

## 2

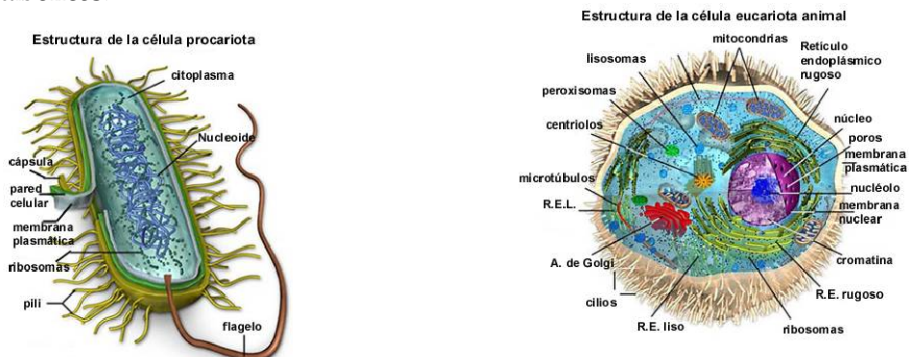
## CONCEPTOS BÁSICOS DE LA COMPUTACIÓN CELULAR CON MEMBRANAS

Esta unidad tiene como objetivo presentar el paradigma de computación celular con membranas, partiendo de la inspiración biológica de la célula y su capacidad para procesar información, y pasando a describir los elementos fundamentales de los sistemas P y algunas de sus variantes principales.

### 1 LA CÉLULA COMO UNIDAD BIOLÓGICA DE PROCESAMIENTO

La célula es la unidad fundamental de todo organismo vivo. Posee una estructura compleja y, a la vez, muy organizada que permite la ejecución simultánea de un gran número de reacciones químicas.

Existen, básicamente, dos tipos de células: las *procariotas*, que carecen de un núcleo bien definido y son propias de los organismos unicelulares (bacterias), y las *eucariotas*, que poseen un núcleo rodeado por una doble membrana y son específicas de los animales y plantas. Ambas realizan de manera similar una serie de procesos complejos que son esenciales para la vida, tales como la replicación del ADN, la producción de energía, la síntesis de proteínas y los procesos metabólicos.



En un primer análisis podemos distinguir tres partes claramente diferenciadas en una célula: (a) una especie de piel (*membrana plásmica*) que delimita a la célula

de su entorno, permitiendo diferenciar el interior del exterior de la célula; (b) el corazón de la célula (*núcleo*), que almacena la información genética a través de moléculas de ADN y de ARN; y (c) el resto de la célula (*citoplasma*), que contiene distintas componentes de la célula.

Dentro del citoplasma podríamos destacar las siguientes componentes:

1. *La mitocondria*, encargada de la producción de energía para la célula.
2. El *aparato de Golgi*, que viene a ser una fábrica de proteínas y juega un papel esencial en el metabolismo de la célula.
3. El *retículo endoplásmico*, que es una red de membranas interconectadas estructurada en dos partes: el *retículo endoplásmico rugoso*, que contiene *ribosomas* adheridos a sus paredes lo cual facilita el paso del ARN mensajero del núcleo a fin de producir la síntesis proteínica, y el *retículo endoplásmico liso*, que se encarga del transporte de moléculas y la comunicación entre las componentes de la célula.
4. Los *lisosomas*, que constan de una única membrana y se encargan de digerir sustancias que proceden del exterior (*vacuolas digestivas*) y de ingerir restos celulares que han llegado a ser obsoletos para el organismo (*vacuolas autofágicas*). Son como los *estómagos* de las células y reciben el nombre de *bolsas suicidas* debido a que la rotura de su membrana provocaría la destrucción de la célula por la acción de las enzimas encerradas en su interior.

En lo que respecta a la disposición interna de la célula, la primera característica que llama la atención es el hecho de que las distintas partes del sistema biológico que la componen se encuentran delimitadas por varios tipos de *membranas*, desde el compartimento que separa el exterior de la célula del interior de la misma, hasta las distintas membranas que delimitan las vesículas interiores. El premio Nobel de química del año 2003, recayó en los científicos P. Agre y R. MacKinnon por sus descubrimientos relacionados con los canales de ciertas membranas celulares, que han supuesto un gran avance en los estudios

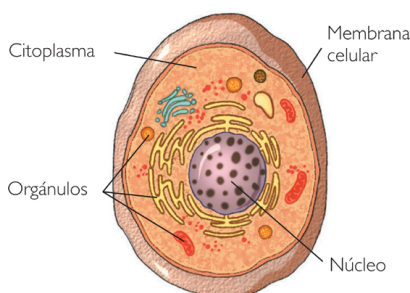
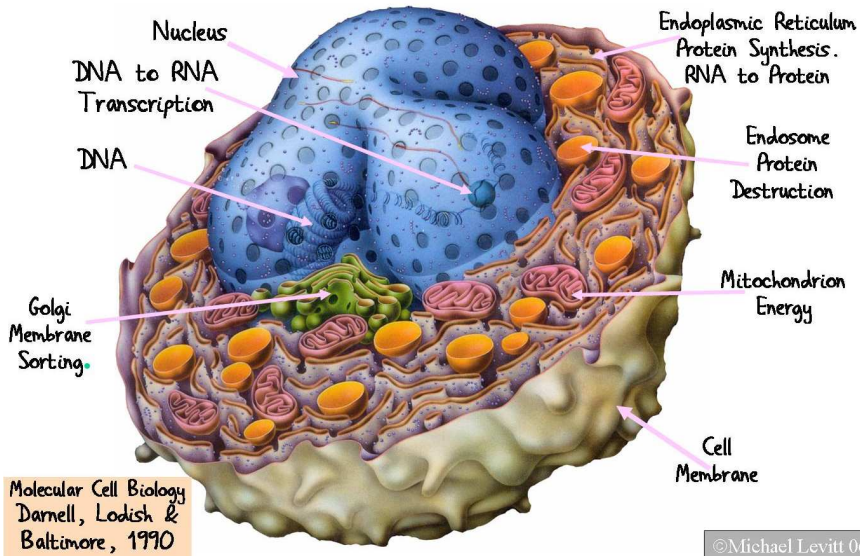


FIGURA 1: La célula.

de ciertas membranas celulares, que han supuesto un gran avance en los estudios

de la química celular e ilustran su importancia en los procesos vitales.



Las células eucariotas de nuestro organismo, contienen diferentes partes delimitadas por las *membranas biológicas*. Su estructura y funcionamiento resultan muy interesantes y nos sirven de inspiración. Juegan un papel crucial en las células vivas ya que se estima que el 90% de las actividades celulares se realizan a través de ellas, y, por tanto, es un concepto esencial a la hora de definir el fenómeno que usualmente denominamos como *vida*.

Las membranas biológicas son unas finas capas, de unos 5 nm de espesor, que están compuestas de complejos moleculares de lípidos (fosfolípidos y colesterol), proteínas y carbohidratos que delimitan territorios celulares y relacionan unos compartimentos con otros y a la célula con su entorno. Estas moléculas interactúan según el modelo de mosaico fluido de Singer y Nicholson (1973) donde los fosfolípidos se disponen en bicapas y son los responsables de la función de barrera de las membranas biológicas, mientras que la permeabilidad selectiva y el resto de las funciones que realizan las diversas membranas dependen de las diferentes proteínas que interactúan con esta bicapa.

## 1.1 Células versus máquinas

En una célula viva:

- Cada *membrana* trabaja con *compuestos* químicos de acuerdo con unas *reacciones* específicas

En una máquina paralela:

- Cada *procesador* trabaja con *datos* de acuerdo con un *programa* específico

Célula	Máquina
Membranas	Procesadores
Compuestos químicos	Datos
Reacciones químicas	Instrucciones

Es el momento de formularnos algunas preguntas:

- ¿Es la *célula* una fuente de inspiración para la informática?
- ¿Es posible definir un modelo de computación inspirado en las *células*?
- ¿Es posible implementar computaciones a través de las *células* vivas?

El comportamiento de una célula puede ser considerado como el de una *máquina que realiza ciertos procesos de cálculo*: un dispositivo complejo, desde el punto de vista biológico, en el que por medio de una distribución jerárquica de membranas interiores se produce el flujo y alteración de las sustancias que procesa a través de reacciones químicas.

## 2 SISTEMAS P. ESTRUCTURA Y FUNCIONAMIENTO

A finales de 1998, Gh. Păun introdujo un nuevo paradigma de computación natural, denominado *computación celular con membranas* (*Membrane Computing*), inspirado en la estructura y el funcionamiento de las células. Éstas constituyen la unidad básica de todo organismo vivo, poseen una estructura compleja y, a la vez, muy organizada que permite la ejecución simultánea (*paralela*) de reacciones químicas.

Los ingredientes básicos de un dispositivo computacional de este modelo son: la *estructura de membranas* que representa las vesículas que componen la célula, incluida en una *membrana piel* que las separa del entorno exterior, junto con ciertos *objetos* situados en ellas, que son abstracciones de las sustancias químicas. Estos objetos pueden evolucionar de acuerdo con una serie de *reglas* que son abstracciones de las reacciones químicas

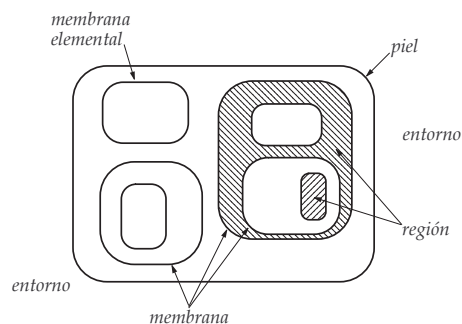


FIGURA 2: Estructura de membranas.

que se producen en el interior de las membranas. Las reglas son aplicadas de forma *no determinista* (si a un objeto se le pueden aplicar varias reglas, se elegirá una de ellas), *paralela* y *maximal* (tras la ejecución de las reglas, no pueden quedar objetos que hayan podido reaccionar). Las membranas biológicas no generan compartimentos estancos sino que permiten, de forma selectiva, el paso (flujo) de ciertos compuestos químicos. Esta propiedad relevante es capturada en el modelo de computación celular que, además, implementa un paralelismo masivo en dos niveles básicos: en el primero, cada membrana aplica sus reglas de forma paralela sobre los objetos presentes en ella; en un segundo nivel, todas las membranas realizan esta operación en paralelo, trabajando simultáneamente, sin interferencia de las operaciones que se estén produciendo en las demás membranas del sistema (para más detalles, ver [8]).

Las bacterias son organismos unicelulares que, cuando conviven en un organismo anfitrión, son capaces de comunicarse entre sí y detectar la existencia de un cierto *quorum*. En ese caso activan unos genes para realizar conjuntamente una determinada acción. Los mecanismos moleculares que rigen la *comunicación inteligente* de bacterias, han sido modelizados computacionalmente, concretamente para el caso de las bacterias *Vibrio Fischeri* que conviven con una especie de calamar. La acción que realizan estas bacterias consiste en la emisión de luz, de la que se aprovecha el calamar para atraer a sus presas y, a la vez, camuflarse para no ser detectado por los depredadores.

Los ingredientes básicos de un *sistema P* (que es como también se conocen los sistemas que se utilizan en computación celular con membranas) son la *estructura de membranas*, que consiste en un conjunto de membranas (al modo de las vesículas que componen las células) incluidas en una *piel* exterior que las separa del entorno, junto con ciertos *multiconjuntos de objetos* (es decir, conjuntos en los que los elementos pueden aparecer repetidos) situados en las *regiones* que delimitan dichas membranas (al modo de los compuestos que hay en el interior de dichas vesículas). Estos objetos pueden transformarse de acuerdo con unas *reglas de evolución* que son aplicadas de una forma no determinista, paralela y maximal (al modo de las reacciones que se pueden producir entre dichos compuestos). Para simular la permeabilidad de las membranas celulares, las reglas de evolución no sólo pueden modificar los objetos presentes en una membrana, sino que pueden pasar a través de dos regiones adyacentes *atravesando* la membrana que las separa.

Este modelo de computación implementa un paralelismo masivo en dos niveles básicos: en un primer nivel, cada membrana aplica sus reglas de forma paralela sobre los objetos presentes en ella, produciendo los nuevos objetos y comunicándolos a las membranas adyacentes si procediera; en un segundo nivel, todas las membranas realizan esta operación en paralelo, trabajando simultáneamente, sin interferencia alguna de las operaciones que se estén produciendo en las demás membranas del sistema.

Los siguientes apartados estudiarán en detalle distintas variantes de sistemas *P* inspiradas en el funcionamiento de las células, los tejidos y las neuronas.

### 3 SISTEMAS P QUE TRABAJAN A MODO DE CÉLULAS

Teniendo presente la descripción de los elementos básicos de los sistemas P introducidos en la sección anterior, podemos distinguir un buen número de variantes de sistemas P que enfatizan determinados elementos que podemos encontrar en la naturaleza, y que podemos agrupar en tres tipos de sistemas principales:

- Sistemas P que trabajan a modo de células.
- Sistemas P que trabajan a modo de tejidos.
- Sistemas P que trabajan a modo de neuronas.

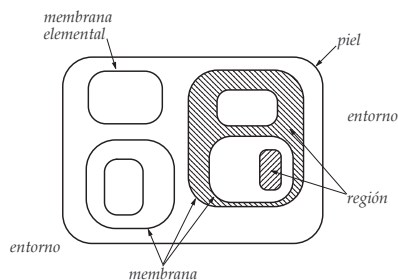
Desde el punto de vista de su inspiración, la diferencia estriba en los elementos principales en los que centramos la atención: células, tejidos o neuronas. Desde el punto de vista computacional, por su parte, tenemos distintos ingredientes fruto de esa inspiración, implicando distintas estructuras internas (árboles o grafos) y distintos tipos de elementos (objetos diversos o un único tipo de objetos simbolizando impulsos). Comenzaremos esta sección introduciendo los elementos principales de los sistemas P básicos funcionando a modo de células. No es objetivo de este capítulo detallar todas las variantes de este tipo de sistemas. En las siguientes secciones se introducirán los restantes sistemas.

De forma breve podemos establecer para el modelo básico inicial de la computación celular con membranas que se trata de un:

- Modelo de computación no determinista de tipo distribuido, paralelo y maximal.

En estos sistemas P, la **estructura de membranas** define:

- Conjunto de membranas
- Multiconjuntos de objetos situados en las *regiones*.
- Reglas de evolución.



Los ingredientes básicos del sistema son:

- Un **alfabeto**, cuyos elementos se denominan **objetos**
- Una **estructura de membranas** (*regiones*)
- Un multiconjunto asociado a cada región
- Conjunto de **reglas de evolución**
- Dos membranas distinguidas: una de **entrada** y otra de **salida**

Se definen los conceptos de:

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

### Sintaxis de los Sistemas P

*Sistema P básico* de grado  $q$ , con  $q \geq 1$ :

$$\Pi = (\Gamma, L, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, (R_1, \rho_1), \dots, (R_q, \rho_q), i_0)$$

en donde:

- $\Gamma$  es un alfabeto (**objetos**).
- $L$  es un conjunto finito (**etiquetas**).
- $\mu$  es una **estructura de membranas** de grado  $q$ , con las membranas etiquetadas biyectivamente con  $\{1, \dots, q\}$ .
- Para cada  $i$ ,  $\mathcal{M}_i$  es un multiconjunto finito sobre  $\Gamma$ .
- Para cada  $i$ ,  $R_i$  es un conjunto de **reglas de evolución** sobre  $\Gamma$  y  $\rho_i$  es un orden parcial estricto sobre  $R_i$ . Las reglas de  $R_i$  (asociadas a la región etiquetada por  $i$ ) son del tipo  $u \rightarrow v$ , con

$$\bullet v = (v_1, here)(v_2, out), (v_3, in_j) \circ v = (v'_1, here)(v'_2, out), (v'_3, in_j)\delta.$$

siendo  $u, v_1, v'_1, v_2, v'_2, v_3, v'_3$  multiconjuntos sobre  $\Gamma$ .

- El número  $i_0$  ( $1 \leq i_0 \leq n$ ) representa la **membrana de salida** del sistema P.

### Semántica de los Sistemas P

**Configuración inicial** de  $\Pi$ :  $(\mu, \mathcal{M}_1, \dots, \mathcal{M}_q)$ .

Una **configuración** de  $\Pi$  es una tupla  $(\mu', \mathcal{M}'_{i_1}, \dots, \mathcal{M}'_{i_k})$ , en donde:

- $\mu'$  se obtiene de  $\mu$  eliminando las membranas distintas de  $i_1, \dots, i_k$ .
- $\mathcal{M}'_{i_1}, \dots, \mathcal{M}'_{i_k}$  son multiconjuntos sobre  $V$ .

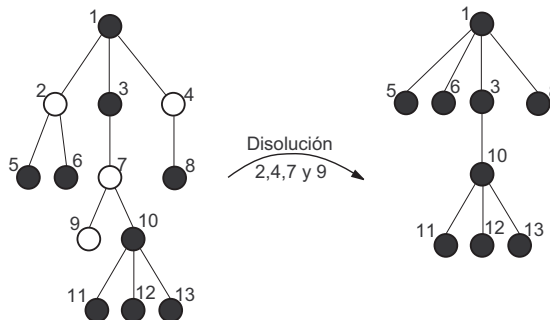
- $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$ , y debe contener la etiqueta asociada a la membrana piel.

Sean  $C_1$  y  $C_2$  dos configuraciones de  $\Pi$ , con

$$C_1 = (\mu', m'_{i_1}, \dots, m'_{i_k}) \text{ y } C_2 = (\mu'', m'_{j_1}, \dots, m'_{j_l})$$

$C_2$  se obtiene de  $C_1$  en un **paso de transición** ( $C_1 \Rightarrow_{\Pi} C_2$ ), si se puede pasar de  $C_1$  a  $C_2$  usando las reglas que aparecen en  $R_{i_1}, \dots, R_{i_k}$  de la siguiente forma:

- Si  $u \rightarrow v$  pertenece a  $R_i$  y el multiconjunto  $u$  aparece en la membrana de  $\mu'$  etiquetada por  $i$ , entonces:
  - El multiconjunto  $u$  se elimina de la membrana  $i$ .
  - Si  $(v_1, here) \in v$ , se añade  $v_1$  a la membrana  $i$ .
  - Si  $(v_1, out) \in v$ , se añade  $v_1$  a la membrana *padre* de  $i$  (al entorno si  $i$  es la piel).
  - Si  $(v_1, in_j) \in v$ , se añade el  $v_1$  a la membrana  $j$  (siempre que  $j$  sea un *hijo* de  $i$ ).
  - Si  $\delta \in v$ , la membrana  $i$  se disuelve y su contenido pasa al primer antecesor no disuelto (la piel **no** se puede disolver).
- **Prioridad en la *versión fuerte***
- **Aplicabilidad** de una regla a una configuración.
  - Condiciones necesarias para que una regla  $u \rightarrow v$  sea aplicable:
    - Existencia de un elemento  $(v_1, in_j) \in v$ .
    - Existencia de un elemento  $\delta \in v$ .
    - Existencia de una regla aplicable de *mayor* prioridad.
- Multiconjuntos de reglas aplicables a una configuración.
- Las reglas se ejecutan en **paralelo**, en forma **maximal** y **no determinista**.





**Computaciones en sistemas P**

Una **computación**  $\mathcal{C}$  es una sucesión (finita o infinita) de configuraciones  $C_0 \Rightarrow_{\Pi} C_1 \Rightarrow_{\Pi} \dots \Rightarrow_{\Pi} C_r$ , con  $r \geq 0$ , tal que:

- $C_0$  es la configuración inicial de  $\Pi$ .
- Para cada  $i \geq 0$ ,  $C_{i+1}$  se obtiene de  $C_i$  por un paso de transición.

Una computación,  $\mathcal{C}$ , es de **parada** si  $r \in \mathbb{N}$  y en la configuración  $C_r$  (denominada *final*) no hay reglas que se puedan aplicar.

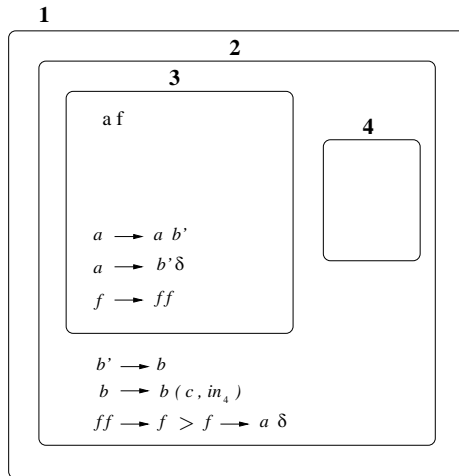
El **resultado** de una computación de *parada* está codificado en la membrana de salida de la configuración final.

Un sistema P básico se puede considerar como:

- Una máquina *generadora*.
- Una máquina de *cálculo*.
- Una máquina de *decisión*.

**3.1 Un ejemplo**

Un sistema celular con membranas que genera el conjunto  $\{n^2 : n \geq 1\}$ .



En donde la membrana 4 es la de salida.

Se analiza las computaciones en función del instante  $m \geq 0$  en el que se aplica la regla  $a \rightarrow b' \delta$  por primera vez.

Paso	Membrana 1	Membrana 2	Membrana 3	Membrana 4
0			$af$	
1			$ab'f^2$	
2			$ab'^2f^{2^2}$	
3			$ab'^3f^{2^3}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$m$			$ab'^mf^{2^m}$	
$m+1$		$b^{(m+1)}f^{2^{m+1}}$	<i>disuelta</i>	
$m+2$		$b^{m+1}f^{2^m}$	<i>disuelta</i>	
$(m+2)+1$		$b^{m+1}f^{2^{m-1}}$	<i>disuelta</i>	$c^{m+1}$
$(m+2)+2$		$b^{m+1}f^{2^{m-2}}$	<i>disuelta</i>	$c^{2(m+1)}$
$(m+2)+3$		$b^{m+1}f^{2^{m-3}}$	<i>disuelta</i>	$c^{3(m+1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(m+2)+m$		$b^{m+1}f^{2^{m-m}}$	<i>disuelta</i>	$c^{m(m+1)}$
$2m+3$	$ab^{m+1}$	<i>disuelta</i>	<i>disuelta</i>	$c^{(m+1)(m+1)}$

#### 4 SISTEMAS P QUE TRABAJAN A MODO DE TEJIDOS

La *división celular* es un proceso elegante que se produce en todo organismo vivo y que les permite crecer y reproducirse. La *mitosis* es un proceso de división celular que da como resultado la producción de dos células hijas a partir de una sola célula original. Las células hijas son idénticas y exactamente igual que la célula madre original. Mediante una secuencia de pasos, el material genético replicado en una célula madre se distribuye por igual entre sus dos células hijas. La mitosis es muy similar en todos los organismos, aunque puede haber diferencias sutiles entre ellos.

El ciclo celular consta, básicamente, de cuatro fases. La primera consiste en una etapa de crecimiento de la célula. Seguidamente se produce una de los procesos esenciales de la Vida: la replicación del ADN. Una enzima, la ADN polimerasa, se encarga de realizar de manera mecánica dos copias exactas del ADN identificador de la célula. A continuación se produce una nueva fase de crecimiento, para finalizar el ciclo celular con la mitosis que da lugar a las dos células hijas.

El proceso mitótico ha sido introducido en el marco de *Membrane Computing* a través de una regla de división de membranas que es una abstracción de la mitosis. De esta manera surgen los *sistemas P con membranas activas*, dispositivos que operan a modo de células, con una estructura jerarquizada implementada a través de un árbol enraizado y en donde existen reglas de comunicación entre membranas, evolución, disolución, amén de la regla de división celular que proporciona al modelo la capacidad de generar una cantidad de espacio exponencial en tiempo lineal. En esta variante de sistemas P, no existe cooperación ni priori-

dad entre reglas; en cambio, las membranas pueden estar cargadas eléctricamente (positiva, neutra o negativamente).

Este marco computacional puede ser extendido a sistemas celulares que están organizados a modo de tejidos. En el nuevo contexto van a existir unidades fundamentales que denominaremos células y un elemento singular que denominaremos *entorno* del sistema. Como es usual, los símbolos de un alfabeto de trabajo nos permitan considerar objetos que serán abstracciones de las sustancias químicas, así como reglas de reescritura que serán abstracciones de las reacciones químicas del sistema. Admitiremos que los objetos situados inicialmente en el entorno aparecen en un número arbitrariamente grande de copias.

Las células del sistema estarán identificadas a través de una etiqueta. Las diferentes células se pueden comunicar entre sí o con el entorno, a través de ciertas reglas. Cada una de ellas proporciona una serie de arcos virtuales bien entre dos células o entre una célula y el entorno. Así mismo, admitiremos la existencia de un tipo de reglas que implementa la división celular, por la cual a partir de una célula se pueden obtener dos células hijas (con la misma etiqueta) de tal manera que el contenido de la madre se hereda a las hijas, con excepción del objeto que dispara la regla. Estos modelos se denominarán *tissue P systems with cell division* y, en ellos, existe una estructura subyacente de grafo dirigido cuyos nodos ordinarios son las células. En el sistema no existen polarizaciones y la dinámica está regida por reglas de comunicación y de división de tal manera que si a una célula se le aplica una regla de división, entonces no se le puede aplicar en ese paso de computación ninguna regla de comunicación; es decir, podemos imaginar que cuando una célula se divide, entonces su interacción con otras células o con el entorno queda bloqueada. En cierto sentido, esto significa que mientras que una célula se divide se cierran sus canales de comunicación.

Vamos a tratar de precisar un poco estas ideas a través de una definición precisa.

**Definición 4.1** *Un sistema de tejidos con división celular de grado  $q \geq 1$  es una tupla  $\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{out})$ , donde:*

1.  $\Gamma$  es un alfabeto finito.
2.  $\mathcal{E} \subseteq \Gamma$ .
3.  $\mathcal{M}_1, \dots, \mathcal{M}_q$  son cadenas sobre  $\Gamma$ .
4.  $\mathcal{R}$  es un conjunto finito de reglas del siguiente tipo:
  - (a) Reglas de Comunicación:  $(i, u/v, j)$ , para  $i, j \in \{0, 1, 2, \dots, q\}, i \neq j$ ,  $u, v \in \Gamma^*, |u + v| \neq 0$ ;
  - (b) Reglas de División:  $[a]_i \rightarrow [b]_i[c]_i$ , donde  $i \in \{1, 2, \dots, q\}, i \neq i_{out}$  y  $a, b, c \in \Gamma$ .
5.  $i_{out} \in \{0, 1, 2, \dots, q\}$ .

Un sistema de tejidos con división celular de grado  $q \geq 1$

$$\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{out})$$

se puede considerar como un cierto conjunto de  $q$  células, etiquetadas de  $1, \dots, q$ , con un entorno que será etiquetado por  $0$ , y de tal manera que:

- (a)  $\mathcal{M}_1, \dots, \mathcal{M}_q$  son cadenas del alfabeto  $\Gamma$  que representan los multiconjuntos finitos de objetos (elementos de  $\Gamma$ ) inicialmente colocados en las  $q$  células del sistema;
- (b)  $\mathcal{E}$  es el conjunto de objetos inicialmente localizados en el entorno del sistema, todos ellos aparecen en un número arbitrario de copias;
- (c)  $i_{out} \in \{0, 1, 2, \dots, q\}$  representa una región distinguida que nos va a permitir codificar la salida del sistema. Usaremos el termino *región  $i$*  ( $0 \leq i \leq q$ ) para referirnos a la célula  $i$ , en el caso de  $1 \leq i \leq q$ , o al entorno, en el caso de  $i = 0$ .

Cuando se aplica una regla  $(i, u/v, j)$ , los objetos del multiconjunto representado por  $u$  son enviados desde la región  $i$  a la región  $j$  y, al mismo tiempo, los objetos del multiconjunto  $v$  son enviados desde la región  $j$  a la región  $i$ . La longitud de la regla de comunicación  $(i, u/v, j)$  es definida como  $|u| + |v|$ , es decir, es el número total de objetos que aparecen en la regla.

Una regla de la comunicación  $(i, u/v, j)$  se dice que es una regla del tipo *symport* si  $u = \lambda$  o  $v = \lambda$ . Toda regla del tipo *symport rule*  $(i, u/\lambda, j)$ , con  $i \neq 0, j \neq 0$ , proporciona un arco virtual desde la célula  $i$  a la célula  $j$ . Una regla de la comunicación  $(i, u/v, j)$  se dice que es del tipo *antiport* si  $u \neq \lambda$  y  $v \neq \lambda$ . Toda regla del tipo *antiport*  $(i, u/v, j)$ , con  $i \neq 0, j \neq 0$ , proporciona dos arcos: uno que va de la célula  $i$  a la célula  $j$  y otro que va de la célula  $j$  a la célula  $i$ . Por tanto, a cada *tissue P system* se le puede asociar un grafo dirigido subyacente cuyos nodos son las células del sistema y los arcos se obtienen a partir reglas de la comunicación. En este contexto, el medio ambiente (el entorno) puede ser considerado como un nodo virtual del grafo de tal manera que sus conexiones están definidas por reglas de la comunicación de la forma  $(i, u/v, j)$ , con  $i = 0$  or  $j = 0$ .

Cuando se aplica una regla de la división  $[a]_i \rightarrow [b]_i[c]_i$ , en presencia del objeto  $a$ , la célula etiquetada por  $i$  se divide en dos nuevas células con la misma etiqueta, de tal manera que en la primera copia, el objeto  $a$  es reemplazado por objeto  $b$ , y en la segunda copia, el objeto  $a$  se sustituye por el objeto  $c$ . Además, los restantes objetos existentes en la célula que se divide se replican y se copian en las dos células nuevas. La célula de salida  $i_{out}$  no puede ser dividida.

Las reglas de un sistema P de tejidos se ejecutan en paralelo, de manera no determinista y maximal, tal y como suele ser habitual en *Membrane Computing*. En cada paso de computación, todas las células que puedan evolucionar deben evolucionar de forma paralela maximal: en cada paso se aplica un multiconjunto de reglas que es maximal respecto de la relación de inclusión; es decir, tal que tras la aplicación de dicho multiconjunto de reglas al sistema, no existe ninguna otra

regla que se le pueda aplicar al conjunto de objetos que quedan. En la aplicación antes citada de las reglas existe una sólo restricción: si en una célula se está ejecutando una regla de división, entonces en ese paso sólo se aplicará esa regla; es decir, los objetos situados dentro de esa célula no pueden evolucionar por medio de reglas de comunicación. En otras palabras, podemos interpretar que la aplicación de una regla de división de una célula interrumpe todas las comunicaciones con otras células y con el entorno. Las nuevas células resultante de la división sólo van a interactuar con otras células o con el entorno en el siguiente paso, siempre y cuando no se dividan otra vez. La etiqueta de una célula identifica con precisión las reglas que pueden ser aplicadas.

Un *configuración o descripción instantánea* de un sistema  $P$  de tejidos en un instante determinado está descrito por todos los multiconjuntos de objetos sobre el alfabeto  $\Gamma$  asociados a todas las células presentes en el sistema, así como el multiconjunto de objetos sobre  $\Gamma - \mathcal{E}$  asociados al entorno en ese momento. Teniendo en cuenta que los objetos del  $\mathcal{E}$  aparecen en un número infinito de copias en la configuración inicial del sistema, esos objetos del entorno podremos considerar que no son alterados por la aplicación de reglas. La *configuración inicial* es  $(\mathcal{M}_1, \dots, \mathcal{M}_q; \emptyset)$ . Diremos que una configuración es de *parada* si no existe ninguna regla del sistema que pueda ser aplicada a esa configuración.

Dado un sistema  $P$  de tejidos  $\Pi$  con división celular, diremos que de una configuración  $\mathcal{C}_1$  se pasa a otra configuración  $\mathcal{C}_2$  a través de un paso de computación, y lo notaremos por  $\mathcal{C}_1 \Rightarrow_{\Pi} \mathcal{C}_2$ , si es posible pasar de  $\mathcal{C}_1$  a  $\mathcal{C}_2$  mediante la aplicación de reglas pertenecientes a  $\mathcal{R}$  de una manera paralela y maximal. Una *computación* de  $\Pi$  es una sucesión, finita o infinita, de configuraciones del sistema que verifica las siguientes condiciones:

1. El primer término de la sucesión es una configuración inicial del sistema.
2. Cada configuración distinta de la configuración inicial del sistema, se obtiene a partir de la configuración anterior mediante la aplicación de las reglas del sistema de manera paralela, maximal y no determinista, con las restricciones antes citadas.
3. Si la sucesión es finita, en cuyo caso diremos que la computación es de *parada*, entonces el último término de la sucesión es una *configuración de parada*.

Todas las computaciones comienzan con una configuración inicial y evolucionan de acuerdo con lo indicado anteriormente. Únicamente las *computaciones de parada* proporcionan un resultado que estará codificado por los objetos presentes en una cierta región de salida, que puede ser una célula ordinaria o bien el entorno, en la *configuración de parada*.

## 5 SISTEMAS P QUE TRABAJAN A MODO DE NEURONAS

En esta última sección, veremos cómo definir un modelo de computación inspirado en la manera en que las neuronas están interconectadas formando una red, y en cómo se comunican entre sí enviándose impulsos eléctricos. Esta adaptación de la Computación con Membranas a la arquitectura de estilo red neuronal se conoce como *Spiking Neural P systems* (o sistemas SNP). En los sistemas SNP, las unidades básicas de procesamiento se denominan *neuronas*, y el grafo dirigido que representa las interconexiones existentes (siendo las neuronas los nodos del grafo) se conoce como *grafo de sinapsis*. La operación básica de computación en este modelo es el envío de “impulsos eléctricos” entre las neuronas a través de las sinapsis. Dichos impulsos se representan en el modelo usando un alfabeto con un único símbolo, llamado *spike*. Cada neurona puede almacenar en su interior una o más copias de este objeto especial, y también pueden tener reglas para enviar impulsos a sus vecinas (a veces con un cierto tiempo de espera asociado). Las reglas que envían objetos se llaman reglas de “disparo” (*firing rules*), mientras que las reglas que hacen desaparecer objetos se llaman reglas de “olvido” (*forgetting rules*).

En cada paso de computación, si una neurona tiene alguna regla aplicable, entonces obligatoriamente debe aplicar alguna de dichas reglas. En caso de que hubiera varias reglas aplicables, se elegirá una de ellas de manera aleatoria (no determinista). Es decir, las reglas se aplican de manera secuencial en cada neurona, pero las neuronas trabajan de forma paralela. Se asume la existencia de un reloj global, que marca los pasos de computación en todo el sistema, y por tanto el sistema actúa de forma síncrona.

Partiendo de la primera versión presentada de este modelo [4], varias características biológicas han sido exploradas para incorporarlas al marco de los sistemas SNP. Una de estas extensiones, con motivación matemática, se introdujo en [1], donde se permite que las reglas disparen más de un impulso, aunque el número de objetos emitidos no puede superar al de los consumidos. Otras de las variantes consideradas incluyen astrocitos [5], pesos asociados a las conexiones [3, 6, 11], anti-impulsos (anti-spikes) [7], o división de neuronas [10]. En esta unidad vamos a centrarnos en los sistemas SNP con pesos en las sinapsis y sin retraso en las reglas<sup>1</sup>.

**Definición 5.2** *Un sistema SNP, o spiking neural P system, de grado  $m \geq 1$  viene definido por una tupla de la forma*

$$\Pi = (O, \sigma_1, \sigma_2, \dots, \sigma_m, \text{syn}, \text{in}),$$

donde:

1.  $O = \{a\}$  es un alfabeto unitario (cada objeto  $a$  se denominará spike);

<sup>1</sup>Para una descripción más general, véase [4].

2.  $\sigma_1, \sigma_2, \dots, \sigma_m$  son neuronas, de la forma  $\sigma_i = (n_i, R_i)$ ,  $1 \leq i \leq m$ , donde:

a)  $n_i \geq 0$  es el número inicial de spikes que contiene  $\sigma_i$ ;

b)  $R_i$  es un conjunto finito de reglas de uno de estos dos tipos:

(1) Firing rules  $E/a^c \rightarrow a^d$ , donde  $E$  es una expresión regular<sup>2</sup> sobre  $a$ , y  $c \geq d \geq 1$  son dos números naturales; en el caso de que  $E = a^b$  (siendo  $b \geq c$  otro número natural), entonces la regla quedaría escrita del siguiente modo:  $a^b/a^c \rightarrow a^d$ ;

(2) Forgetting rules  $a^b/a^c \rightarrow \lambda$ , siendo  $b \geq c \geq 1$  dos números naturales.

3.  $syn \subseteq \{1, 2, \dots, m\} \times \{1, 2, \dots, m\} \times \mathbb{W}$  es el conjunto de sinapsis entre neuronas, siendo el conjunto de posibles pesos  $\mathbb{W} = \mathbb{Q} \cap [0, 1]$ , i.e., el conjunto de números racionales entre 0 y 1. El conjunto de sinapsis también verifica que  $(i, i, k) \notin syn$  para cualquier  $1 \leq i \leq m, k \in \mathbb{W}$ .

4.  $in$  es el índice o etiqueta de la neurona de entrada de  $\Pi$ .

Una regla de disparo  $E/a^c \rightarrow a^d \in R_i$  es aplicable en una neurona  $\sigma_i$  si dicha neurona contiene  $b$  spikes,  $b \geq c$ , y  $a^b$  pertenece al lenguaje asociado con  $E$ . Para aplicar una regla de tipo  $a^b/a^c \rightarrow a^d$ , la neurona debe contener exactamente  $b$  spikes. La ejecución de una de estas reglas conllevaría eliminar  $c$  spikes de  $\sigma_i$  y enviar  $d$  spikes a todas las neuronas  $\sigma_j$  tales que exista una sinapsis  $(i, j, k) \in syn$ . El número de spikes que llegarán a la neurona  $\sigma_j$  a través de una sinapsis  $(i, j, k)$  depende del número  $d$  de spikes emitidas, así como de la "calidad de la conexión", representada por el peso  $k$ . Concretamente, la contribución de  $\sigma_i$  a  $\sigma_j$  se obtiene multiplicando ambos valores y redondeando hacia abajo ( $\lfloor d \times k \rfloor$  spikes).

Una regla de olvido  $a^b/a^c \rightarrow \lambda$  es aplicable en una neurona  $\sigma_i$  si ésta contiene exactamente  $b$  spikes. La ejecución de esta regla consiste simplemente en eliminar  $c$  spikes de  $\sigma_i$  (y dejando por tanto  $b - c$  spikes).

En cada neurona, el número de spikes que quedan después de ejecutar un paso de computación se obtiene sumando el número de spikes del paso anterior no consumidos por ninguna regla junto con las contribuciones recibidas de las otras neuronas a través de las sinapsis correspondientes.

Una configuración del sistema viene dada por un vector de números que indican cuántos spikes hay en cada neurona. Al principio de una computación, el número de spikes presentes en cada neurona  $\sigma_i$  es  $n_i$ , salvo en la neurona de entrada (etiquetada por  $in$ ), que contendrá  $n_{in} + N$  (siendo  $N$  la entrada recibida para esa computación).

**Ejemplo 5.3** Consideremos el sistema SNP  $\Pi = (\{a\}, \sigma_1, \sigma_2, \sigma_3, syn, 1)$  (ver la Figura 3) con  $\sigma_1 = (7, R_1)$ ,  $\sigma_2 = (3, R_2)$ ,  $\sigma_3 = (8, R_3)$ , y cuyos conjuntos de reglas y de sinapsis son:

<sup>2</sup>Nos referimos a expresiones regulares del tipo  $a^n a^*$ , con  $n \in \mathbb{N}$ , cuyo lenguaje asociado es el conjunto  $\{a^{n+k} \mid k \in \mathbb{N}\}$ . Para más detalles acerca de las expresiones regulares, véase por ejemplo [9].

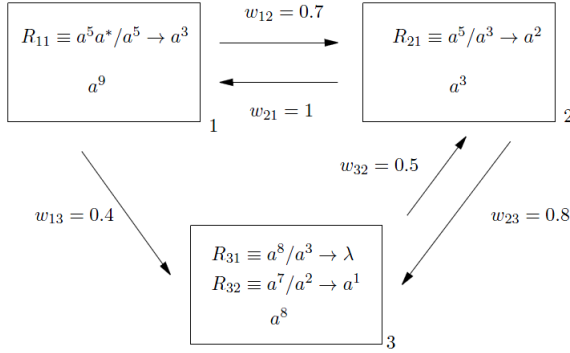


FIGURA 3: Ejemplo. La neurona de entrada 1 tiene  $9 = 7+2$  spikes en la configuración inicial.

$$\begin{aligned}
 R_1 &= \{R_{11} \equiv a^5 a^* / a^5 \rightarrow a^3\} \\
 R_2 &= \{R_{21} \equiv a^5 / a^3 \rightarrow a^2\} \\
 R_3 &= \{R_{31} \equiv a^8 / a^3 \rightarrow \lambda, \quad R_{32} \equiv a^7 / a^2 \rightarrow a^1\}
 \end{aligned}$$

$$syn = \{(1, 2, 0,7), (2, 1, 1), (2, 3, 0,8), (3, 2, 0,5), (1, 3, 0,4)\}$$

Consideremos como entrada  $N = 2$ . Entonces, antes de empezar la computación contamos con 9 spikes ( $7+2$ ) en la neurona 1. Nótese que la regla  $R_{11}$  es aplicable siempre que la neurona 1 contenga al menos 5 spikes.  $R_{31}$  es la única regla de olvido del sistema SNP.

Como hemos dicho, al ser la entrada  $N = 2$ , tenemos que la configuración inicial es  $C_0 = \langle 9, 3, 8 \rangle$ . Sobre ella son aplicables las reglas  $R_{11}$  y  $R_{31}$ . La regla  $R_{11}$  consume 5 spikes de la neurona 1 y emite 3 spikes hacia las neuronas 2 y 3 (que se multiplicarán por los correspondientes pesos antes de llegar a su destino). El peso de la sinapsis entre las neuronas 1 y 2 es  $w_{12} = 0,7$ , así que la ejecución de la regla  $R_{11}$  produce un incremento de  $\lfloor 3 \times 0,7 \rfloor = 2$  spikes en la neurona 2. Análogamente, puesto que  $w_{13} = 0,4$ , la aplicación de la regla  $R_{11}$  contribuye con  $\lfloor 3 \times 0,4 \rfloor = 1$  spike a añadir en la neurona 3. La regla de olvido  $R_{31}$  elimina 3 spikes de la neurona 3 y por tanto, la configuración obtenida tras el primer paso de computación es  $C_1 = \langle 4, 5, 6 \rangle$ .

En este momento, la única regla aplicable es  $R_{21}$ . Esta regla consume 3 spikes de la neurona 2 y envía 2 spikes hacia las neuronas 1 y 3. Teniendo presente los correspondientes pesos, el número de spikes en la neurona 1 se ve incrementado en  $\lfloor 2 \times 1 \rfloor = 2$ , mientras que el número de spikes en la neurona 3 se ve incrementado en  $\lfloor 2 \times 0,8 \rfloor = 1$ . Una vez que efectuamos estas modificaciones, la configuración obtenida es  $C_2 = \langle 6, 2, 7 \rangle$ .

En este caso, las reglas aplicables son  $R_{11}$  y  $R_{32}$ . La regla  $R_{11}$  ya se ha descrito previamente. La regla  $R_{32}$  consume 2 spikes de la neurona 3 y envía 1 spike hacia la neurona 2. Ahora bien, dado que  $w_{32} = 0,5$ , este envío no genera incremento en el número de spikes de la neurona 2, ya que  $\lfloor 0,5 \times 1 \rfloor = 0$ . Teniendo en cuenta estos cambios, la siguiente configuración obtenida es  $C_3 = \langle 1, 4, 4 \rangle$ . Ya no hay ninguna regla aplicable, y  $C_3$  es por tanto la configuración de parada.



## BIBLIOGRAFÍA

- [1] Chen, H., Ionescu, M., Ishdorj, T.O., Păun, A., Păun, Gh., Pérez-Jiménez, M.: Spiking neural P systems with extended rules: universality and languages. *Natural Computing* 7, 147–166 (2008)
- [2] Corne, D.W., Frisco, P., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): *Membrane Computing - 9th International Workshop, WMC 2008, Edinburgh, UK, July 28-31, 2008, Revised Selected and Invited Papers, Lecture Notes in Computer Science*, vol. 5391. Springer (2009)
- [3] Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Hebbian learning from spiking neural P systems view. In: Corne et al. [2], pp. 217–230
- [4] Ionescu, M., Păun, Gh., Yokomori, T.: Spiking neural P systems. *Fundamenta Informaticae* 71(2-3), 279–308 (2006)
- [5] Pan, L., Wang, J., Hoogeboom, H.: Spiking neural P systems with astrocytes. 24(3), 805–825 (2012)
- [6] Pan, L., Zeng, X., Zhang, X., Jiang, Y.: Spiking neural P systems with weighted synapses. *Neural Processing Letters* 35(1), 13–27 (2012)
- [7] Pan, L., Păun, Gh.: Spiking neural P systems with anti-spikes. *International Journal of Computers, Communications & Control* IV(3), 273–282 (September 2009)
- [8] Pérez Jiménez, M.J. y Sancho Caparrini, F. *Computación celular con membranas: Un modelo no convencional*. Editorial Kronos, Sevilla, 2002.
- [9] Rozenberg, G., Salomaa, A.: *Handbook of Formal Languages: Word, language, grammar*. Handbook of Formal Languages, Springer (1997)
- [10] Wang, J., Hoogeboom, H.J., Pan, L.: Spiking neural P systems with neuron division. In: Gheorghe, M., Hinze, T., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) *Int. Conf. on Membrane Computing*. Lecture Notes in Computer Science, vol. 6501, pp. 361–376. Springer, Berlin Heidelberg (2010)
- [11] Wang, J., Hoogeboom, H.J., Pan, L., Păun, Gh., Pérez-Jiménez, M.J.: Spiking neural P systems with weights. *Neural Computation* 22(10), 2615–46 (2010)