.
- ▶ Importantes computaciones tienen lugar en la vida de los ciliados.
- ▶ Uso de estructuras de datos en la computación in vivo (listas enlazadas).

# Computación **in vivo** (II)

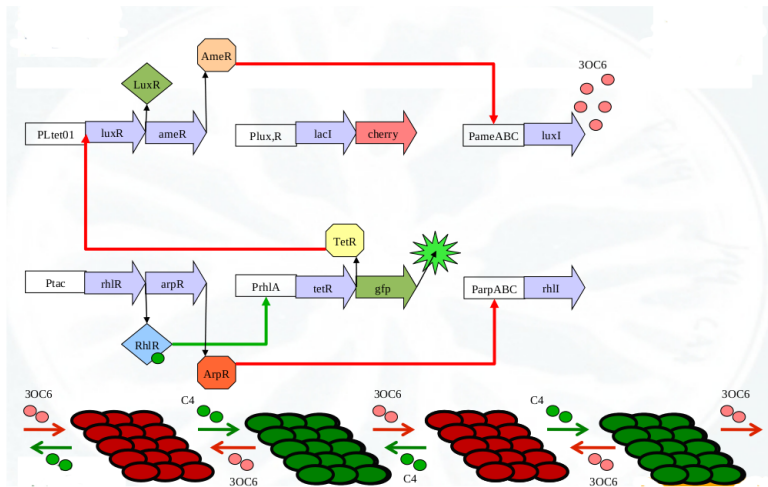
## Biología sintética:

- ▶ Incorporar principios de la ingeniería para el diseño e **implementación in vivo** de circuitos de genes con comportamiento deseado.
- ▶ Se proponen modelos formales para evaluar circuitos de genes sintéticos.
  - ▶ Membrane Computing.

# Computación in vivo (III)



# Computación in vivo (IV)



# 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: proteínas + **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*.

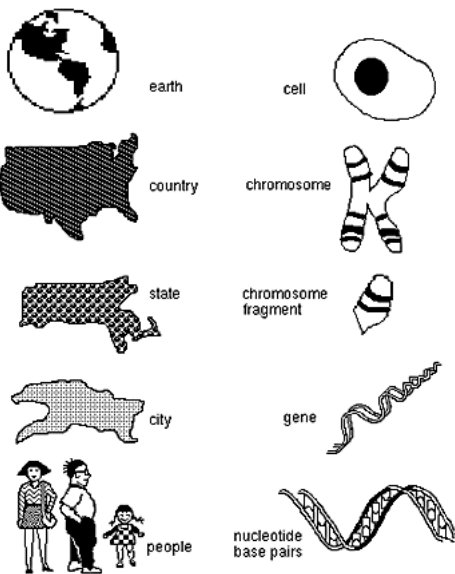
**Célula:** unidad fundamental de todo organismo vivo.

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

Existen dos tipos de células:

- ▶ **Procariontas:** 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).

# Escalas comparativas



**Comparative Scale of Mapping**

# Curiosidades ...

Cada célula contiene una molécula de ADN en el núcleo:

- ▶ ¿Qué dimensiones aproximada tiene? (grosor y longitud)
  - Diámetro: 2 nanómetros.
  - Longitud: 2'3 metros.

En el cuerpo humano existen un total de  $10^{14}$  células. Imaginemos que “desplegamos” todas las moléculas de ADN de nuestro cuerpo

- ▶ ¿Qué distancia aproximada cubriría?
  - Aproximadamente: 600.000 veces la distancia entre la tierra y la luna.

# Modelos de computación celular con membranas

## *Membrane Computing:*

- ▶ 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.
- ▶ El artículo fundacional<sup>1</sup> fue nominado por el ISI como *Fast Breaking Paper* (febrero de 2003).
- ▶ Declarada por el ISI Thomson Reuters como *Fast Emerging Research Front in Computer Science* (noviembre 2003).

Modelos no determinista de tipo distribuido, paralelo y maximal.



---

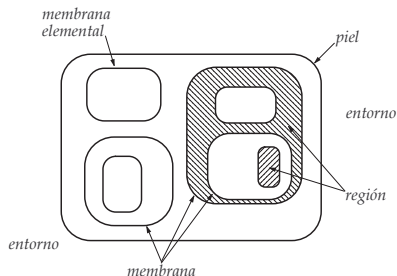
<sup>1</sup>Gh. Păun. Membrane Computing. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for Computer Science-TUCS Report Nr. 208*, 1998.



# Sistemas P que trabajan a modo de células

Ingredientes sintácticos:

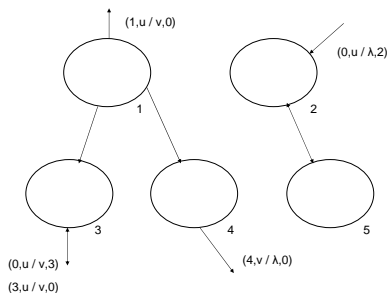
- ▶ Un **alfabeto**, cuyos elementos se denominan **objetos**.
- ▶ Un conjunto de **membranas** estructurado en un árbol enraizado (*regiones*)
- ▶ Unos multiconjuntos de objetos asociados a cada región.
- ▶ Conjunto de **reglas de evolución**.
- ▶ Dos membranas distinguidas: una de **entrada** y otra de **salida**.
- ▶ Un **entorno pasivo**.



# Sistemas P que trabajan a modo de tejidos

Ingredientes sintácticos:

- ▶ Un **alfabeto**, cuyos elementos se denominan **objetos**.
- ▶ Un conjunto de **células** estructurado en un grafo dirigido.
- ▶ Unos multiconjuntos de objetos asociados a cada célula.
- ▶ Conjunto de **reglas de evolución**.
- ▶ Dos células distinguidas: una de **entrada** y otra de **salida**.
- ▶ Un **entorno activo**.



# Sistemas P: semántica

Las reglas del sistema se aplican de forma paralela, maximal y no determinista

Se definen los conceptos de:

- ▶ **Configuración.**
- ▶ **Transición** de una configuración a otra.
- ▶ **Computación** a partir de una configuración inicial.

# Biología de Sistemas (I)

Dinámica de los sistemas biológicos:

- ▶ Aproximación clásica:
  - ▶ Identificación y caracterización de cada componente individual.
- ▶ Nueva aproximación:
  - ▶ Estudio y análisis de las interacciones entre los elementos que integral el sistema.

La Biología de Sistemas es un campo multidisciplinar que integra la Biología clásica, las Matemáticas, la Física y la Informática.

# Biología de Sistemas (II)

Los métodos clásicos de estudio se basan en el *método científico*:

- ▶ Confirmación o refutación de hipótesis a través de la experimentación.

En Biología de Sistemas la herramienta fundamental es la *modelización matemático/computacional*:

- ▶ La complejidad de los procesos biológicos y la gran cantidad de datos obtenidos por la experimentación, hacen necesario el modelización formal, la simulación y el análisis.
- ▶ El modelo trata de describir el comportamiento del sistema objeto de estudio como un sistema dinámico complejo, mediante el estudio de las relaciones y las interacciones entre las diferentes partes del sistema.