

# 7

## PBE. MODELOS EN DINÁMICA DE POBLACIONES

Esta unidad tiene como objetivo aplicar la metodología de modelización y simulación descrita en unidades anteriores y el patrón PBE a la dinámica de poblaciones, concretamente al estudio de la evolución de ecosistemas reales y al control de una pandemia. Comenzaremos estudiando en más detalle un modelo teórico sobre una cadena trófica por considerarse un poco más sencillo de entender, y posteriormente proporcionaremos los ejemplos reales en toda su dimensión.

### 1 Ejemplo 5. Interacciones tritróficas

Este ejemplo extraído de [9] y [8] se enmarca en el estudio de la dinámica de poblaciones, mostrando el proceso de modelización y simulación de un ecosistema de literatura implicando una cadena trófica. Describimos en detalle este modelo por considerarlo suficientemente ilustrativo del funcionamiento de este tipo de poblaciones, presenta los elementos fundamentales que se suelen dar, y no es tan complejo como los casos reales que podemos encontrar (se hará referencia a ellos en los ejemplos 7 y 8, de modo que el alumno pueda consultar la documentación disponible, pero no se exigirán resultados acerca de ellos).

#### 1.1 Escenario

Modelizar un sistema biológico es generalmente muy complicado, si uno quiere tener en cuenta los factores más importantes involucrados y las relaciones entre ellos. Por lo tanto es normal definir un escenario en el cuál el número de variables e interacciones entre las mismas sea tan limitado como sea posible.

Hoy en día las ecuaciones diferenciales ordinarias (EDOs) constituyen la alternativa más ampliamente utilizada para el estudio de dinámicas de poblaciones, pero esta aproximación tiene algunos inconvenientes. Por un lado, cuando el número de especies en un modelo es mayor que 2, el sistema de ecuaciones propuesto es tan complejo que se suele resolver usando métodos numéricos. Además, las mejoras en el comportamiento de los modelos se obtienen general-

mente añadiendo nuevos ingredientes, lo que en el caso de las EDOs puede implicar tener que llevar a cabo de nuevo el proceso de modelización completo prácticamente desde cero.

Ya existe una literatura bastante amplia acerca de algunas alternativas para modelizar diferentes fenómenos dentro del marco de la computación celular con membranas (ver por ejemplo [1, 5, 6] y el capítulo dedicado a *Probabilistic/stochastic models* en el libro [4]). Los modelos de computación basados en sistemas P ofrecen ventajas significativas: modularidad, paralelismo y no limitación sobre el número de variables interrelacionadas que evolucionan de forma simultánea. Estas propiedades los hacen muy atractivos para la modelización de ecosistemas complejos.

Cada ecosistema tiene sus propias peculiaridades importantes. Sin embargo, hay una serie de aspectos comunes a considerar en la mayor parte de los ecosistemas, como pueden ser:

- Contienen un gran número de especies un gran número de individuos de cada especie.
- El ciclo de vida incluye algunos procesos básicos tales como: alimentación, crecimiento, reproducción y muerte.
- Los procesos que tienen lugar se suelen repetir de forma cíclica (podría ser anual, mensual, diaria, etc.).
- La evolución depende a menudo del entorno: condiciones climáticas, de suelo, etc.
- La dinámica natural sufre modificaciones debido a determinadas actividades humanas.

Estas características comunes establecen algunos requisitos para el modelo, desde un punto de vista computacional: tienen lugar muchos procesos simultáneamente, hay cooperación entre individuos y elementos del ecosistema, es necesaria una sincronización parcial de los subsistemas (por ejemplo, podrían darse unas condiciones meteorológicas adversas un año, y esto no afectaría a un solo subsistema sino que tendría una influencia global sobre el ecosistema completo), de modo que las condiciones iniciales se vean actualizadas de forma global de forma periódica (anual, mensual, diaria, etc.).

Estas consideraciones llevan a la definición de un contexto semántico apropiado para el sistema P, que se ha denominado "PDP system" (Population Dynamics P system). Este tipo de sistemas se ha empleado para modelizar dos ecosistemas reales: uno tratando con *aves carroñeras* en el Pirineo catalán [7] y otro centrado en el estudio del *mejillón cebra* en el pantano de Ribarroja [3]. En el primer caso se modeliza una especie en peligro de extinción, con el propósito de estudiar la evolución del ecosistema bajo diferentes escenarios para ayudar en la toma de decisiones de gestión más adecuadas e informadas para la conservación de la especie. El segundo caso se corresponde con una situación completamente

diferente: el mejillón cebra es una especie exótica invasora que ha mostrado una increíble capacidad de adaptación desde que apareció en el embalse. Su reproducción incontrolada causa un daño económico y ecológico muy significativo.

En ambos casos se diseñó un simulador para validar los resultados, de modo que ahora los gestores disponen de dos herramientas que les permiten llevar a cabo experimentos virtuales bajo diferentes condiciones.

En este primer ejemplo se tratan de evitar las particularidades y excesiva complejidad de los ecosistemas reales, proporcionando un ecosistema idealizado que pueda ser usado a la hora de testear los modelos y simuladores, en lugar de ejecutar pruebas sobre enormes instancias del problema mediante casos reales. El ecosistema presentado contiene especies perteneciendo a tres niveles tróficos diferentes: en el nivel inferior tenemos el pasto, la hierba, en el siguiente tenemos 5 especies de herbívoros y finalmente arriba del todo tenemos una especie carnívora en el tercer nivel. Se trata como decimos de un ecosistema idealizado, en el sentido de que no se corresponde con ningún ecosistema real. Sin embargo, se ha diseñado bajo la dirección de ecólogos, y tiene en cuenta algunos hechos relevantes que lo hacen de alguna forma plausible. Algunos de esos detalles se comentarán posteriormente.

Los apartados siguientes introducirán el marco de modelización de los sistemas PDP, con diferencias significativas respecto a otros sistemas P que hemos visto en unidades anteriores, basados en células (de transición, membranas activas) o en tejidos. Posteriormente se presentará el modelo diseñado haciendo uso de este paradigma, se realizarán las consideraciones pertinentes acerca de la simulación y se discutirán los resultados esperados.

## 1.2 Modelo

La subsección 1.0.1 describe el marco de modelización general basado en sistemas P empleado en el caso de los ecosistemas. A continuación se presenta el modelo en la subsección 1.0.2.

### 1.0.1. Sistemas P de Dinámica de Poblaciones

Los *sistemas P de Dinámica de Poblaciones* constituyen una variante de sistemas P que trabajan a modo de tejidos y proporcionan un marco para la modelización de procesos relacionados con la dinámica de poblaciones, en general. Estos sistemas deben ser capaces de capturar la aleatoriedad inherente a dichos procesos a través de una semántica basada en reglas de reescritura que pueden ser aplicadas en determinadas circunstancias y con una cierta probabilidad. Las células que integran el tejido se denominarán entornos y cada uno de ellos contiene un sistema P que trabaja a modo de células de tal manera que el esqueleto (alfabeto, estructura de membranas y reglas) son idénticos y las membranas de esa estructura común tienen cargas eléctricas. Además, en cada entorno, el sistema P que

contiene tendrá unos multiconjuntos iniciales específicos y cada regla tendrá asociada una función de probabilidad también específica del entorno.

**Definición 1.1** *Un sistema P de Dinámica de Poblaciones (PDP) de grado  $(q, m)$ ,  $q, m \geq 1$ , tomando  $T \geq 1$  unidades de tiempo, es una tupla*

$$\Pi = (G, \Gamma, \Sigma, T, \mathcal{R}_E, \mu, \mathcal{R}, \{f_{r,j} : r \in \mathcal{R}, 1 \leq j \leq m\}, \{\mathcal{M}_{i,j} : 1 \leq i \leq q, 1 \leq j \leq m\})$$

donde:

- $G = (V, S)$  es un grafo dirigido. Sea  $V = \{e_1, \dots, e_m\}$ .
- $\Gamma$  y  $\Sigma$  son alfabetos tales que  $\Sigma \subsetneq \Gamma$ .
- $T$  es un número natural.
- $\mathcal{R}_E$  es un conjunto finito de reglas del tipo  $(x)_{e_j} \xrightarrow{p_r} (y_1)_{e_{j_1}} \cdots (y_h)_{e_{j_h}}$ , donde  $x, y_1, \dots, y_h \in \Sigma$ ,  $(e_j, e_{j_l}) \in S$ ,  $1 \leq l \leq h$ , y  $p_r$  es una función computable de  $\{1, \dots, T\}$  en  $[0, 1]$  tal que para cada  $e_j \in V$  y  $x \in \Sigma$ , la suma de las funciones asociadas a las reglas del tipo antes descrito es la función constante 1.
- $\mu$  es un árbol enraizado con  $q$  nodos etiquetados de forma inyectiva por elementos del conjunto  $\{1, \dots, q\} \times \{0, +, -\}$ . Si la etiqueta de una membrana es  $(i, \alpha)$ , entonces notaremos  $[\ ]_i^\alpha$  y diremos que la membrana tiene etiqueta  $i$  y carga eléctrica  $\alpha$ . La raíz del árbol está etiquetada por  $(1, 0)$ .
- $\mathcal{R}$  es un conjunto finito de reglas de la forma  $u[v]_i^\alpha \rightarrow u'[v']_i^{\alpha'}$ , donde  $u, v, u', v' \in M_f(\Gamma)$ ,  $u+v \neq \emptyset$ ,  $1 \leq i \leq q$  y  $\alpha, \alpha' \in \{0, +, -\}$ , tal que si  $(x)_{e_j} \xrightarrow{p_r} (y_1)_{e_{j_1}} \cdots (y_h)_{e_{j_h}}$  es una regla de  $\mathcal{R}_E$ , entonces no existe una regla de  $\mathcal{R}$  cuya parte izquierda sea del tipo  $u[v]_1^\alpha$  con  $x \in u$ .
- Para cada  $r \in \mathcal{R}$  y  $1 \leq j \leq m$ ,  $f_{r,j}$  es una función computable de  $\{1, \dots, T\}$  en  $[0, 1]$  tal que para cada  $u, v \in M_f(\Gamma)$ ,  $1 \leq i \leq q$  y  $\alpha, \alpha' \in \{0, +, -\}$ , la suma de las funciones  $f_{r,j}$  con  $r \equiv u[v]_i^\alpha \rightarrow u'[v']_i^{\alpha'}$ , es la función constante 1.
- Para cada  $i, j$  ( $1 \leq i \leq q, 1 \leq j \leq m$ ),  $\mathcal{M}_{i,j}$  es un multiconjunto finito sobre  $\Gamma$ .

Un sistema PDP se puede considerar como un conjunto de  $m$  entornos  $e_1, \dots, e_m$  interconectados por los arcos de un grafo dirigido  $G$ . Cada entorno  $e_j$  sólo puede contener símbolos del alfabeto  $\Sigma$  y, además, contiene un sistema P,  $\Pi_j = (\Gamma, \mu, \mathcal{M}_{1,j}, \dots, \mathcal{M}_{q,j}, \mathcal{R}, \{f_{r,j} : r \in \mathcal{R}\})$ , de grado  $q$ , donde:

- (a)  $\Gamma$  es el alfabeto de trabajo cuyos elementos son denominados objetos.
- (b)  $\mu$  es un árbol enraizado que describe una estructura de  $q$  membranas etiquetadas de forma inyectiva por  $1, \dots, q$  a las que se le asocia una carga eléctrica del conjunto  $\{0, +, -\}$ . La membrana piel (raíz del árbol) está etiquetada por 1.

- (c)  $\mathcal{M}_{1,j}, \dots, \mathcal{M}_{q,j}$  son multiconjuntos finitos sobre  $\Gamma$  que describen los objetos ubicados inicialmente en las regiones de  $\mu$ , dentro del entorno  $e_j$ .
- (d)  $\mathcal{R}$  es el conjunto de reglas del sistema P, con la singularidad de que en el entorno  $e_j$  cada una de esas reglas tiene asociada una función de probabilidad  $f_{r,j}$ . Notaremos  $\mathcal{R}_{\Pi_j}$  el conjunto  $\mathcal{R}$  junto con las funciones  $f_{r,j}$  asociadas.

Además, un sistema PDP dispone de un conjunto  $\mathcal{R}_E$  de reglas de comunicación entre entornos; por tanto, el conjunto total de reglas es  $\bigcup_{j=1}^m \mathcal{R}_{\Pi_j} \cup \mathcal{R}_E$ . El número natural  $T$  representa el tiempo de simulación del sistema.

La *semántica* de los sistemas PDP está definida a través de un modelo paralelo, no determinista y sincronizado, en el sentido de que admitimos la existencia de un reloj global. A continuación describimos los aspectos semánticos de estos sistemas.

Una regla  $r \in \mathcal{R}$  del tipo  $u[v]_i^\alpha \rightarrow u'[v']_i^{\alpha'}$  es *aplicable* a toda membrana etiquetada por  $i$  cuya carga eléctrica sea  $\alpha$  y de tal manera que contiene al multiconjunto  $v$  y su membrana padre (el entorno en el caso de que la membrana sea la raíz del árbol) contiene el multiconjunto  $u$ . La ejecución de dicha regla produce los siguientes efectos: (a) los multiconjuntos de objetos  $v$  y  $u$  se eliminan de la membrana  $i$  y de su membrana padre, respectivamente; (b) de manera simultánea, el multiconjunto  $u'$  se añade a la membrana padre de  $i$  y el multiconjunto  $v'$  se añade a la membrana  $i$ ; y (c) la nueva carga eléctrica de la membrana  $i$  pasa a ser  $\alpha'$  (que puede coincidir con  $\alpha$ ). Recuérdese que en cada entorno  $e_j$ , la regla  $r$  tiene asociada una función de probabilidad  $f_{r,j}$  que proporciona un índice de probabilidad para que la regla sea aplicada, en el caso en que varias reglas compitan por objetos. Es interesante observar que a la hora de aplicar un cierto conjunto de reglas a una cierta membrana, las cargas eléctricas de la parte derecha de esas reglas deben coincidir.

Una regla  $r \in \mathcal{R}_E$ , del tipo  $(x)_{e_j} \xrightarrow{pr} (y_1)_{e_{j_1}} \dots (y_h)_{e_{j_h}}$  es aplicable al entorno  $e_j$  si éste contiene el objeto  $x$ . La ejecución de dicha regla produce los siguientes efectos: (a) el objeto  $x$  es eliminado del entorno  $e_j$ ; y (b) en los entornos  $e_{j_1}, \dots, e_{j_h}$  se añaden los objetos  $y_1, \dots, y_h$  respectivamente. Téngase presente que en cualquier instante  $t$ ,  $1 \leq t \leq T$ , para cada objeto  $x$  en el entorno  $e_j$ , si existen reglas de comunicación del tipo  $(x)_{e_j} \xrightarrow{pr} (y_1)_{e_{j_1}} \dots (y_h)_{e_{j_h}}$ , entonces alguna de ellas será aplicada. En caso de existir varias cuya parte izquierda sea  $(x)_{e_j}$ , entonces se aplicarán de acuerdo con las correspondientes probabilidades asociadas en ese instante  $t$ .

Una *descripción instantánea* o *configuración* del sistema en cualquier instante  $t$  es una tupla de los multiconjuntos de objetos presentes en los  $m$  entornos en cada una de las regiones de cada  $\Pi_j$ , junto a las polarizaciones de las membranas en cada sistema P. Asumiremos que todos los entornos están inicialmente vacíos y que todas las membranas inicialmente tienen polarización neutral.

En cada unidad de tiempo, una configuración dada se transformará en otra configuración mediante la aplicación de reglas del sistema como sigue: en cada paso de transición, las reglas a aplicar son seleccionadas de forma no determinista de acuerdo a las probabilidades asignadas a ellas, y todas las reglas aplicables

son aplicadas simultáneamente de forma maximal. De esta forma, obtenemos una *transición* desde una configuración a la siguiente del sistema. Diremos que una *computación* es una secuencia de configuraciones de tal forma que el primer término de la secuencia es la configuración inicial, y cada configuración no inicial se obtiene de la anterior mediante la aplicación de reglas.

En [11] se presentó el marco general basado en sistemas P para la modelización de dinámica de ecosistemas, el cual ha sido usado para ecosistemas reales, como en el de unas aves carroñeras del Pirineo Catalán [7], y el mejillón cebrado en el pantano de Ribarroja [11]. El mencionado marco de modelización tiene asociadas unas herramientas informáticas que ofrecen, entre otras, las siguientes características: una interfaz de usuario gráfica (entendiéndose por usuario los expertos ecólogos y los diseñadores de modelos), definición del modelo y parámetros del ecosistema, ejecución de simulaciones, creación de datos estadísticos en forma de tablas, gráficas etc. Esta aplicación se denomina MeCoSim [10], y su núcleo es pLinguaCore [12], el cual carga un fichero definido en P-Lingua, configura los correspondientes parámetros, llama a pLinguaCore y colecciona los resultados de la simulación.

Como hemos visto en los capítulos anteriores, a fin de simular un modelo de sistemas P es necesario definir algoritmos de simulación que sean capaces de representar los elementos sintácticos (estructura de membranas, multiconjunto de objetos, reglas, etc.) y de reproducir fielmente la semántica (cómo se aplican las reglas) del modelo. Con respecto a los sistemas PDP, remarcamos las siguientes propiedades semánticas que deberían de ser reflejadas en un algoritmo de simulación: comportamiento probabilístico (reproducción de la aleatoriedad inherente a los procesos de la naturaleza), competición de recursos (las reglas dictaminan como evolucionan grupos de individuos, y además, cómo éstos compiten entre sí), maximalidad del modelo, y consistencia de reglas.

### 1.0.2. Un modelo basado en sistemas P sobre interacciones tritróficas

La alimentación, la reproducción y la mortalidad son procesos básicos para todos los organismos vivos. Estos procesos naturales son los principales que se van a considerar en el modelo de interacciones tritróficas mostrado aquí. El modelo es una simplificación del ecosistema real y se han dejado al margen muchos detalles. Debería considerarse como una aproximación simple facilitando una mejor comprensión de la metodología que permite modelizar ecosistemas reales a través de sistemas P (ver por ejemplo [7] y [3] donde se han modelizado dos ecosistemas reales de alto interés ecológico).

El esquema del modelo está compuesto por 5 módulos como muestra la figura 1. Un ciclo completo se ejecuta en 9 pasos de computación y representa un año en el ecosistema. Dividiremos el ecosistema en 10 áreas con diferentes condiciones climáticas, orográficas y de suelo. Los animales solamente se mueven entre distintas áreas si existe escasez de recursos para su alimentación.

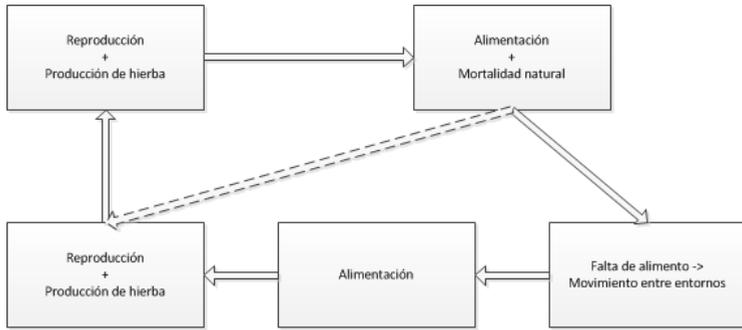


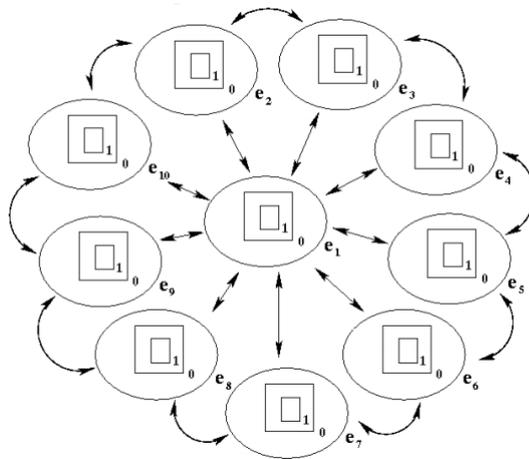
FIGURA 1: Esquema de los módulos del modelo

El ecosistema es modelizado empleando un sistema PDP de grado  $(10, 2)$  tomando  $T$  unidades de tiempo:

$$(G, \Gamma, \Sigma, \Pi, R_E, \{f_{r,j} : r \in R_{II}, 1 \leq j \leq 10\}, \{\mathcal{M}_{ij} : 0 \leq i \leq 1, 1 \leq j \leq 10\})$$

donde:

- El grafo del sistema es  $G = (V, S)$ , donde  $V = \{e_1, \dots, e_{10}\}$  y los arcos son  $(v, v)$  para cada  $v \in V$ , añadiendo los arcos representados en la figura:



- $\Gamma = \{X_1, G\} \cup \{X_i, X'_i, Y_i : 2 \leq i \leq 7\} \cup \{R_i : 0 \leq i \leq 6\}$ .

El símbolo  $X_1$  está asociado con 1 hectárea (He) de tierra. El símbolo  $G$  representa 1 Kg de hierba. Los símbolos  $X_i, X'_i$  y  $Y_i$  representan animales de la especie  $i$ , para  $2 \leq i \leq 7$ .  $R_i$  con  $0 \leq i \leq 6$  es un contador auxiliar de sincronización.

- $\Sigma = \{X_i, X'_i : 2 \leq i \leq 7\}$ .

- $\Pi = (\Gamma, [[ ]_1]_0, R_\Pi)$ .
- $R_E$  es el siguiente conjunto de reglas:

$$\{(X_i)_{e_k} \xrightarrow{p_{k,s,i}} (X'_i)_{e_s} : 1 \leq k, s \leq 10, 2 \leq i \leq 7\}$$

- $\{f_{r,j} : r \in R_\Pi, 1 \leq j \leq 10\}$  es un conjunto de funciones constantes cuyo rango está contenido en  $[0, 1]$ .
- $\{\mathcal{M}_{0j}, \mathcal{M}_{1j} : 1 \leq j \leq 10\}$  son cadenas sobre  $\Gamma$ , describiendo los multi-conjuntos de objetos situados inicialmente en las dos membranas de  $\mu$  para cada  $\Pi_j : 1 \leq i \leq 10$ .
  - $\mathcal{M}_{0j} = \{R_0, X_1^{q_{1,j}}\}$ , para  $1 \leq j \leq 10$ .
  - $\mathcal{M}_{1j} = \{X_2^{q_{2,j}}, \dots, X_7^{q_{7,j}}\}$ , para  $1 \leq j \leq 10$ .
- $R = R_E \cup \bigcup_{j=1}^m R_{\Pi_j}$  es un conjunto compuesto por las siguientes reglas:

- **Módulo 1:**

(Producción de hierba y reproducción de las especies)

- Producción de hierba.

$$r_{1,j} \equiv X_1 [ ]_1^0 \xrightarrow{m_j} [X_1, G^{h_j}]_1^+, 1 \leq j \leq 3$$

Las condiciones ambientales afectan a la cantidad de hierba producida por unidad de superficie (He). Por razones de simplicidad solamente se han considerado tres escenarios: producción baja, normal y alta (con probabilidades  $m_1, m_2$  y  $m_3$ , respectivamente).

- Hembras que se reproducen y generan  $d_i$  descendientes.

$$r_{2,i} \equiv [X_i]_1^0 \xrightarrow{k_{i,1} \cdot 0,5} [X_i^{1+d_i}]_1^+, 2 \leq i \leq 7$$

- Hembras y machos que no se reproducen.

$$r_{3,i} \equiv [X_i]_1^0 \xrightarrow{1-k_{i,1} \cdot 0,5} [X_i]_1^+, 2 \leq i \leq 7$$

- Sincronización del sistema P.

$$r_4 \equiv R_0 [ ]_1^0 \rightarrow [R_0]_1^+$$

Se asume un ratio de hembras y machos de 1:1. Además, se considera que todas las hembras tienen capacidad reproductiva. La primera de estas dos asunciones se cumple por la mayoría de las especies animales, pero no así la segunda, dependiendo la fertilidad de la edad de los individuos.

- **Módulo 2:**

(Alimentación y mortalidad)

- Animales que se alimentan y sobreviven.

$$r_5 \equiv [X_i G^{f_i}]_1^+ \xrightarrow{1-k_{i,2}} [Y_i]_1^-, 2 \leq i \leq 6$$

$$r_6 \equiv [X_7 X_i^{f_7}]_1^+ \xrightarrow{1-k_{7,2}} [Y_7]_1^-, 2 \leq i \leq 6$$

- o Animales que se alimentan y no sobreviven.

$$r_7 \equiv [X_i G^{f_i}]_1^+ \xrightarrow{k_{i,2}} [ ]_1^-, 2 \leq i \leq 6$$

$$r_8 \equiv [X_7 X_i^{f_7}]_1^+ \xrightarrow{k_{7,2}} [ ]_1^-, 2 \leq i \leq 6$$

- o Sincronización del sistema P.

$$r_9 \equiv [R_0]_1^+ \rightarrow [R_0]_1^-$$

Los procesos de alimentación y mortalidad son considerados en el mismo paso. Cuando un animal muere, ya ha ingerido la cantidad necesaria de alimento necesaria para sobrevivir durante un año. Esta asunción es claramente poco realista, ya que los animales pueden morir en cualquier momento a lo largo del año. Obsérvese que en este módulo solamente se ha tenido en cuenta la mortalidad debida a causas naturales.

- **Módulo 3:**

(Movimiento de los animales de las distintas especies entre entornos)

- o Los objetos relacionados con especies que no han comido van a la membrana piel.

$$r_{10,i} \equiv [X_i]_1^- \longrightarrow X_i [ ]_1^-, 2 \leq i \leq 7$$

- o Los objetos de la membrana piel van al entorno.

$$r_{11,i} \equiv [X_i]_0^0 \longrightarrow X_i [ ]_0^0, 2 \leq i \leq 7$$

- o Movimiento de objetos entre entornos.

$$r_{12,k,s,i} \equiv (X_i)_{e_k} \xrightarrow{p_{k,s,i}} (X'_i)_{e_s}, \\ 1 \leq k \leq 10, 1 \leq s \leq 10, 2 \leq i \leq 7$$

- o Los objetos  $X'_i$  entran en la membrana piel.

$$r_{13,i} \equiv X'_i [ ]_0^0 \longrightarrow [X'_i]_0^0, 2 \leq i \leq 7$$

- o Los objetos  $X'_i$  entran a la membrana interna.

$$r_{14,i} \equiv X'_i [ ]_1^- \longrightarrow [X'_i]_1^-, 2 \leq i \leq 7$$

- o Sincronización del sistema P.

$$r_{15,l} \equiv [R_l]_1^- \longrightarrow [R_{l+1}]_1^-, 0 \leq l \leq 4$$

Si hay falta de alimento, es posible que los animales migren a un área vecina buscando recursos.

- **Módulo 4:**

(Alimentación)

Si no hay recursos en el nuevo área, los animales que acaban de llegar tienen la opción de alimentarse y sobrevivir.

$$r_{16} \equiv [X'_i G^{f_i}]_1^- \xrightarrow{1-k_{i,2}} [Y_i]_1^0, 2 \leq i \leq 6$$

$$\begin{aligned}
r_{17} &\equiv [X'_7 X'_i f_7]_1^- \xrightarrow{1-k_{7,2}} [Y_7]_1^0, 2 \leq i \leq 6 \\
r_{18} &\equiv [X'_7 Y_i f_7]_1^- \xrightarrow{1-k_{7,2}} [Y_7]_1^0, 2 \leq i \leq 6 \\
r_{19} &\equiv [X'_i G f_i]_1^- \xrightarrow{k_{i,2}} [ ]_1^0, 2 \leq i \leq 6 \\
r_{20} &\equiv [X'_7 X'_i f_7]_1^- \xrightarrow{k_{7,2}} [ ]_1^0, 2 \leq i \leq 6 \\
r_{21} &\equiv [X'_7 Y_i f_7]_1^- \xrightarrow{k_{7,2}} [ ]_1^0, 2 \leq i \leq 6 \\
r_{22} &\equiv [R_5]_1^- \longrightarrow [R_6]_1^0
\end{aligned}$$

En esta fase del ciclo, los objetos que representan animales que la especie 7 puede depredar pueden encontrarse en el sistema P como  $X_i$  o  $Y_i$ , para  $2 \leq i \leq 6$ .

• **Módulo 5:**

(Reinicialización del ciclo)

$$\begin{aligned}
r_{23,i} &\equiv [Y_i]_1^0 \longrightarrow [X_i]_1^0, 2 \leq i \leq 7 \\
r_{24} &\equiv [R_6]_1^0 \longrightarrow [R_0]_1^0 \\
r_{25} &\equiv [X_1]_1^0 \longrightarrow X_1 [ ]_1^0 \\
r_{26,i} &\equiv [X'_i]_1^0 \longrightarrow [ ]_1^0, 2 \leq i \leq 7 \\
r_{27} &\equiv [G]_1^0 \longrightarrow [ ]_1^0
\end{aligned}$$

A continuación se muestra la lista de parámetros iniciales necesarios:

■ Parámetros relacionados con animales:

- $k_{i,1}$ , ( $2 \leq i \leq 7$ ): Ratio de fertilidad para la especie  $i$  (hembras).
- $k_{i,2}$ , ( $2 \leq i \leq 7$ ): Ratio de supervivencia para la especie  $i$ .
- $f_i$ , ( $2 \leq i \leq 7$ ): Cantidad de unidades de alimento consumidas por animal para la especie  $i$ .
- $d_i$ , ( $2 \leq i \leq 7$ ): Número de descendientes por hembra y ejercicio para la especie  $i$ .
- $q_{i,j}$ , ( $2 \leq i \leq 7, 1 \leq j \leq 10$ ): Número inicial de individuos de la especie  $i$  en el área  $j$ .
- $p_{k,s,i}$ , ( $2 \leq i \leq 7, 1 \leq k, s \leq 10$ ): Probabilidad de movimiento de la especie  $i$  del área  $k$  a  $s$ .

■ Parámetros relacionados con la hierba:

- $h_i$ ,  $1 \leq i \leq 3$ : Cantidad de hierba producida por hectárea bajo diferentes condiciones ( $i = 1$  condiciones desfavorables,  $i = 2$  condiciones normales,  $i = 3$  condiciones favorables).
- $m_i$ ,  $1 \leq i \leq 3$ : Probabilidad de que se den las correspondientes condiciones afectando a la producción de hierba.
- $q_{1,j}$ , ( $1 \leq j \leq 10$ ): Número de hectáreas de tierra del área  $j$ .

### 1.0.3. Definición en P-Lingua

A continuación se va a mostrar el código del programa escrito en P-Lingua que especifica la familia de sistemas P antes descrita. Los parámetros descritos anteriormente forman parte del fichero, y como veremos en el apartado de simulación recibirán distintos valores dependiendo de los valores introducidos por el usuario en la ventana de la aplicación basada en MeCoSim.

El código es el siguiente:

```
@model<probabilistic>

def tritrophic_ampliat()
/*Tritrophic. */
{
    /* Membrane structure */
    /* LA ESTRUCTURA GENERAL ES [[1]e1 [[1]e2,.....[[1]ek]*

@mu= [ [[[]'1]0]'101,101
[[[]'1]0]'102,102
[[[]'1]0]'103,103
[[[]'1]0]'104,104
[[[]'1]0]'105,105
[[[]'1]0]'106,106
[[[]'1]0]'107,107
[[[]'1]0]'108,108
[[[]'1]0]'109,109
[[[]'1]0]'110,110 ]'p;

    /* Initial multisets */

@ms(1,{j+100}) += X{2} * q{2,j}, X{3}*q{3,j}, X{4}*q{4,j}, X{5}*q{5,j},X{6}*q{6,j},X{7}*q{7,j} : 1<=j<=10;
@ms(0,{j+100}) += X{1}*q{1,j}:1<=j<=10;
@ms(0,{j+100}) += R{0}:1<=j<=10;

    /******
    /*Module 1 */
    /******

/*r1*/ X{1}[]'1-->+X{1},G*(h{j})]'1: m{j}: 1<=j<=3;
/*r2*/ [X{i}]'1-->+X{i}*(1+d{i})]'1: (k{i,1}*0.5): 2<=i<=7;
/*r3*/ [X{i}]'1-->+X{i}]'1: 1-(k{i,1}*0.5): 2<=i<=7;
/*r4*/ R{0}[]'1-->+R{0}]'1:1;

    /******
    /* Module 2 */
    /******

/*r5*/ +X{i}, G*(f{i})]'1 --> -[Y{i}]'1 :: (1-k{i,2}): 2<=i<=6;
/*r6*/ +X{7}, X{i}*(f{7})]'1 --> -[Y{7}]'1 :: (1-k{7,2}):2<=i<=6;
/*r7*/ +X{i}, G*(f{i})]'1 --> -[#]'1 :: k{i,2}:2<=i<=6;
/*r8*/ +X{7}, X{i}*(f{7})]'1 --> -[#]'1 :: k{7,2}:2<=i<=6;
/*r9*/ +R{0}]'1-->-R{0}]'1:1;

    /******
    /* Module 3 */
    /******

/*r10*/ -[X{i}]'1 --> X{i} -[]'1 ::1 : 2<=i<=7;
/*r11*/ [X{i}]'0 --> X{i} []'0 ::1 : 2<=i<=7;
/*rel2*/ [[X{i}]'(k+100) [#]'(x+100)]'p --> [[#]'(k+100) [Xp{i}]'(x+100)]'p ::p(k,r,i): 1<=k<=10, 1<=r<=10, 2<=i<=7,k<=r;
/*rel2*/ [[X{i}]'(k+100)]'p --> [[Xp{i}]'(k+100)]'p ::p(k,k,i)+0.00001 :1<=k<=10, 2<=i<=7;
/*r13*/ Xp{i}[]'0-->[Xp{i}]'0:1:2<=i<=7;
/*r14*/ Xp{i} -[]'1 --> -[Xp{i}]'1:1: 2<=i<=7;
/*r15*/ -[R{i}]'1--> -[R{i+1}]'1:1:0<=i<=4;
```

```

/*****
/* Module 4 */
/*****
/*r16*/ -[Xp{i}, G*(f{i})]'1 --> [Y{i}]'1 :: (1-k{i,2}):2<=i<=6;

/*r17*/ -[Xp{7}, Xp{i}*(f{7})]'1 --> [Y{7}]'1 :: (1-k{7,2}):2<=i<=6;
/*r18*/ -[Xp{7}, Y{i}*(f{7})]'1 --> [Y{7}]'1 :: (1-k{7,2}):2<=i<=6;
/*r19*/ -[Xp{i}, G*(f{i})]'1 --> [#]'1 :: k{i,2}:2<=i<=6;
/*r20*/ -[Xp{7}, Xp{i}*(f{7})]'1 --> [#]'1 :: k{7,2}:2<=i<=6;
/*r21*/ -[Xp{7}, Y{i}*(f{7})]'1 --> [#]'1 :: k{7,2}:2<=i<=6;
/*r22*/ -[R{5}]'1-->[R{6}]'1::1;

/*****
/* Module 5 */
/*****
/*r23*/ [Y{i}]'1-->[X{i}]'1::1:2<=i<=7;

/*r24*/ [R{6}]'1-->R{0}[#]'1::1;
/*r25*/ [X{1}]'1-->X{1}[#]'1::1;
/*r26*/ [Xp{i}]'1-->[#]'1::1:2<=i<=7;
/*r27*/ [G]'1-->[#]'1::1;
}
def main()
{
    call tritrophic_ampliat();
}

```

### 1.3 Simulación

El sistema P descrito en el apartado anterior ilustra un ecosistema con distintos niveles.

Para llevar a cabo la simulación planteada en este ejemplo debemos hacer lo siguiente:

1. Es necesario definir una aplicación basada en MeCoSim adaptada a este problema (en este caso interacciones tritróficas). Recordemos que la personalización se lleva a cabo empleando un fichero de configuración Excel con las entradas y salidas (tabuladas y gráficas) deseadas y los parámetros para P-Lingua. El archivo Excel correspondiente está disponible como en la unidad anterior en el repositorio de apps

["http://www.p-lingua.org/mecosim/matvida/apps/"](http://www.p-lingua.org/mecosim/matvida/apps/),

de modo que seleccione este repositorio y seleccione la aplicación "Ejemplo 5 - Interacciones tritróficas" para instalarse con doble click y salimos del repositorio de apps. Añadimos la aplicación "Ejemplo5.Tritrophic.xls". Si abre este fichero en MS Excel u Open Office/Libre Office podrá ver que dispone de información sobre la estructura jerárquica de las pestañas que aparecen en la aplicación, los parámetros a generar, la obtención de las salidas de la computación o la visualización de gráficos.

2. Acudir al gestor de repositorios para importar el archivo de modelo ("Ejemplo5.Tritrophic.pli") y el de escenario ("Ejemplo5.Tritrophic.ec2"), para poder

cargar el sistema P correspondiente expresado en lenguaje P-Lingua y aplicarlo sobre el grafo indicado.

3. Lance la aplicación Tritrophic mediante "Run". Observe la apariencia de la aplicación adaptada para el problema de interacciones tritróficas, y contrástelo con la descripción de los parámetros proporcionada anteriormente y contenida en el fichero .pli. El paso de la ventana al .pli está basado en la pestaña "Simulation Params" del archivo .xls de configuración de la aplicación.
4. Cargue el modelo mediante *Model > Set Model*, abra el escenario descargado mediante *Scenario > Open*, y seleccione en el menú de simulación el algoritmo "dndp2".
5. Depure el modelo desde la pestaña de Debug console, para comprobar que se inicializa correctamente, se generan los parámetros correspondientes y la especificación proporcionada no contiene errores. Puede repasar de nuevo el proceso conforme a lo indicado en los pasos anteriores. Analice un ciclo de computación (recuerde que se compone de 9 pasos), ejecutando 9 "steps" y viendo analizando la configuración tras cada paso (recuerde que tras cada paso de computación puede visualizar esta información en modo texto en "Simulation info" o bien de forma visual mediante los plugins de visualización").
6. Si el proceso anterior ha sido satisfactorio, puede salir de la aplicación Tritrophic con la seguridad de que su modelo está funcionando correctamente. Ahora pensemos que somos un gestor del ecosistema, pertenecientes a algún organismo encargado de controlar la especie. Entremos de nuevo en la aplicación, lancemos la simulación para los datos cargados y observemos los resultados en las subpestañas de "Simulation". Veremos que "Results" nos ofrece la vista detallada de los objetos de cada membrana, pero como gestores del ecosistema no tenemos la menor idea de qué es un sistema P y, por tanto, tampoco tenemos idea de membranas y objetos. Sí será de interés la pestaña Graphics, ya que nos mostrará una lista de elementos Zona - Espacie, y para cada uno nos proporcionará la evolución de los individuos de la especie en esa zona en forma de gráfico de líneas, de forma similar a la figura 2.

Ya que se trata de un sistema probabilístico, es conveniente cambiar el número de simulaciones por ciclo en el menú "Simulate", de manera que el dato obtenido se corresponda con la media de una serie de simulaciones y no con una simulación individual. Repita la simulación para 50 simulaciones por ciclo y observe el resultado.

7. Repita el proceso anterior modificando parámetros relevantes en las pestañas de entrada.

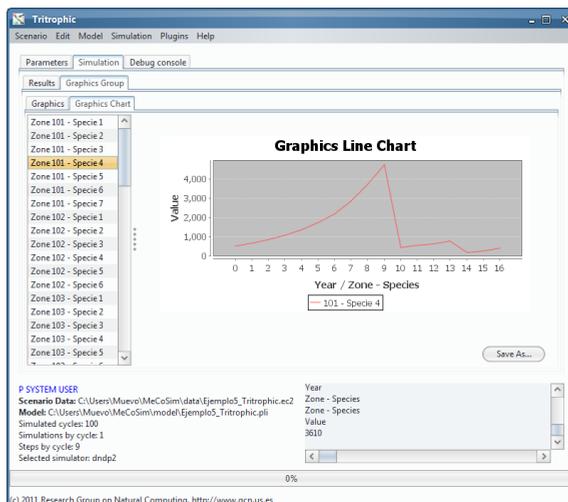


FIGURA 2: Tritrophic - Evolución de una especie en una zona.

Hasta aquí el detalle de la simulación. En el próximo apartado se detallan los resultados esperados y el análisis de los mismos.

## 1.4 Resultados

Se deberá recopilar información de las simulaciones realizadas en el apartado anterior. Observe que los gráficos obtenidos pueden ser almacenados como archivos gráficos. Deberá recopilar adicionalmente información sobre las trazas realizadas paso a paso. Obtenga los resultados gráficos de las especies en una zona que elija, bajo 5 escenarios diferentes en cuanto a los datos de entrada introducidos, de manera que pueda contrastar la diferencia en las gráficas obtenidas conforme varían los parámetros de cada escenario.

## 2 Ejemplo 6. Quebrantahuesos

El mismo marco descrito y empleado en el ejemplo 5 se empleó con éxito para resolver el problema real del quebrantahuesos en el Pirineo catalán (ver [2]).

### 3 Ejemplo 7. Mejillón cebra

El mismo marco descrito y empleado en el ejemplo 5 se empleó con éxito para resolver el problema real del mejillón cebra en el pantano de Ribarroja (ver [3]).

### 4 Ejemplo 8. Pandemic

Los ecosistemas no son el único área de interés dentro de lo que conocemos como dinámica de poblaciones. Por ejemplo, el sistema formado por los individuos de una localidad puede ser considerado un ecosistema, y puede ser interesante estudiar un caso de emergencia médica como podría ser la extensión de una pandemia entre los habitantes de la misma, así como las posibles formas de gestionar la misma. Como en el caso de las interacciones tritróficas, se considera un modelo simplificado en el documento “Unidad7aux.Pandemic.pdf” disponible en Moodle. Además, la descripción del archivo P-Lingua correspondiente se encuentra en: [http://www.p-lingua.org/mecosim/doc/case\\_studies/pandemics.html](http://www.p-lingua.org/mecosim/doc/case_studies/pandemics.html), junto con la aplicación basada en MeCoSim y el escenario.

## BIBLIOGRAFÍA

- [1] D. Besozzi, P. Cazzaniga, D. Pescini, G. Mauri. Modelling metapopulations with stochastic membrane systems. *BioSystems*, **91**, **3**, (2008) 499-514.
- [2] M. Cardona, M.A. Colomer, M.J. Pérez-Jiménez, D. Sanuy, A. Margalida. Modeling ecosystems using P systems: The bearded vulture, a case study. *Lecture Notes in Computer Science*, 5391 (2009), 137-156.
- [3] M. Cardona, M.A. Colomer, A. Margalida, A. Palau, I. Pérez-Hurtado, M.J. Pérez-Jiménez, D. Sanuy. A computational modeling for real ecosystems based on P systems. *Natural Computing*, **10**, **1** (2011), 39-53.
- [4] Gh. Păun, G. Rozenberg, A. Salomaa (eds.) *The Oxford Handbook of Membrane Computing*, Oxford University Press, 2010.
- [5] M.J. Pérez-Jiménez, F.J. Romero. P systems, a new computational modelling tool for Systems Biology. Transactions on Computational Systems Biology VI. *Lecture Notes in Bioinformatics*, **4220**, (2006) 176-197.
- [6] F.J. Romero, M.J. Pérez-Jiménez. Modelling gene expression control using P systems: The Lac Operon, a case study. *BioSystems*, **91**, **3**, (2008) 438-457.

- [7] M. Cardona, M. A. Colomer, I. Pérez-Hurtado, M.J. Pérez-Jiménez, D. Sanuy, A. Margalida. A P system Based Model of an Ecosystem of Some Scavenger Birds. *10th International Workshop, WMC 2009. Lecture Notes in Computer Science*, 5957 (2010), 182–195.
- [8] M.A. Colomer, I. Pérez-Hurtado, M.J. Pérez-Jiménez, A. Riscos. Comparing simulation algorithms for multienvironment probabilistic P system over a standard virtual ecosystem. *Natural Computing*, 11 (2012), 369-379.
- [9] M.A. Colomer, I. Pérez-Hurtado, M.J. Pérez-Jiménez, A. Riscos. Simulating Tritrophic Interactions by Means of P Systems. In A.K. Nagar, R. Thamburaj, K. Li, Z. Tang, R. Li (eds.) *Proceedings 2010 IEEE Fifth International Conference on Bio-inspired Computing: Theories and Applications (BIC-TA 2010)*, IEEE Press, Volume 2, September 8-10, 2010, Liverpool, UK, ISBN 978-1-4244-6438-8, pp. 1621-1628.
- [10] I. Pérez-Hurtado, L. Valencia, M.J. Pérez-Jiménez, M.A. Colomer, A. Riscos. MeCoSim: A general purpose software tool for simulating biological phenomena by means of P Systems. In K. Li, Z. Tang, R. Li, A.K. Nagar, R. Thamburaj (eds.) *Proceedings 2010 IEEE Fifth International Conference on Bio-inspired Computing: Theories and Applications (BIC-TA 2010)*, IEEE Press, Volume 1, September 23-26, 2010, Changsha, China, ISBN 978-1-4244-6439-5, pp. 637-643.
- [11] M.A. Colomer, A. Margalida, M.J. Pérez-Jiménez. Population Dynamics P System (PDP) Models: A Standardized Protocol for Describing and Applying Novel Bio-Inspired Computing Tools. *PLOS ONE*, Volume 8, 4 (2013), 1-13 (doi: 10.1371/journal.pone.0060698).
- [12] M. García-Quismondo, R. Gutiérrez-Escudero, I. Pérez-Hurtado, M.J. Pérez-Jiménez, A. Riscos-Núñez *Lecture Notes in Computer Science*, 01/2010, Volume 5957, p.264-288, (2010)