

Modelización de ecosistemas con Sistemas P

Grupo de Investigación en Computación Natural
Universidad de Sevilla

1. Introducción

La *Computación Natural* es una disciplina cuyo objetivo principal es el estudio y simulación de los procesos dinámicos que se dan en la Naturaleza y que son susceptibles de ser interpretados como procedimientos de cálculo. En ella, se investigan modelos y técnicas computacionales inspiradas en la Naturaleza con el objetivo de entender más y mejor el mundo que nos rodea, en términos de procesamiento de la información. En este marco se encuentra la *Computación Celular con Membranas*, *Membrane Computing*, que es un modelo de computación paralelo de tipo distribuido y no determinista inspirado en la estructura y el funcionamiento de las células de los organismos vivos.

Los dispositivos computacionales en Computación Celular con Membranas se denominan *Sistemas de Membranas* o *Sistemas P* y constan de unos ingredientes sintácticos tales como una estructura de membranas (o compartimentos) en donde se localizan multiconjuntos de objetos (abstracciones de las sustancias químicas) que evolucionan de acuerdo a una serie de reglas de reescritura (abstracciones de las reacciones químicas). Por otra parte, existen unos ingredientes semánticos (no determinismo, paralelismo maximal y masivo) que representan la forma en la cual las reglas son aplicadas a lo largo de una computación.

Entre las diversas aplicaciones de la Computación Celular con Membranas se encuentra la modelización y simulación de procesos de la vida real, especialmente fenómenos biológicos, desde procesos celulares a nivel microscópico hasta dinámicas de poblaciones. Precisamente, las características de los sistemas P los hacen especialmente apropiados para la modelización de estos fenómenos en donde habitualmente existe una gran interacción de individuos o sustancias en una estructura jerarquizada de compartimentos o zonas.

En esta práctica se realizará la modelización y simulación de un ecosistema teórico con tres niveles tróficos. Para realizar la simulación por ordenador es necesario disponer de aplicaciones informáticas que permitan especificar y reproducir el comportamiento de los sistemas P, con este fin, utilizaremos P-Lingua que es un lenguaje de programación desarrollado por el Grupo de Investigación en Computación Natural de la Universidad de Sevilla (GCN). Los programas escritos en P-Lingua definen sistemas P de manera sencilla, paramétrica y modular. Gracias a ello, se proporciona una entrada estándar a las aplicaciones de software para Computación Celular con Membranas. Por otra parte, la simulación de los sistemas P será llevada a cabo por la biblioteca (GNU GPL) *pLinguaCore* que implementa tres funcionalidades principales:

- Lectura y análisis de ficheros que definen sistemas P.
- Simulación de los sistemas P definidos.
- Exportación de sistemas P a ficheros de salida.

Esta biblioteca se puede incluir con facilidad en terceras aplicaciones, proporcionando la capacidad de simular sistemas P, así como procesar ficheros en formato P-Lingua y otros formatos soportados. En la página web www.p-lingua.org se puede encontrar más información.

2. Archivos necesarios

En esta práctica se utilizarán los siguientes archivos:

- **tritrophic-interactions.jar**: Ejecutable de Java que incluye la biblioteca pLinguaCore y un interfaz gráfico para depurar el sistema P que modeliza el ecosistema y probar diversos escenarios iniciales.
- **tritrophic-model.pli**: Fichero de P-Lingua que especifica la familia de sistemas P que definen el ecosistema caso de estudio.
- **caso3.ec2**: Fichero binario para ser cargado por el interfaz gráfico, con algunos datos iniciales.
- **modelización-ecosistemas.pdf**: Documento que describe el modelo del ecosistema caso de estudio.
- **herramientas-simulación.pdf**: Documento que introduce el lenguaje P-Lingua y sus herramientas asociadas.

Todos estos archivos se pueden descargar provisionalmente de <http://boole.cs.us.es/ignacio>

3. Herramientas de simulación utilizadas

El programa **tritrophic-interactions.jar** es similar al utilizado para la gestión de ecosistemas reales mediante modelos computacionales basados en sistemas P (como es el caso del ecosistema del quebrantahuesos o el ecosistema del mejillón zebra). Todos estos programas comparten el mismo motor de simulación (pLinguaCore) y se diferencian en el modelo que define el ecosistema (escrito en un fichero P-Lingua) así como en la manera de introducir los datos de entrada e interpretar los datos de salida (la interfaz gráfica de usuario).

Todas estas aplicaciones proporcionan dos modos de funcionamiento, cada uno de ellos está dirigido a una categoría o tipo de usuario: *el ecólogo* y *el diseñador*. La herramienta permite diferentes tipos de acciones (casos de uso) para cada tipo de usuario.

Por una parte, el usuario ecólogo es el usuario final de la aplicación y, por tanto, no necesita tener ningún conocimiento acerca del paradigma *Membrane Computing*, ni siquiera acerca del modelo. Por ello, el programa se comporta como una especie de *caja negra*. El objetivo del usuario ecólogo es el desarrollo de experimentos virtuales sobre el ecosistema simulado y para este propósito el programa permite las siguientes acciones:

- Editar los parámetros iniciales del ecosistema.
- Seleccionar el número de años a simular.
- Seleccionar el número total de simulaciones a realizar.
- Guardar y cargar todos los parámetros introducidos en ficheros binarios con extensión *ec2*.
- Ejecutar simulaciones del ecosistema.
- Consultar los resultados de las simulaciones, los cuales se presentan mediante tablas y gráficas.
- Guardar las gráficas y tablas como imágenes en formato *png*.

Con estas acciones, *el usuario ecólogo* puede analizar distintos escenarios del ecosistema y realizar experimentos virtuales, pudiendo salvar las distintas gráficas de resultados en ficheros. De este modo, es posible estudiar la evolución de la población de cada una de las especies durante los años simulados bajo unas condiciones iniciales determinadas; así como la variación de otros importantes valores, tales como la biomasa producida por especie y por año.

De manera transparente para el usuario ecólogo, el programa instancia en cada simulación un sistema P que codifica el comportamiento del ecosistema con los parámetros iniciales introducidos.

Por cada año simulado, se realizarán un determinado número de simulaciones fijado a priori, lo cual permite amortiguar el ruido inherente a los procesos estocásticos. Los resultados se expresarán a través de los valores medios y las desviaciones típicas de los valores estudiados.

El *usuario diseñador* es el responsable de especificar, depurar y validar la familia de sistemas P que usa el programa. La validación experimental se realiza mediante la comparación de los resultados obtenidos en las simulaciones con los valores reales observados y obtenidos experimentalmente en el ecosistema. Cada año de simulación corresponde a un determinado número de pasos de computación en el sistema P que se fija a priori por el usuario diseñador.

La especificación de la familia de sistemas P que modeliza el ecosistema objeto de estudio se describe por el usuario diseñador en un fichero en formato P-Lingua con extensión *pli*. Posteriormente, este usuario indica a la aplicación informática la ruta del fichero P-Lingua.

El programa ofrece al usuario diseñador las mismas acciones que al ecólogo, permitiendo interactuar con el proceso de simulación y facilitando así el proceso de depuración del modelo. Con este fin, se han incluido algunas acciones adicionales:

- Selección del fichero P-Lingua asociado a la aplicación.
- Compilación de ficheros P-Lingua.
- Simulación paso a paso.
- Selección del número de pasos de computación que corresponden a un año en el ecosistema modelizado.

4. Ejercicio práctico

Para realizar este ejercicio es necesario disponer de un ordenador con Linux o Windows y Java instalado, con versión 1.6.0 o posterior. Para comprobar qué versión de Java está instalada, se puede ejecutar el comando `java -version` en una consola del sistema. Si la versión es anterior a la 1.6.0, o si Java no está instalado, se puede descargar de www.java.com

1. Analizar el sistema P descrito en el documento `modelización-ecosistemas.pdf` (páginas 15-24). Entender que se trata del modelo computacional de un ecosistema teórico con tres niveles tróficos.
2. Analizar el fichero `tritrophic-model.pli`, comprobar que es un fichero de P-Lingua que especifica el modelo descrito en `modelización-ecosistemas.pdf`. Téngase en cuenta que los identificadores de los entornos $e_1 \dots e_{10}$ son 101...110 en el fichero de P-Lingua.
3. Ejecutar el programa `tritrophic-interactions.jar`:
 - `java -jar tritrophic-interactions.jar`
4. Cargar el fichero `caso3.ec2` que contiene los datos de un escenario inicial de ejemplo. Para ello acceder al menú: **Ecosystem** → **Open**, o simplemente pulsar **Ctrl-O**.
5. Cargar el modelo `tritrophic-model.pli`. Para ello acceder al menú **Model** → **Set Model**, o simplemente pulsar **Ctrl-M**. la apariencia de la interfaz gráfica debería ser similar a la mostrada en la figura 1. En la pestaña **Parameters** se pueden editar los parámetros del ecosistema para definir un escenario inicial, existen cuatro sub-pestañas:
 - **Animals**: Parámetros asociados a cada especie i , concretamente:
 - f_i : Cantidad de comida que necesita la especie i .
 - $k_{i,1}$: Probabilidad de que se reproduzca una hembra de la especie i .
 - $k_{i,2}$: Probabilidad de que se alimente un animal de la especie i .
 - d_i : Número de descendientes por hembra fértil de la especie i .
 - **Grass**: Parámetros asociados a la producción de hierba:
 - $h_1 \dots h_3$: Producción anual de hierba para tres situaciones climatológicas diferentes.
 - $m_1 \dots m_3$: Probabilidad de que se produzca cada una de las situaciones anteriores.
 - **Populations**: Poblaciones de cada especie en cada una de las 10 zonas, incluyendo la cantidad de hectáreas de hierba por cada zona.
 - **Movements**: Conjunto de probabilidades del tipo $p_{j,k,i}$ que indican la probabilidad de que un animal de la especie i se mueva del entorno j al k .
6. Acceder a la pestaña **Debug console**, esta pestaña ofrece al diseñador un interfaz para la depuración del modelo escrito en el fichero P-Lingua. Pulsar el botón **Init model**, en ese momento, el programa instancia un sistema P concreto utilizando el escenario inicial cargado desde el fichero `caso3.ec2`. La interfaz debería mostrar en la pestaña **Information** la descripción del sistema P generado, tal como se muestra en la figura 2. Comprobar también las pestañas **Errors** y **Warnings**.

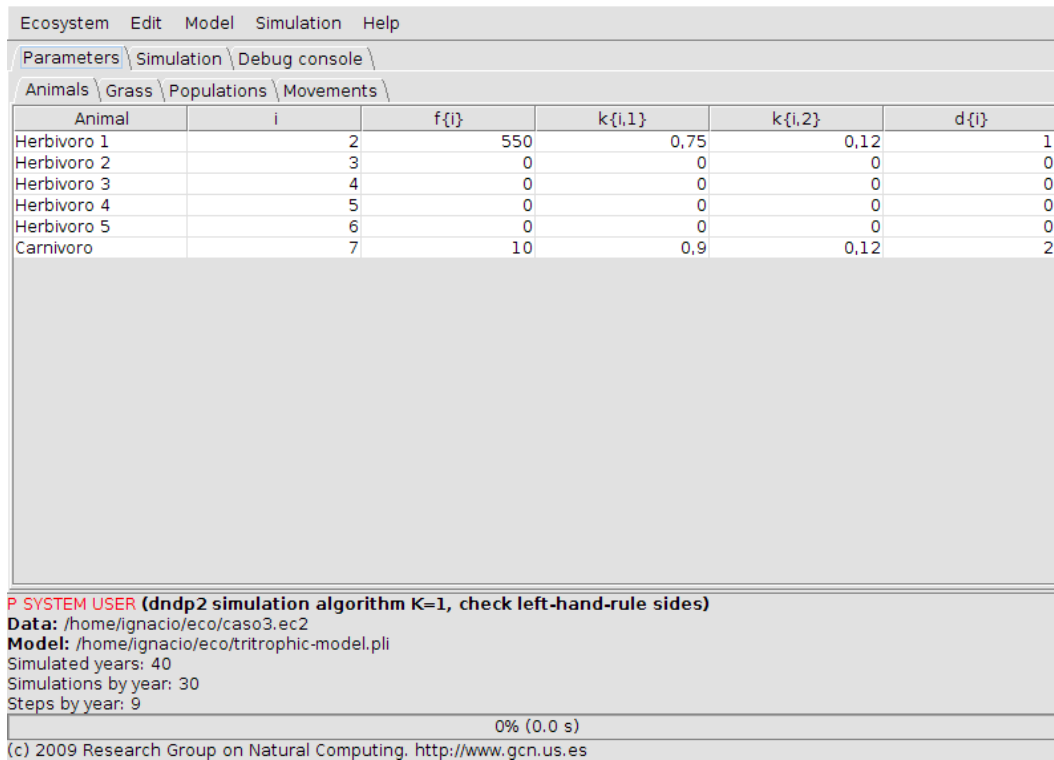


Figura 1: La interfaz gráfica de usuario de *tritrophic-interactions.jar*

7. Editar el fichero `tritrophic-model.pli` con cualquier editor de textos, introducir algún error y salvar. Pulsar el botón **Init model**, comprobar que los errores aparecen en la ventana de **Errors**. Deshacer los cambios en el fichero `tritrophic-model.pli` y volver a pulsar el botón **Init model** para cargar el modelo sin errores.
8. Según el marco de modelización, las probabilidades de las reglas con la misma parte izquierda deben de sumar 1. Comprobar en la pestaña **movements** que $\forall i, j : 2 \leq i \leq 7, 1 \leq j \leq 10$ se cumple que $\sum_{k=1}^{10} p_{j,k,i} = 1$, editar la celda correspondiente a $p_{1,1,2}$ e introducir un valor de 0,2, de tal manera que el conjunto de probabilidades ya no sume 1. Volver a la pestaña **Debug console** y pulsar el botón **Init model**, comprobar en la sub-pestaña **errors** el mensaje de error generado al no tener probabilidades que sumen 1. Una vez comprobado, volver a restaurar la situación anterior, editando la probabilidad $p_{1,1,2}$ e inicializando de nuevo el modelo. Cargar de nuevo la aplicación si hace falta.
9. Una vez que el modelo se carga sin fallos, en la pestaña **Debug console** pulsar el botón **step** para simular un paso de computación del sistema P, comprobar en la sub-pestaña **Information** las reglas aplicadas y los multiconjuntos resultantes, tal como se muestra en la figura 3. Examinar los resultados, comprobar que es un paso de computación válido y que las reglas se han aplicado maximalmente de acuerdo a sus probabilidades asociadas. Pulsar el botón **Reset** para reiniciar la simulación y volver a pulsar el botón **Step** para repetir el primer paso de computación. Comprobar que se ha dado otra computación válida, aunque diferente.
10. En este modelo, un ciclo completo en el ecosistema (un año) se modeliza en 9 pasos de computación, pulsar el botón **reset** y luego **Run steps** para simular varios pasos de computación, aparecerá una ventana preguntando el número de pasos a simular, introducir el valor de 9 y pulsar **aceptar**. Analizar la computación generada, comprobando cada una de las configuraciones intermedias y las reglas que se han aplicado para pasar a la siguiente configuración. Identificar la ejecución de las reglas correspondientes a cada uno de los módulos del modelo (representados en la página 17 del documento *modelización-ecosistemas.pdf*). ¿Qué mecanismos se han utilizado para realizar la sincronización de los módulos? Es decir, para que los módulos se ejecuten en orden. Comprobar que tras los 9 pasos de computación, el sistema P está preparado para modelizar un nuevo ciclo y los objetos $X_i : 1 \leq i \leq 7$ que representan cantidades de hierba y animales han sido actualizados.
11. La aplicación también ofrece la posibilidad de realizar experimentos virtuales, que consisten en la simulación de un número determinado de años para obtener la evolución de la población de cada una

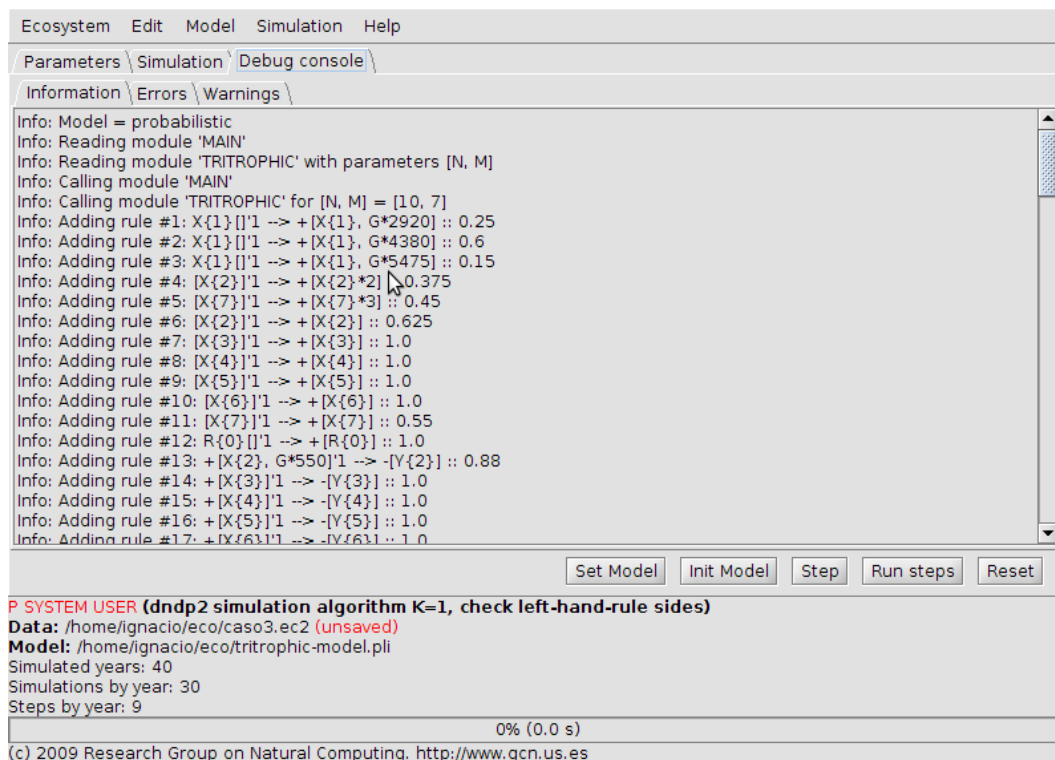


Figura 2: Inicialización del modelo para un escenario concreto

de las especies y de la cantidad de hierba. Cada simulación se repite un número de veces y se calculan valores medios y desviaciones típicas. Por defecto, el fichero `Caso3.ec2` está configurado para simular 40 y realizar 30 simulaciones. Pulsar **Simulation** → **Simulate!**, o simplemente **Ctrl-L** para realizar el proceso completo de simulación. Tras la simulación, examinar los resultados obtenidos en la pestaña **Simulation**. Comprobar la tabla de resultados en la sub-pestaña **Results** y los gráficos generados automáticamente en la sub-pestaña **Graphics**.

12. Es posible indicar el número de años a simular en **Simulation** → **Options** → **Number of Years**, el número de simulaciones a realizar en **Simulation** → **Options** → **Number of simulations by year** y el número de pasos de computación por ciclo en **Model** → **Set number of steps by year**, aunque este último parámetro no se debe cambiar salvo que se cambie el modelo y un ciclo en el ecosistema venga representado por un número de pasos de computación diferente a 9. Probar a realizar varios procesos de simulación, cambiando el número de años a simular y el total de simulaciones a realizar.
13. Finalmente, editar los parámetros iniciales del ecosistema para realizar diversos experimentos virtuales.

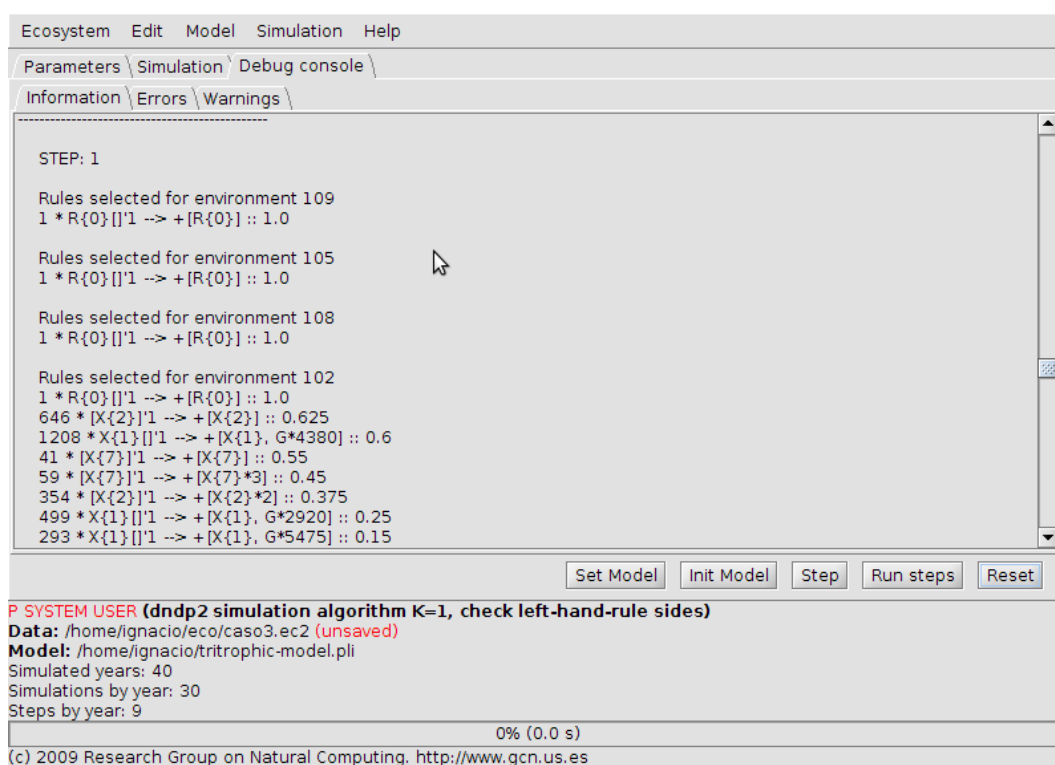


Figura 3: Simulación de un paso de computación