



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

LA MORFOGÉNESIS COMO RESULTADO
DE LA TRANSMISIÓN E INTEGRACIÓN
DE INFORMACIÓN BIOLÓGICA.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
M A T E M Á T I C O

P R E S E N T A:
LUIS DE JESÚS MARTÍNEZ LOMELI

DIRECTOR DE TESIS:
DR. PABLO PADILLA LONGORIA



2012



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno
Martínez
Lomeli
Luis de Jesús
56 16 11 60
Universidad Nacional Autónoma de México
Facultad de Ciencias
Matemáticas
407056763

2. Datos del tutor
Dr
Pablo
Padilla
Longoria

3. Datos del sinodal 1
Dr
Francisco
Hernández
Quiroz

4. Datos del sinodal 2
Dr
Pedro Arturo
Góngora
Luna

5. Datos del sinodal 3
Dr
Arturo Carlos II
Becerra
Bracho

6. Datos del sinodal 4
Dr
Rafael Ángel
Barrio
Paredes

7. Datos del trabajo escrito
La morfogénesis como resultado de la transmisión e integración de información biológica
64 p
2012

La morfogénesis como resultado de la
transmisión e integración de información
biológica.

Luis de Jesús Martínez Lomeli

31 de enero de 2012

A mis padres Elisa y José Luis y mi hermano Alfonso, por su apoyo incondicional en mis estudios.

A mi hermano José por ser mi compañero en la carrera.

Índice general

1. Introducción.	5
2. Álgebras de procesos.	9
2.1. Preliminares, concurrencia y movilidad.	9
2.2. Nociones básicas del Cálculo-II.	11
2.2.1. Autómatas.	11
2.2.2. Sistemas de Transiciones Etiquetados y la bisimilaridad.	12
2.3. El Cálculo-II.	15
2.3.1. Definición.	15
2.3.2. Transiciones del Cálculo-II.	18
2.3.3. Movilidad.	20
3. Modelación de fenómenos naturales.	22
3.1. Notación.	22
3.2. Simulaciones de reacciones químicas.	23
3.2.1. Decaimiento radiactivo.	23
3.2.2. Disociación del cloruro de sodio (NaCl).	24
3.2.3. Formación de Amidas.	26
3.3. Simulaciones en procesos biológicos.	27
3.3.1. Sistema Biestable.	27
3.3.2. Represilador.	28
3.4. Aplicaciones en compuertas lógicas.	30
3.4.1. Compuertas constitutivas.	30
3.4.1.1. Productos de los genes.	32
3.4.2. Puerta unaria: gen con control negativo.	33
3.4.3. Puerta unaria: gen con control positivo.	33
3.4.4. Simulaciones de compuertas unarias.	33
4. La morfogénesis.	35
4.1. El modelo de Turing.	36
4.2. Un sistema de reacción difusión clásico: El Activador-Inhibidor.	38
4.3. Álgebras de procesos y sistemas de reacción difusión.	40
4.4. Modelo del sistema Activador Inhibidor usando Cálculo II-Estocástico.	42

<i>ÍNDICE GENERAL</i>	3
4.4.1. Modelo de una celda aislada	42
4.4.1.1. Resultados Numéricos	44
4.4.2. Modelo de un tejido celular, $m \times n$ celdas con un sistema <i>AI</i> cada una.	46
4.4.2.1. Resultados numéricos preliminares del modelo de $m \times n$ celdas.	50
5. Conclusiones.	53
Bibliografía	55
A. Códigos de simulaciones.	58
A.0.3. Simulaciones de reacciones químicas.	58
A.0.3.1. Decaimiento radiactivo.	58
A.0.3.2. NaCl.	58
A.0.3.3. (* Na + Cl \rightleftharpoons Na+ + Cl- *).	58
A.0.3.4. Formación Amidas.	59
A.0.4. Simulaciones de procesos biológicos.	59
A.0.4.1. Sistema Biestable.	59
A.0.4.2. Represilador.	60
A.0.5. Simulaciones de compuertas lógicas.	60
A.0.5.1. Compuertas negativas.	60
B. Códigos de simulaciones del activador-inhibidor	62
B.1. Código del <i>AI</i> de una celda.	62
B.2. Código del <i>AI</i> de $m \times n$ celdas.	62

Resumen

En su trabajo original de “Las bases químicas de la morfogénesis”, A. M. Turing propuso un mecanismo basado en el efecto combinado de difusión y reacción química de dos diferentes morfógenos, con el objeto de explicar la emergencia de patrones en la naturaleza. En ese mismo trabajo él sugirió que este mismo enfoque puede ser usado para el caso de las redes de regulación genéticas. Sin embargo esta aproximación ha sido criticada debido a que los así llamados morfógenos no han podido ser identificados en diversos casos de estudio en biología. En este trabajo proponemos que el efecto de difusión y reacción química puede ser remplazado ó complementado, por un mecanismo intercelular de comunicación. Conceptualmente esto es equivalente a decir que lo que se está difundiendo es información. En otras palabras, la información juega el papel del morfógeno que le permite a la célula tomar la decisión sobre qué destino celular tomará.

En el presente trabajo hacemos uso del formalismo llamado cálculo-II estocástico, inicialmente desarrollado para representar comunicación entre agentes. Es en base al cálculo-II que logramos diseñar un modelo de red de regulación genética del tipo activador-inhibidor. Al simular esta red en un contexto de $m \times n$ células acopladas espacialmente, logramos reproducir algunos patrones tipo mancha como los descritos por Turing.

El contenido de la tesis muestra una introducción formal al cálculo-II estocástico además de su uso para representar procesos biológicos y químicos diversos. Posteriormente se introduce a los sistemas de reacción-difusión (como el activador-inhibidor) y se muestran los resultados de las simulaciones en celdas aisladas como espacialmente.

Cabe mencionar que este trabajo en su versión de artículo de investigación fue aprobado para su publicación en la Revista Mexicana de Física el 14 de Octubre de 2011.

Capítulo 1

Introducción.

A través de la historia del conocimiento humano, la existencia de formas y estructuras en la naturaleza ha planteado numerosas preguntas sobre su formación. Muchos patrones pueden ser identificados a nuestro alrededor. Por ejemplo, si tenemos la oportunidad de visitar un zoológico, podremos apreciar los bellos patrones rayados que poseen los tigres y cebras, mientras que los leopardos y las hienas poseen patrones de manchas en su piel. A su vez, las jirafas poseen patrones manchados y las mariposas los tienen de todos los tipos. Pero no sólo en los animales podemos observar patrones, si nos detenemos un momento en nuestras vacaciones cuando visitemos alguna playa; podremos observar los patrones de ondas que se forman en de los océanos, o en lagos y estanques también; probablemente pocos habrán notado que en el desierto la arena se mueve con el mismo patrón de ondas formadas por el viento. En el cielo podemos observar cómo se forman arcos circulares de colores que llamamos arcoiris. Vemos también aves alejándose al atardecer volando a sus nidos en formación de “v”; o en ocasiones franjas de nubes que se mueven paralelamente a merced del viento. Si mirásemos hacia abajo, a nuestro jardín, podremos apreciar cómo se forman los arreglos de hojas, pétalos, semillas y flores en una forma espiral muy especial. Esta espiral guarda una relación matemática con la secuencia de Fibonacci 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... la cual fue estudiada por Leonardo de Pisa en el siglo XIII.

De este modo es natural que la curiosidad científica trate de explicar el porqué de estos elegantes patrones a nuestro alrededor. Con este propósito físicos y matemáticos han tratado de describir la formación de estos patrones y

estructuras complejas usando diversos mecanismos. En particular, un área del conocimiento sumamente interesante es la biología del desarrollo, pues, como su nombre lo indica, describe el proceso mediante el cual los organismos crecen y se desarrollan. La clave de estos procesos subyace en mecanismos que los científicos denominan diferenciación celular y la morfogénesis (del griego *morphê*: forma y *genesis*: creación).

La diferenciación celular es un proceso en el cual las células de un cigoto en crecimiento se convierten en células más especializadas formando parte de un tejido celular. La morfogénesis por su parte, es el proceso por el cual las células parecen reconocer su ubicación espacial dentro del cigoto y, figurativamente, deciden diferenciarse dirigiéndose hacia algún destino celular en función de su posición. Ambos mecanismos permiten que el cigoto adquiera la forma de un organismo maduro con todas las funciones y capacidades normales de su especie.

Es cuando surge la pregunta de, ¿cómo modelar estos fenómenos adecuadamente?. Para responder a dicha pregunta, los científicos poseen en la actualidad una gran cantidad de formas de describir este tipo de fenómenos, tanto biológicos como de otras áreas del conocimiento humano: física, finanzas, medicina, etc. En las distintas ramas de la biología, por ejemplo, existen representaciones de DNA en tres dimensiones para biología molecular, simulaciones de poblaciones en ecología, modelos estadísticos en paleontología, modelos de difusión en genética, etc. Y para el caso concreto de la biología del desarrollo y la morfogénesis existen los llamados sistemas de reacción difusión propuestos por el matemático inglés Alan M. Turing, los cuales abordaremos más adelante.

Estos modelos son utilizados para representar, analizar e incluso predecir comportamientos o sucesos importantes para enriquecer el estudio de los fenómenos biológicos. Algunos son muy detallados y pueden predecir cuantitativamente datos importantes, mientras que otros pueden solo describir cualitativamente los fenómenos. Suele suceder también que hay modelos que no se ajusten a las observaciones experimentales de un fenómeno dado, y es natural pensar si el modelo es bueno o malo, pero más allá de estos calificativos lo que está detrás es si el enfoque usado fue el apropiado. Así, cada modelo tiene sus ventajas y desventajas para su uso.

En la presente tesis hacemos un estudio de la morfogénesis desde una forma no tradicional, utilizando una herramienta desarrollada en las ciencias de computación, llamada Cálculo-II. Este formalismo ha sido utilizado con éxito para representar y simular correctamente una gran variedad de fenómenos biológicos y químicos, con precisión como lo hacen las ecuaciones diferenciales.

Este formalismo se centra en las nociones de concurrencia y movilidad, ambos términos de las ciencias de la computación que surgen al representar fenómenos computacionales con una gran cantidad de interacciones y comunicación entre agentes. Un ejemplo de ello puede ser el acceso a una base de datos por múltiples usuarios, o la misma Internet.

Es aquí donde la interacción entre el cálculo-II y la biología se da de forma natural. Esto debido a la gran cantidad de moléculas, genes, células, y en general interacciones, que un ser vivo posee dentro de una población dada. Además del hecho de que al interactuar estos agentes cambian su funcionamiento, su comportamiento, o simplemente su ubicación espacial.

El cálculo-II es pues una herramienta viable para la modelación de fenómenos biológicos de distinta naturaleza. Su mayor uso se ha desarrollado para la modelación de redes de regulación genética. Cabe mencionar que en la actualidad existen extensiones que permiten modelar ambientes, o localidades, donde suceden las interacciones entre agentes, difusión entre membranas y otros fenómenos relacionados. En nuestro caso basta utilizar la versión estocástica del cálculo-II que nos permitirá simular reacciones químicas y redes de regulación biológicas.

Existen también varias ventajas del uso del cálculo-II con respecto a las ecuaciones diferenciales, entre ellas el hecho de que es posible representar una gran cantidad de agentes sin tener que representar cada interacción posible explícitamente. Otra ventaja es que las descripciones de cada agente son de forma modular, con lo cual es posible añadir o quitar agentes sin que la descripción formal se altere significativamente. Además el uso de la movilidad permite establecer una forma de comunicación entre los mismos agentes más sofisticada, permitiendo también el incorporar más detalles biológicos. La movilidad en el contexto de la morfogénesis puede pensarse como un mecanismo en el cual se transmite información posicional entre agentes, dando pie a la diferenciación celular.

En la presente tesis hacemos uso del cálculo-II para describir la morfogénesis. Nuestra descripción se basa en un mecanismo clásico que se ha propuesto para la formación de patrones en la naturaleza, el Activador-Inhibidor.

La estructura de la tesis es la siguiente: en la sección 2.1 se plantea el panorama general de las álgebras de procesos. En la sección 2.2 presentamos la teoría preliminar necesaria para el estudio del cálculo-II. Y en la sección 2.3 se recopilan las nociones básicas del cálculo-II y sus propiedades, y damos un ejemplo del uso de la movilidad. En el capítulo 3 desarrollamos aplicaciones para el modelado

de fenómenos químicos, redes de regulación y compuertas lógicas con sus códigos respectivos en el apéndice A. Posteriormente, en el capítulo 4 retomamos el problema de la morfogénesis y proponemos un nuevo modelo de este fenómeno en la sección 4.4 Finalmente, en la sección 5 damos nuestras conclusiones.

Para aquellos lectores que busquen introducirse formalmente al cálculo-II se les recomienda que incluyan el capítulo 2 secciones 2.2 y 2.3 en su lectura, mientras que los lectores que sólo busquen conocer las aplicaciones la modelación se les sugiere pasar directamente al capítulo 3.

Cabe señalar que la presente tesis es una adaptación del artículo de investigación “La morfogénesis como resultado de la transmisión e integración de información biológica” el cual fue aceptado para su publicación en la *Revista Mexicana de Física* el 14 de Octubre de 2011.

Capítulo 2

Álgebras de procesos.

2.1. Preliminares, concurrencia y movilidad.

El describir formalmente sistemas computacionales comunicándose entre sí y ejecutándose al mismo tiempo (*concurrencia*) ha sido el tema de estudio de varios trabajos en ciencias de la computación en las últimas tres décadas. Estos sistemas al comunicarse en ocasiones intercambien información lo cual cambia su conectividad (y por lo tanto su estructura). Estos cambios de conectividad pueden darse al añadir o retirar formas de comunicación. A este fenómeno se le denomina *movilidad*¹.

A la fecha, existen distintos formalismos matemáticos que representan y describen a este tipo de sistemas concurrentes. Algunos ejemplos de estos formalismos son *Petri Nets* [19], *Communicating Sequential Processes* (CSP [9]), *Algebra of Communicating Processes* (ACP [3]) y *Calculus of Communicating Systems* (CCS [13]) . Dentro de estos formalismos existe una familia llamada *álgebras* o *cálculos de procesos* (*process calculi*) las cuales fueron desarrolladas para describir con un lenguaje de alto nivel las interacciones, comunicaciones y sincronizaciones de agentes o procesos² en estos sistemas concurrentes.

En particular, CCS [13] es un tipo de álgebra de procesos, presentada por Robin Milner en 1980, en la cual se expresan agentes como procesos del álgebra. En CCS es posible expresar comunicación entre procesos, los cuales pueden ser ejecutados al mismo tiempo, y además puede existir una elección no determinista

¹Del inglés: *mobility*.

²El uso del término agente y proceso en ocasiones es indistinto, esto se ejemplifica más adelante.

de alguno de ellos, entre otras propiedades. Esta álgebra de procesos presenta leyes que permiten manipular y analizar las descripciones de los agentes dados permitiendo establecer una noción formal de equivalencia entre dichos agentes. Esta noción de equivalencia es conocida como *bisimulación* y, como lo veremos más adelante, es sumamente importante.

A pesar de los avances de CCS para representar sistemas concurrentes, no permite representar movilidad causada por los cambios de interconectividad de los procesos. En base a esto Robin Milner, Joachim Parrow y David Walker en 1992 publicaron un álgebra de procesos descendiente de CCS llamada Cálculo-II [14] en la cual es posible expresar la movilidad dentro de las descripciones de los procesos. De este modo el cálculo-II permitió describir y analizar sistemas concurrentes de agentes cuyas configuraciones o conexiones cambian constantemente al interactuar entre ellos.

Parte del contenido del presente capítulo fue tomado de las notas de clase del curso *Lógicas No Clásicas* del Dr. Pedro Arturo Góngora Luna impartido en la Facultad de Ciencias, UNAM en 2008.

2.2. Nociones básicas del Cálculo-II.

2.2.1. Autómatas.

En ciencias de la computación, un autómata es la representación abstracta de una máquina de estados. Estas máquinas abstractas cambian su estado interno de acuerdo con su interacción con un ambiente externo. Dicha interacción se da, generalmente, con el intercambio de símbolos de un conjunto llamado alfabeto.

En [15], Robin Milner propone utilizar la teoría clásica de los autómatas como un modelo para representar sistemas concurrentes. En específico, Milner propone una convención en la cual dependiendo de la interconexión del autómata, distintos procesos conectados dentro de él pueden interactuar y evolucionar de acuerdo a sus descripciones propias.

Para ello el autómata utiliza un alfabeto, que será un conjunto numerable de nombres *Act*. Llamaremos acciones a los elementos de este conjunto. Con el conjunto *Act* damos la siguiente definición:

Definición 1. Un autómata A sobre *Act* esta compuesto de:

- un conjunto $Q = \{q_0, q_1, \dots\}$ de estados,
- un estado $q_0 \in Q$ llamado el estado inicial,
- un subconjunto $F \subseteq Q$ llamado el conjunto de estados de aceptación, o finales,
- un conjunto $T \subseteq Q \times Act \times Q$ llamado de transiciones.

Una transición $(q, a, q') \in T$ se escribe $q \xrightarrow{a} q'$. Decimos además que el autómata A es *determinista* si para cada pareja (q, a) sólo existe a lo más un $q' \in Q$ tal que $q \xrightarrow{a} q'$. En caso contrario llamaremos a A un autómata no determinista. \square

Con esta definición de autómata Robin Milner propuso un concepto llamado *caja negra*, donde se representa explícitamente la interacción con algún agente externo y el autómata. Dada una cadena $s \in Act^*$ no importa si se llega a un estado final (aceptación) o no, sino que la pregunta es si el autómata puede reproducir s dadas ciertas interacciones con el agente externo.

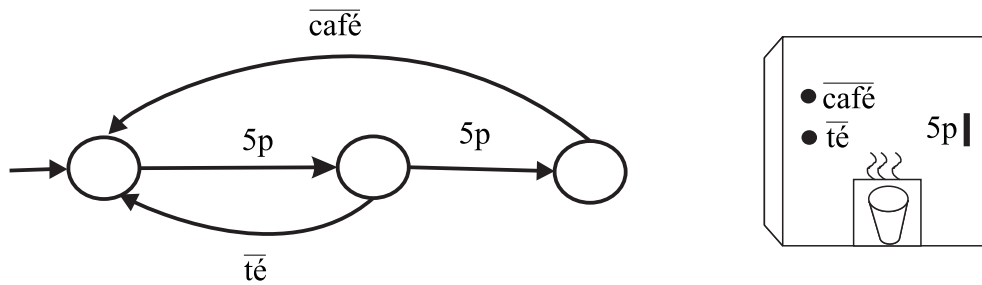


Figura 2.2.1: Analogía con una máquina de cafés.

Un ejemplo clásico de caja negra definida por un autómata A sobre un conjunto $Act = \{5p, \overline{t\acute{e}}, \overline{café}\}$; es una máquina de bebidas donde el agente externo será una persona que deposite $5p$ para obtener un té, ó deposite otros $5p$ y obtenga un café de la máquina (Figura 2.2.1).

2.2.2. Sistemas de Transiciones Etiquetados y la bisimilitud.

La definición intuitiva de *caja negra* puede formalizarse con la ayuda de un *sistema de transiciones etiquetado* (*LTS* por sus siglas en inglés). Estos sistemas están contruidos sobre un conjunto N de nombres como a, b, c, \dots (en nuestro ejemplo anterior usamos $t\acute{e}, café, 5p$) y el conjunto $\overline{N} = \{\overline{a} \mid a \in N\}$ cuyos elementos llamaremos *co-nombres*. Suponemos que N y \overline{N} son disjuntos y denotamos su unión como $L = N \cup \overline{N}$, donde L es llamado conjunto de *etiquetas*. Formalmente:

Definición 2. Un *Sistema de Transiciones Etiquetado* (*LTS*) sobre Act es una pareja (Q, T) en la cual:

- Q es un conjunto de estados,
- $T \subseteq (Q \times Act \times Q)$ es una relación ternaria también llamada *relación de transición*.

Si $(q, \alpha, q') \in T$ lo escribimos también como $q \xrightarrow{\alpha} q'$. Si $q \xrightarrow{\alpha_1} q_1 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} q_n$ entonces decimos que q_n es derivable de q usando la cadena $\alpha_1\alpha_2\dots\alpha_n$. \square

Nota: Un LTS puede verse como un autómata sin estado inicial ni estados finales.

Es natural preguntarse dados dos LTS's si son equivalentes o no. Al tratar de contestar esta interrogante, surge la idea de verificar si su comportamiento es el mismo, de tal forma que un agente que interactúe con ellos no los pueda distinguir. Para ello definimos:

Definición 3. Sea (Q, T) un LTS y sea S una relación binaria en Q . Llamamos a S una *simulación fuerte* en (Q, T) si dados pSq ,

- si $p \xrightarrow{\alpha} p'$ entonces existe $q' \in Q$ tal que $q \xrightarrow{\alpha} q'$ y $p'Sq'$

Decimos que q *simula fuertemente a p* si existe una simulación fuerte S tal que pSq . \square

En palabras podemos decir que si “ q simula fuertemente a p ”, o “ p es simulado fuertemente por q ”, significa que para cualquier camino que p tome en el autómata, q puede asociarlo con uno propio. De este modo todas las opciones de p están contenidas en las de q .

Una forma visual, utilizada frecuentemente dentro de las álgebras de procesos para representar que S es una simulación fuerte consiste en utilizar los diagramas:

$$\begin{array}{ccc} p & S & q \\ \text{Si } \downarrow \alpha & & \downarrow \alpha \\ p' & & p' S q' \end{array} \quad \text{entonces para alguna } q',$$

Nota: Se ha definido la noción de simulación como una relación entre estados de un mismo LTS. Además, dentro de un LTS pueden definirse uno o más autómatas.

Hasta ahora podemos saber si un autómata puede simular a otro, i.e., el segundo tiene una descripción análoga al primero. Para poder decir que dos autómatas son equivalentes necesitamos extender nuestra relación considerando ahora su inversa S^{-1} , como se explica en la siguiente definición:

Definición 4. Decimos que una relación binaria S en Q es una *bisimulación fuerte* en un LTS (Q, T) si S y su inversa son simulaciones. Decimos también que p y q son *fuertemente bisimilares* o *fuertemente equivalentes* si existe una bisimulación fuerte S tal que pSq . Esto se denota $p \sim q$. \square

Para ejemplificar nuestra definición de bisimulación fuerte utilizamos un LTS con dos autómatas (ver figura ??),

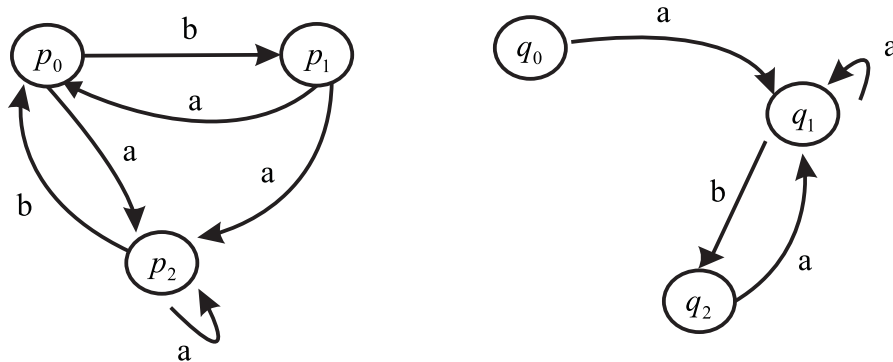


Figura 2.2.2: Autómatas bisimilares.

entonces $p_0 \sim q_0$ con la relación $S = \{(p_0, q_0), (p_1, q_2), (p_0, q_1), (p_2, q_1)\}$ al tener que p_0Sq_0 .

Como dato adicional, puede ser que en un mismo LTS dos autómatas se simulen fuertemente entre sí, sin embargo no necesariamente las simulaciones son bisimulaciones. En la figura 2.2.2 se observa un ejemplo de autómatas bisimilares

Con estas herramientas que hemos definido podemos comparar dos autómatas en un mismo LTS y ver si sus comportamientos son equivalentes. Hasta este punto hemos definido una buena parte de la teoría previa del Cálculo-II. En la siguiente sección presentaremos a esta álgebra de procesos y algunas de sus propiedades.

2.3. El Cálculo-II.

El Cálculo-II fue desarrollado con la intención de representar formalmente agentes que evolucionan e interactúan al mismo tiempo. Esta álgebra de procesos al ser descendiente de CCS conservó muchas de sus propiedades, sin embargo, nuevas características fueron añadidas con la finalidad de representar más acertadamente sistemas concurrentes. Por ahora, definiremos los conceptos básicos del Cálculo-II y luego trabajaremos un poco con la movilidad.

2.3.1. Definición.

Al estudiar sistemas concurrentes es útil describir los estados de los sistemas por medio de una *expresión que represente un proceso o expresión de un proceso*. Estas expresiones contienen información que describen el comportamiento y la estructura del sistema. Podemos utilizar indistintamente *expresión de un proceso* o *agente*.

Retomamos nuestro conjunto N de nombres tales como a, b, c, \dots y definimos un conjunto Id de identificadores de agentes, como A, B, C, \dots . Estos identificadores pueden ser parametrizados por nombres como en $C(a, x, y)$ donde la aridad de los parámetros es no negativa. Utilizamos las letras P, Q, R, \dots para representar los agentes.

Con estas convenciones definimos el conjunto del Cálculo-II como sigue:

Definición 5 El conjunto de *agentes* se define en notación BNF³ como:

$$P ::= 0 \mid \bar{x}y.P \mid x(z).P \mid \tau.P \mid (\nu x)P \mid [x = y]P \mid P \mid P \mid P + P \mid A(y_1, \dots, y_n)$$

donde:

- 0 es el agente nulo,

³Backus-Naur Form

- $\bar{x}y.P$ es llamado *prefijo negativo*. \bar{x} puede verse como un puerto de *salida* de un agente por el cual se envía el nombre y , y luego el agente se comporta como P . Cuando no se especifica y decimos que se envía un nombre vacío, y solo escribimos \bar{x} .
- $x(z).P$ es llamado *prefijo positivo*. x puede verse como un puerto de *entrada* de un agente por el cual se puede recibir algún nombre y y luego se comporta como $P\{y/z\}$.
- $\tau.P$ es llamado *prefijo silencioso*. Se realiza la acción τ y luego el agente se comporta como P . Observe que $\tau \notin N$
- $(\nu x)P$ es llamado *restricción*. ν es un símbolo especial no incluido en el conjunto de nombres. Este prefijo restringe la comunicación a través de los puertos x y \bar{x} a P . De este modo comunicaciones entre P y algún otro agente Q por medio de x están prohibidas.
- $[x = y]P$ es llamado *condicional*, si los nombres x y y son iguales entonces el agente se comporta como P .
- $P|Q$ es llamada *composición paralela*. Los agentes P y Q actúan al mismo tiempo, de tal modo que puede existir comunicación síncrona entre ellos por medio de algún nombre de la forma \bar{x} en P y x en Q .
- $P + Q$ es llamada *elección no determinista* entre P y Q .
- $A(y_1, \dots, y_n)$ es llamado *identificador* A con parámetros y_1, \dots, y_n .

El operador 0 es nulo, $\bar{x}y.$, $x(z).$, $\tau.$, (νx) y $[x = y]$ son operadores unarios; $|$ y $+$ son operadores binarios. La precedencia de los operadores está implícita por el orden en el que se dieron. \square

Necesitamos ahora definir el término *nombre acotado* ya que en las descripciones de los agentes como por ejemplo $(\nu x)P$ el nombre x está restringido sólo a P , lo que significa que x es un *nombre acotado*, formalmente:

Definición 6 En cada agente de la forma $x(z).P$ ó $(\nu z)P$, al encontrarse el nombre z dentro de los paréntesis decimos que z *aparece acotado*, y que su alcance son todas las apariciones de z en P . Decimos también que la

apariciencia de un nombre y en un agente es *libre*, si no se encuentra bajo el alcance de alguna aparición acotada de y . Denotamos el conjunto de nombres libres de P como $\text{fn}(P)$ ⁴. \square

Definición 7 Decimos que si un nombre que aparece dentro de un agente es *acotado* si no es libre. Además definimos el conjunto de todos los nombres acotados en P como $\text{bn}(P)$ ⁵. Denotamos como $\text{n}(P)$ al conjunto $\text{fn}(P) \cup \text{bn}(P)$ i.e., todos los nombres que aparecen en P . Nótese que $\text{n}(P) \subseteq N$. \square

Para cada agente podemos determinar en base a estas últimas dos definiciones sus nombres acotados y libres. Así dado P un proceso del cálculo-II obtenemos los conjuntos $\text{n}(P)$, $\text{fn}(P)$ y $\text{bn}(P)$ como en el cuadro 2.1.

P	$\text{n}(P)$	$\text{fn}(P)$	$\text{bn}(P)$
0	\emptyset	\emptyset	\emptyset
$\bar{x}y.P$	$\{x, y\} \cup \text{n}(P)$	$\{x, y\} \cup \text{fn}(P)$	$\text{bn}(P)$
$x(z).P$	$\{x, z\} \cup \text{n}(P)$	$\{x\} \cup \text{fn}(P) - \{z\}$	$\{z\}$
$\tau.P$	$\text{n}(P)$	$\text{fn}(P)$	$\text{bn}(P)$
$(\nu x)P$	$\{x\} \cup \text{n}(P)$	$\text{fn}(P) - \{z\}$	$\text{bn}(P) \cup \{z\}$
$P Q$	$\text{n}(P) \cup \text{n}(Q)$	$\text{fn}(P) \cup \text{fn}(Q)$	$\text{bn}(P) \cup \text{bn}(Q)$
$P + Q$	$\text{n}(P) \cup \text{n}(Q)$	$\text{fn}(P) \cup \text{fn}(Q)$	$\text{bn}(P) \cup \text{bn}(Q)$
$A(y_1, \dots, y_n)$	$\text{fn}(P)$	$\text{bn}(P)$	$\text{n}(P)$

Cuadro 2.1: Nombres libres y acotados de los procesos P del cálculo-II.

En la definición 5 cuando se define lo que es un prefijo positivo utilizamos la notación $P\{u/z\}$. Esto se explica en la siguiente definición.

Definición 8 Dado un agente P y un nombre u tal que $u \notin \text{n}(P)$ definimos $P\{u/z\}$ como el proceso que resulta de sustituir todas las apariciones libres de z en P por u . \square

⁴Del inglés *free name*

⁵También del inglés, *bound name*

Definición 9 Dos procesos P y Q son α -convertibles ó α -equivalentes si Q puede obtenerse de P a través de un número finito de cambios en nombres acotados en sus subtérminos, donde:

1. Si un nombre w no ocurre en un proceso P , entonces el proceso $P\{w/x\}$ se obtiene al sustituir con w cada ocurrencia libre de x en P .
2. Un cambio de nombres ligados en un término $x(z).P$ ó $(\nu z)P$ se obtiene al reemplazarlos por los términos $x.(w).(P\{w/x\})$ ó $(\nu z).(P\{w/x\})$, respectivamente.

Si P y Q son α -equivalentes escribimos $P =_\alpha Q$. \square

2.3.2. Transiciones del Cálculo-II.

En el Cálculo-II, como en CCS, existe una forma de representar las comunicaciones entre agentes, para ello utilizamos un conjunto de acciones y reglas de interacción. Este conjunto de acciones que denotamos Act esta construido de la siguiente forma:

$$Act = \{\bar{x}y \mid x, y \in N\} \cup \{x(y) \mid x, y \in N\} \cup \{\bar{x}(y) \mid x, y \in N\} \cup \{\tau\}$$

Así, en el Cálculo-II una transición tiene la forma:

$$P \xrightarrow{\alpha} P'$$

donde $\alpha \in Act$, formalmente:

Definición 9 Las *relaciones de transición* son el menor conjunto que satisfacen las reglas de acción del cuadro 2.2. (Ver [15])

$$\begin{array}{ll}
\tau - \text{ACT} : \frac{-}{\tau.P \xrightarrow{\tau} P} & \\
\text{INPUT} - \text{ACT} : \frac{-}{x(z).P \xrightarrow{x(w)} P\{w/z\}} \quad w \notin \text{fn}((z)P) & \text{OUTPUT} - \text{ACT} : \frac{-}{\bar{x}y.P \xrightarrow{\bar{x}y} P} \\
\text{SUM} : \frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'} & \text{MATCH} : \frac{P \xrightarrow{\alpha} P'}{[x=x]P \xrightarrow{\alpha} P'} \\
\text{IDE} : \frac{P\{\bar{y}/\bar{x}\} \xrightarrow{\alpha} P'}{A(\bar{y}) \xrightarrow{\alpha} P'} \quad A(\bar{x}) \stackrel{\text{def}}{=} P & \text{PAR} : \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \quad \text{bn}(\alpha) \cap \text{fn}(Q) = \emptyset \\
\text{COMM} : \frac{P \xrightarrow{\bar{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P|Q \xrightarrow{\alpha} P'|Q'\{y/z\}} & \text{CLOSE} : \frac{P \xrightarrow{\bar{x}(w)} P' \quad Q \xrightarrow{x(w)} Q'}{P|Q \xrightarrow{\alpha} (w)(P'|Q')} \\
\text{RES} : \frac{P \xrightarrow{\alpha} P'}{(y)P \xrightarrow{\alpha} P'} \quad y \notin \text{n}(\alpha) & \text{OPEN} : \frac{P \xrightarrow{\bar{x}y} P'}{(y)P \xrightarrow{\alpha} P'\{w/y\}} \quad y \neq x \quad w \notin \text{fn}((y)P')
\end{array}$$

Cuadro 2.2: Reglas de transición del cálculo-II. Las reglas SUM, COMM, PAR y CLOSE tienen una versión simétrica. Todas las reglas operan módulo $=_{\alpha}$.

Con estas transiciones hemos hecho explícito las reglas bajo las cuales la comunicación entre agentes del Cálculo-II puede realizarse. Dejamos un sencillo ejemplo para la siguiente sección.

Como lo hicimos en la sección 2.2.2, es natural preguntarse dados ciertos agentes del Cálculo-II si son equivalentes o no. Los conceptos que presentamos en dicha sección, esto es, un LTS y las relaciones de simulación y bisimulación fuertes, son válidas en CCS. Pero en Cálculo-II necesitamos crear un LTS especial que contemple las acciones en las cuales se envíen nombres acotados y posteriormente nos permita definir una relación de simulación y de bisimulación fuertes.

El desarrollo y presentación de la teoría necesaria para definir bisimulación en cálculo- π está fuera del alcance de esta tesis. Sin embargo, referimos al lector interesado a textos como [16] y [15]. En esta tesis, solo mencionaremos nuevamente que el uso de la bisimulación permite establecer cuando dos agentes poseen el mismo comportamiento siendo indistinguibles para un observador externo. También, existen además varios tipos de bisimulación lo que enriquece la cantidad de aplicaciones a sucesos de la vida real.

Para terminar desarrollaremos en la siguiente sección, en base a la propiedad de movilidad del Cálculo-II, un ejemplo clásico simplificado.

2.3.3. Movilidad.

Como mencionamos en la introducción, cuando varios sistemas computacionales interactúan entre sí, eventualmente pueden aparecer o desaparecer formas de interacción que expanden o limitan su comunicación.

Podemos representar lo anterior mediante agentes del cálculo- Π capaces de enviar y recibir formas de comunicación a través de canales. Cuando decimos que uno de estos agentes obtiene una nueva forma de comunicación nos referimos a que el agente adquiere un nuevo canal distinto a los que ya poseía, con el cual puede interactuar y ahora evolucionar de forma distinta.

Más concretamente, lo anterior nos dice que, al estar dos agentes interactuando por un mismo canal, alguno de ellos puede enviar un nombre de tal forma que el segundo lo recibe. Dependiendo de la definición del segundo agente, este nuevo nombre adquirido puede convertirse en un nuevo canal de comunicación.

Un ejemplo de esto puede ser representado por tres agentes, P , Q y R , en el cual P tiene acceso a R , y este acceso se lo transfiere posteriormente a Q . El proceso R , por su parte, comunica que adquirió el acceso con Q . Los agentes los definimos como:

$$P = \bar{x}v.0$$

$$Q = x(y).\bar{y}u.u$$

$$R = v(z).\bar{z}$$

y se ejecutan al mismo tiempo: $P|Q|R$. Al comunicarse P y Q por \bar{x} y x , respectivamente:

$$\bar{x}v.0 | x(y).\bar{u}.u | v(z).\bar{z} \xrightarrow{\tau} 0 | (\bar{y}u.u)\{v/y\} | v(z).\bar{z}$$

Luego, al darse la sustitución, el sistema puede comunicarse por el nuevo canal adquirido v :

$$0 | \bar{v}.v | v.\bar{v} \xrightarrow{\tau} 0 | v | \bar{v}$$

y finalmente se responde que se adquirió el canal enviando por él mismo la

respuesta.

$$0 \mid v \mid \bar{v} \xrightarrow{\tau} 0 \mid 0 \mid 0$$

En la figura 2.3.1 se ilustra el comportamiento del agente Q al adquirir el canal v . Por su parte, el proceso P desaparece del modelo.

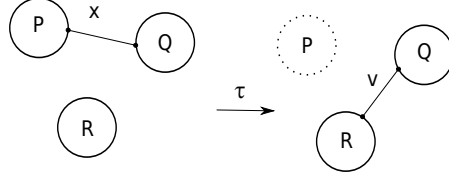


Figura 2.3.1: Movilidad en la comunicación.

Puede ser el caso también que exista otro agente S que pueda recibir el acceso a R , lo cual en la vida real si estamos representando una transacción por Internet, significaría que la seguridad se ve comprometida.

Para evitar este tipo de sucesos, podemos utilizar canales privados restringiendo el uso de canales a determinados agentes. De este modo las interacciones quedan resguardadas de algún otro agente externo.

En nuestro ejemplo anterior podemos restringir a x en P y Q , definiendo un agente P' de la siguiente forma:

$$P' = (\nu x) (\bar{x}v.0 \mid x(y).\bar{y}u.u)$$

así, al ejecutarlos en paralelo ($P' \mid R$):

$$(\nu x) (\bar{x}v.0 \mid x(y).\bar{y}.y) \mid v(z).\bar{z} \xrightarrow{\tau} (\nu x) (0 \mid (\bar{y}u.u)\{v/y\}) \mid v(z).\bar{z}$$

solamente Q recibió v por el canal privado x . Lo que dejó fuera la acción de algún posible agente S externo al sistema.

En este punto hemos desarrollado la teoría necesaria para realizar modelos matemáticos basados en cálculo- π . Procederemos a la explicación de algunos modelos en el siguiente capítulo. Aquellas personas que deseen profundizar sus conocimientos en cálculo- π pueden hacerlo revisando las referencias [16] y [15].

Capítulo 3

Modelación de fenómenos naturales.

En este capítulo vamos a revisar algunos modelos simples basados en cálculo-II. Cada modelo es implementado en un simulador llamado *Stochastic Pi Machine (SPiM)* [20]. La mayoría de los ejemplos a revisar en este capítulo fueron obtenidos y complementados de la documentación del simulador. Los códigos de simulaciones se pueden consultar en el apéndice A. La representación gráfica de los procesos en los modelos está basada en el trabajo de [22].

3.1. Notación.

Antes de mostrar los primeros modelos modificaremos un poco nuestra notación de los procesos en cálculo-II con la única finalidad de poder capturar los modelos en el simulador de computadora SPiM.

- El proceso $\bar{x}(y).P$ es ahora representado por $!x(y).P$.
- El proceso $x(y).P$ es ahora representado por $?x.P$.
- Las acciones τ tendrán ahora asociada una tasa $\varepsilon, \tau_\varepsilon$.
- La ejecución secuencial $.$ tendrá ahora el símbolo $;$.

3.2. Simulaciones de reacciones químicas.

3.2.1. Decaimiento radiactivo.

El decaimiento radiactivo es un fenómeno propio de los elementos químicos radiactivos. Este fenómeno es caracterizado por la descomposición espontánea de un núcleo, generando núcleos de menor masa, partículas pequeñas y energía. Dependiendo del isótopo, la tasa de descomposición es distinta, en algunos el proceso es instantáneo y en otros paulatino.

La ecuación diferencial que modela el decaimiento exponencial en el tiempo de un isótopo X es:

$$\frac{dX}{dt} = -cP \quad (3.2.1)$$

donde c es la tasa de proporcionalidad.

La ecuación (3.2.1) tiene solución

$$P(t) = P_0 e^{-ct}. \quad (3.2.2)$$

Podemos representar este decaimiento con el hecho de que el isótopo X desaparezca en el tiempo con cierta tasa c :



En cálculo-II es posible representar al isótopo X como un agente cuya única acción posible es convertirse en el agente vacío \emptyset . Esta acción tiene asociada la misma tasa c del decaimiento en (3.2.3). De este modo, el agente puede ser representado como:



Además podemos dar una representación gráfica de X como se muestra en la figura 3.2.1.

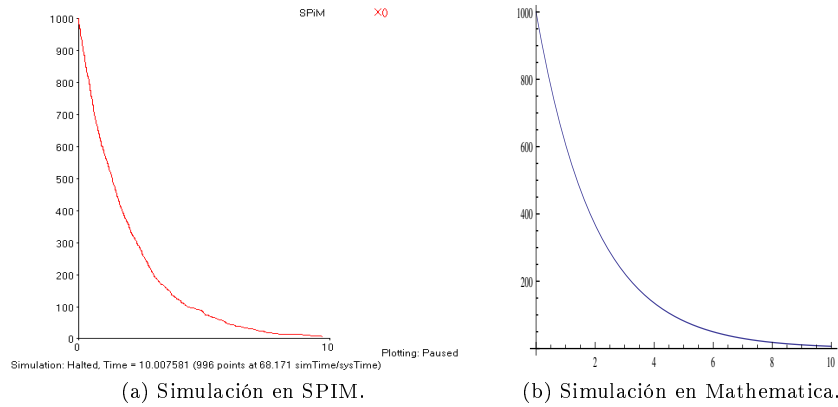


Figura 3.2.2: Simulación del decaimiento radiactivo con tasa $c = 0,5$. Para el código de la simulación ver el apéndice A.

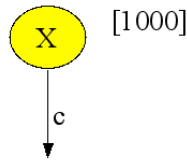
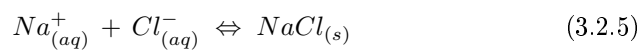


Figura 3.2.1: Decaimiento radiactivo.

En la figura 3.2.2 se muestra la simulación de (3.2.4). El resultado reproduce la curva clásica del decaimiento exponencial cuando se grafica la solución (3.2.2) de la ecuación diferencial.

3.2.2. Disociación del cloruro de sodio (NaCl).

El cloruro de sodio o NaCl es un compuesto iónico sólido que al disolverse en agua se disocia en iones Na^+ e iones Cl^- (3.2.5). Cuando se hace pasar electricidad a través de la solución, los iones Na^+ se dirigen al electrodo negativo y los Cl^- al electrodo positivo. Este flujo de iones establece una corriente eléctrica que equivale al flujo en un alambre metálico. Como la disolución de $NaCl$ conduce la electricidad, se dice que el NaCl es un electrolito.



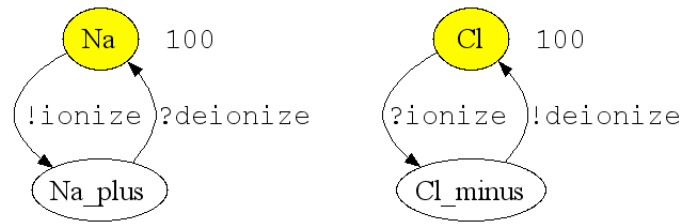


Figura 3.2.3: Formación de $NaCl$ representada en cálculo-II gráfico.

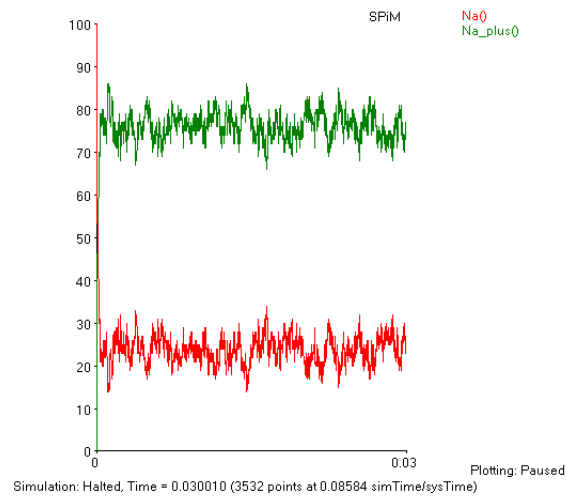


Figura 3.2.4: Concentraciones de Na y Na^+ en la asociación-disociación de $NaCl$. Para el código de las simulaciones ver el apéndice A.

Para el modelo en cálculo-II (3.2.6), representamos esta disociación con el uso de los agentes Na , Na_plus , Cl y Cl_plus , los cuales representan los estados posibles del electrolito $NaCl$. Los agentes se comunican a través de los canales $ionize$ y $deionize$. De esta forma se establece la posibilidad de asociarse y formar el $NaCl$ o la posibilidad de disociarse y regresar a su forma iónica.

$$\begin{aligned}
 Na &\equiv !ionize; Na_plus \\
 Na_plus &\equiv ?deionize; Na \\
 Cl &\equiv ?ionize; Cl_minus \\
 Cl_minus &\equiv !deionize; Cl
 \end{aligned}
 \tag{3.2.6}$$

Los agentes nuevamente pueden ser representados con una gráfica dirigida,

figura 3.2.3, con su respectiva simulación en la figura 3.2.4.

3.2.3. Formación de Amidas.

La formación de amidas puede llevarse químicamente a cabo por distintas reacciones. Una de estas reacciones es el uso de un ácido carboxílico y una amina formando la amida más agua, reacción (3.2.7). Este tipo de reacciones involucran la activación del ácido carboxílico por el método conocido como Shotten-Baumann [26], el cual involucra la conversión del ácido en cloruros de acilo.



Para el modelo de cálculo-II de esta reacción reversible definimos los agentes RNH_2 , $RNHCOR$, $RCOOH$ y H_2O los cuales, en base a (3.2.7), interactúan comunicándose a través de los canales *amine* e *hydrolysis* (3.2.8).

Finalmente, en la figura 3.2.5 representamos gráficamente el modelo, y su simulación en la figura 3.2.6.

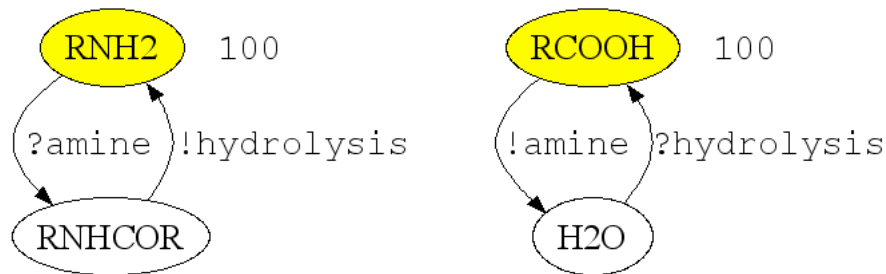
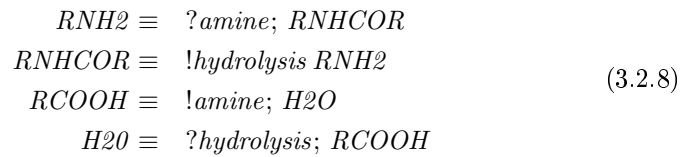


Figura 3.2.5: Formación de amidas representada en cálculo-II gráfico.

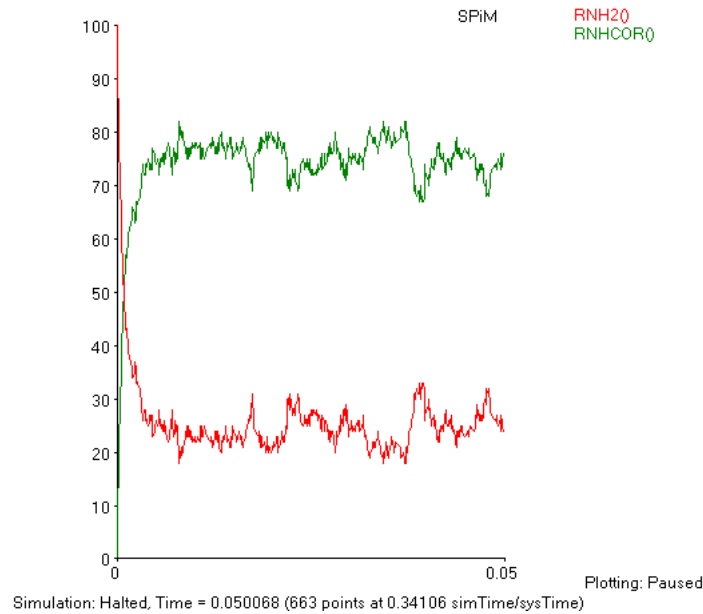


Figura 3.2.6: Concentraciones de RNH_2 y $RNHCOR$ durante la síntesis de amidas.

3.3. Simulaciones en procesos biológicos.

3.3.1. Sistema Biestable.

Los sistemas biestables son muy comunes en redes de regulación genética en la naturaleza. Algunos de estos sistemas están involucrados en el ciclo celular, la división celular y la apoptosis [11, 28]. Estos sistemas se caracterizan principalmente por poseer dos estados estables a través del tiempo (Figura (3.3.1)).

En cálculo-II podemos representar los sistemas biestables como dos genes a y b que se reprimen mutuamente (ver (3.3.1)). El gen a posee la capacidad de inhibir al gen b por medio del envío de la señal u a través del canal *inhibit*. Al recibir la señal de inhibición, el gen b se mantendrá reprimido permanentemente a menos que la señal u desaparezca. Para una descripción más a detalle sobre el funcionamiento del modelo, el lector puede consultar la sección 3.4.

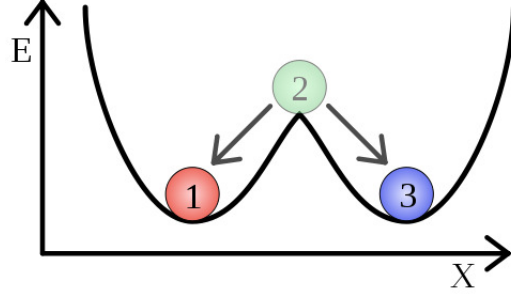


Figura 3.3.1: Sistema biestable de genes.

$$\begin{aligned}
 a &\equiv \tau_{tA}; (a | A) \\
 A(u) &\equiv \text{!inhibit}(u); A_b(u); A \\
 &\quad +?u \\
 &\quad +\tau_{dA} \\
 &\quad +\text{!bind}(u); A_B(u) \\
 A_B(u) &\equiv \tau_{dAB} \\
 &\hspace{15em} (3.3.1) \\
 b &\equiv \tau_{tB}; (b | B) \\
 &\quad +?\text{inhibit}(u); b_A(u) \\
 b_A(u) &\equiv \text{t}B'; (b_A(u) | B) \\
 B &\equiv \tau_{dB} \\
 &\quad +?\text{bind}(u); B_A(u)
 \end{aligned}$$

3.3.2. Represilador.

El *represilador* (en inglés *repressilator*) es una red de regulación sintética desarrollada por Michael B. Elowitz y Sanislas Leibler [6]. Esta red fue diseñada desde cero, con el propósito de reproducir un comportamiento oscilante en la expresión de una proteína verde fluorescente (GFP). Esta red fue implementada en *Escherichia coli* y fue posible obtener oscilaciones a nivel poblacional.

Los genes involucrados en el represilador son: *TetR* de resistencia a tetraciclina [25], *LacI* del operón de la lactosa [10],9 y λcl del operón del fago lambda

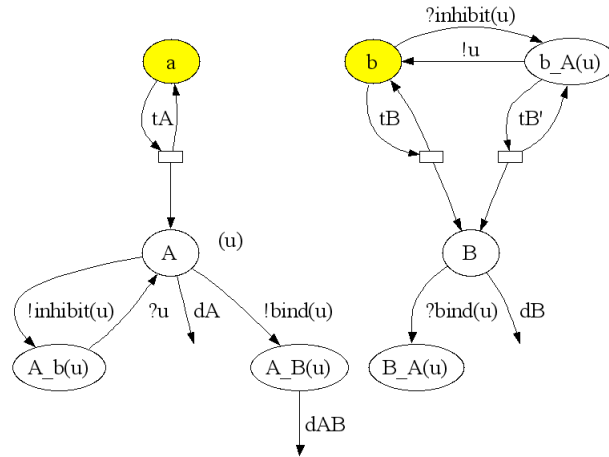


Figura 3.3.2: Sistema biestable de genes que se reprimen mutuamente representados en cálculo-II gráfico.

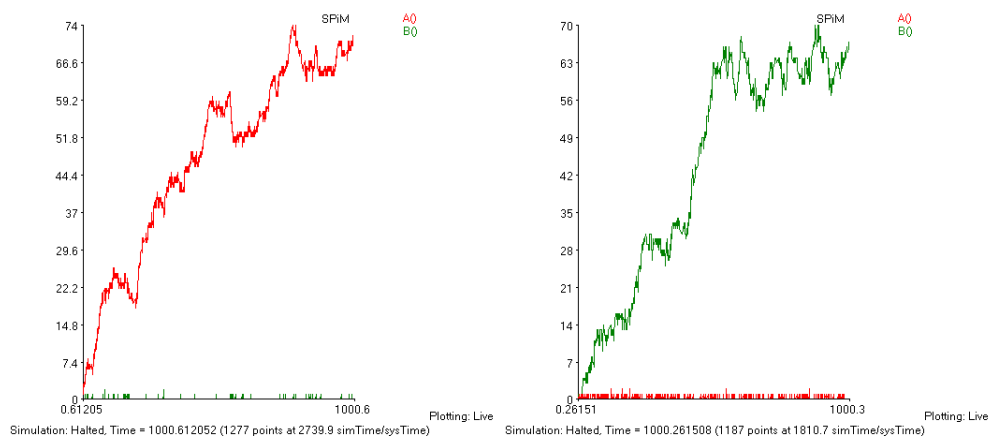


Figura 3.3.3: Simulaciones del sistema biestable, se muestran de los dos resultados posibles.

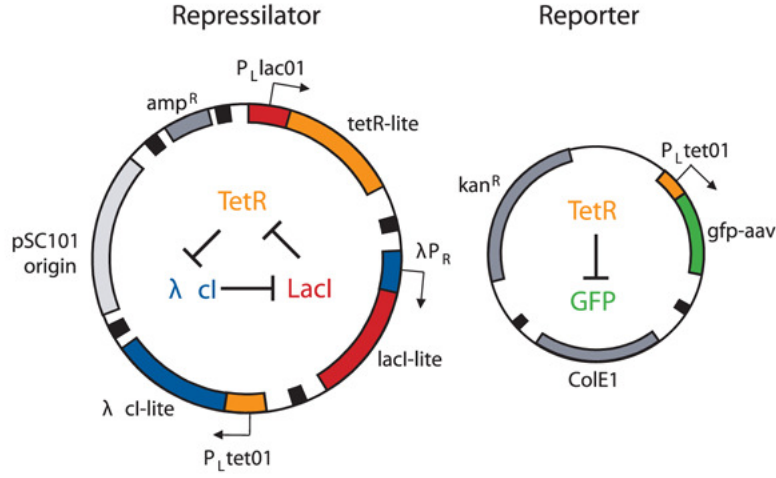


Figura 3.3.4: Represilador de Elowitz en forma de plásmidos. Figura tomada de [6].

[12].

$$\begin{aligned}
 GeneN &\equiv \tau_t; (GeneN | ProteinN) \\
 &\quad +?c; \tau_u; GeneN \\
 ProteinN &\equiv !a; ProteinN \\
 &\quad +\tau_d
 \end{aligned}
 \tag{3.3.2}$$

3.4. Aplicaciones en compuertas lógicas.

Existen distintas aplicaciones de las álgebras de procesos en la modelación de fenómenos biológicos. Una de ellas es la modelación y simulación de redes de regulación genética. Los modelos de compuertas lógicas son una simplificación de tres distintas formas de regulación que existen en las redes de regulación genética: constitutiva, negativa y positiva.

Presentamos a continuación un enfoque al modelaje de estos tres tipos de compuertas haciendo énfasis en la modularidad de los modelos y su simulación numérica. Para más información se recomienda consultar [4].

3.4.1. Compuertas constitutivas.

La característica de esta compuerta es describir los genes cuya transcripción es constitutiva o que no existe ninguna clase de regulación. El modelo se centra

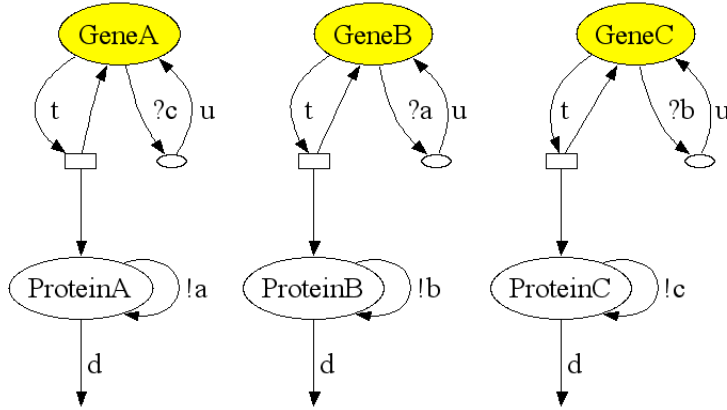


Figura 3.3.5: Representación de los genes del represilador en cálculo-II gráfico.

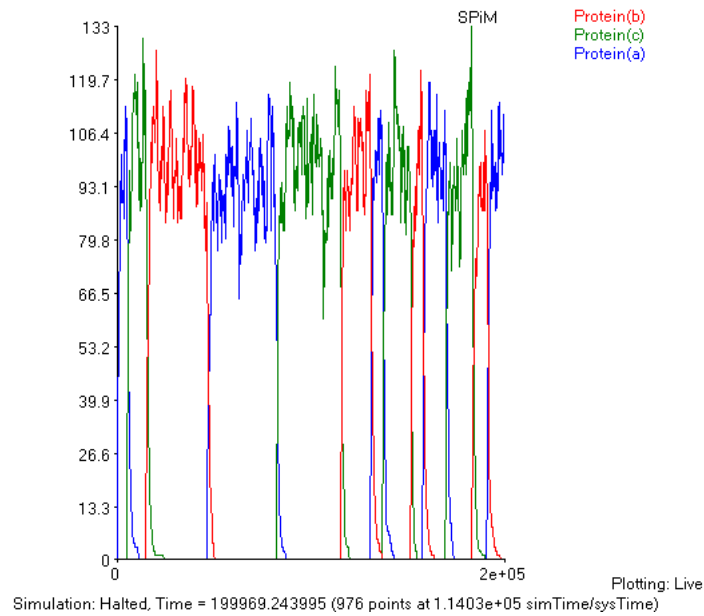


Figura 3.3.6: Simulaciones del represilador que muestran comportamiento oscilante.

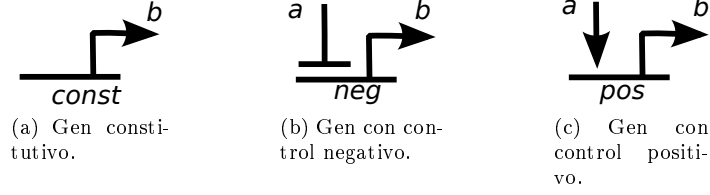


Figura 3.4.1: Esquemización de los genes regulados por distintos mecanismos. Figuras adaptadas de [4].

en las acciones que el gen puede realizar y en este ejemplo llamaremos *proceso* a cualquier mecanismo que realice acciones y avance pasando distintos estados.

$$const(b) = \tau_\varepsilon.(tr(b)|const(b)) \quad (3.4.1)$$

La ecuación (3.4.1a) muestra un gen $const(b)$ con transcripción constitutiva y sin regulación (nullary). El producto es una proteína traducida $tr(b)$ que puede unirse a un sitio b de otro gen, la definición de $tr(b)$ se da más adelante. En palabras, $const(b)$ espera el retraso estocástico τ con tasa ε y evoluciona en dos procesos paralelos, el primero $tr(b)$ y el segundo $const(b)$, el cual es el estado inicial.

3.4.1.1. Productos de los genes.

Un factor de transcripción $tr(b)$ realiza una elección no determinista (+) entre unirse a un promotor disponible b a través de una acción de salida $!b$ ó tener un retraso (τ) con tasa δ . Si ocurre el primer caso de la elección no determinista, entonces la acción de salida interactúa con su correspondiente señal de entrada dada por el sitio b de algún promotor disponible. Luego (.) el factor de transcripción regresa a su estado inicial $tr(b)$ y esta listo para interactuar nuevamente. En el segundo caso de la elección no determinista, el factor de transcripción se degrada al proceso nulo (0).

$$tr(b) = !b.tr(b) + \tau_\delta,0 \quad (3.4.2)$$

Observamos que $tr(b)$ tiene la capacidad de interactuar múltiples ocasiones con promotores con sitios b si están disponibles. Sin embargo, existe la posibilidad que se degrade también. Sin sitios de unión disponibles, dada una cierta cantidad de factores de transcripción, el factor de transcripción comenzará a

degradarse de forma exponencial.

3.4.2. Puerta unaria: gen con control negativo.

Una puerta unaria es un gen con la posibilidad de ser reprimido o regulado negativamente.

$$neg(a, b) = ?a \cdot \tau_\eta \cdot neg(a, b) + \tau_\varepsilon \cdot (tr(b) | neg(a, b)) \quad (3.4.3)$$

$neg(a, b)$ tiene la capacidad de elegir (+) entre transcribir constitutivamente o recibir un estímulo de represión (Figura 3.4.1b). La transcripción constitutiva es representada por (3.4.2) pero en esta ocasión se encuentra en competencia con el estímulo de represión. Si ocurre una interacción con $?a$ entonces el gen es llevado a un retraso estocástico τ_η durante el cual el gen se mantiene reprimido (puede verse como ocupado) y después regresa al estado original.

3.4.3. Puerta unaria: gen con control positivo.

Este modelo representa un gen $pos(a, b)$ que es activado por un control positivo (Figura 3.4.1c).

$$pos(a, b) = ?a \cdot \tau_\eta \cdot (tr(b) | pos(a, b)) + \tau_\varepsilon \cdot (tr(b) | pos(a, b)) \quad (3.4.4)$$

Es casi idéntico a (3.4.3) sin embargo a través de la interacción $?a$ el gen adquiere una tasa mayor de transcripción τ_η para $tr(b)$.

3.4.4. Simulaciones de compuertas unarias.

Partiendo de las descripciones de las compuertas lógicas antes presentadas es posible generar diversas combinaciones. En la figura 3.4.2 se muestran distintas combinaciones de las redes de regulación vistas como compuertas lógicas.

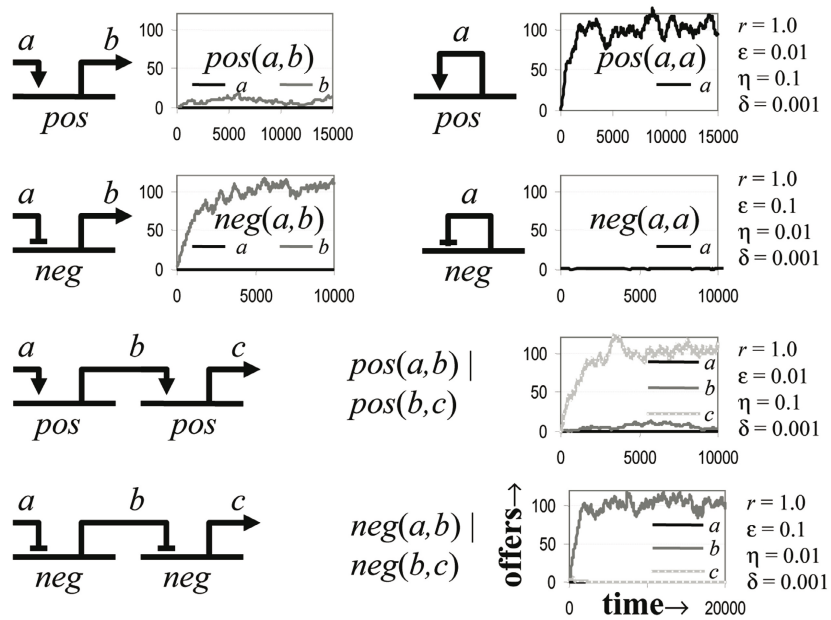


Figura 3.4.2: Redes de regulación como compuertas lógicas. Figura tomada de [4].

Capítulo 4

La morfogénesis.

La formación de estructuras en la Naturaleza es uno de los problemas más interesantes y complejos en la biología del desarrollo. Diversos mecanismos y modelos han sido propuestos para explicar el surgimiento de patrones en sistemas biológicos. En este trabajo nos enfocamos en uno de ellos, el que propuso A. Turing en 1952 que él mismo llama *La base química de la morfogénesis* [27]. Esta propuesta ha sido controvertida y sujeta a muy diversas críticas, en parte al menos, porque las sustancias que en su planteamiento son responsables de la morfogénesis al difundirse y reaccionar químicamente de manera simultánea, han sido difíciles de identificar en sistemas biológicos (ver sin embargo [5]), en donde se proporciona evidencia experimental de la existencia de este tipo de patrones en química).

Nuestro objetivo es hacer ver que el modelo de Turing puede justificarse sin necesidad de que haya sustancias que se difundan en el medio considerado, sino que este proceso de difusión puede ser sustituido por otros que involucren diversos tipos de comunicación (señalización intercelular de diversas índoles, incluyendo transmisión de señales a través de la membrana celular, transporte activo, procesos mecánicos, efectos piezoeléctricos, etc.). Esto siempre y cuando, de manera equivalente, haya un morfógeno *efectivo* que se difunda y que proponemos identificar como la información que se propaga a través del sistema.

Para ello, partimos del modelo original de ecuaciones de reacción-difusión propuesto por Turing y formulamos una versión discreta, tal como se hace normalmente en la implementación del método de diferencias finitas. Hacemos ver que dicha discretización no es otra cosa que una regla de mayoría en el contexto

de autómatas celulares. Posteriormente usando metodologías adecuadas para el estudio de procesos concurrentes de comunicación en sistemas biológicos provenientes del cálculo Π -estocástico [23] y su extensión para representar sistemas bioquímicos [24], mostramos que este modelo discreto es equivalente a considerar la transmisión de información a través de canales de comunicación, de tal forma que el estado del sistema en cierta posición a un cierto tiempo depende de la información recibida, de su estado mismo en tiempos anteriores y de la forma en la que integre a ambos. Finalmente, utilizamos esta propuesta para presentar ejemplos específicos. Desarrollamos en detalle una versión del sistema activador-inhibidor [7] que es un modelo prototípico y mostramos simulaciones numéricas.

La estructura del capítulo es la siguiente. En la primera sección recordamos los elementos centrales del modelo de Turing y mostramos que en una versión discreta, la difusión puede entenderse como señalización a través de canales. En la sección dos presentamos un ejemplo específico de sistema de reacción difusión, el activador-inhibidor. En la sección tres presentamos nuestra justificación de uso de las álgebras de procesos y en particular del cálculo Π -estocástico que requerimos para implementar el modelo. Dicho modelo y las simulaciones correspondientes para el sistema activador-inhibidor constituyen el contenido de la sección cuatro.

4.1. El modelo de Turing.

En el modelo de Turing original [27] se propone que la formación de ciertas estructuras biológicas (patrones espaciales) se debe a la combinación de dos mecanismos. Por una parte se propone la existencia de dos químicos que se difunden en la región en donde el proceso morfogénico tiene lugar. Estas sustancias se conocen como morfógenos. Por otra parte se supone también que, además, reaccionan químicamente entre sí. En un principio esta propuesta parece poco intuitiva, pues la difusión tendería a homogeneizar las concentración de los morfógenos en el medio. A su vez, es de esperarse que como resultado de la reacción se llegue a un equilibrio químico en el que la concentraciones se estabilicen en el tiempo. En resumen, cada uno de los mecanismos propuestos produciría patrones espacialmente homogéneos. Lo sorprendente de la propuesta es que al actuar de manera simultánea, los morfógenos que se difunden y

reaccionan químicamente pueden dar origen a estructuras espacio-temporales no homogéneas. Si denotamos por $u(x, y, t)$ y $v(x, y, t)$ las concentraciones respectivas de los morfógenos en un dominio bidimensional como funciones de la posición, (x, y) , y del tiempo, t , las ecuaciones de Turing se pueden escribir de forma adimensional como

$$\begin{aligned} u_t &= \gamma f(u, v) + \Delta u \\ v_t &= \gamma g(u, v) + d\Delta v, \end{aligned} \quad (4.1.1)$$

en donde γ es un parámetro positivo, $f(u, v)$ y $g(u, v)$ representan la cinética química de los morfógenos, d es la razón de los coeficientes de difusión de los mismos y Δ es el operador de Laplace en dos dimensiones[17].

A continuación mostramos que si analizamos por separado en el modelo de Turing la parte correspondiente a la difusión, ésta puede entenderse de manera efectiva como el promedio de la superposición de cuatro canales de comunicación de una celda con sus vecinos. En efecto, utilizando diferencias finitas[1] tenemos que el laplaciano de una función expresado como:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

y aproximando las primeras derivadas parciales mediante

$$\begin{aligned} \frac{\partial u}{\partial x} &= \frac{1}{\Delta x} (u(x + \Delta x) - u(x)) \quad \text{en } x \\ \frac{\partial u}{\partial x} &= \frac{1}{\Delta x} (u(x) - u(x - \Delta x)) \quad \text{en } x - \Delta x \end{aligned}$$

mientras que las segundas derivadas parciales quedarían aproximadas por

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2}(x) &= \frac{1}{\Delta x} \left[\frac{\partial u}{\partial x}(x) - \frac{\partial u}{\partial x}(x - \Delta x) \right] \\ &= \frac{1}{\Delta x^2} [u(x + \Delta x) - 2u(x) + u(x - \Delta x)] \end{aligned}$$

y finalmente la versión aproximada discreta del laplaciano

$$\frac{\partial^2 u}{\partial x^2}(x) = \frac{2}{\Delta x^2} \left[\frac{u(x + \Delta x) + u(x - \Delta x)}{2} - u(x) \right] \quad (4.1.2)$$

De manera análoga se puede proceder para la variable v .

Sin embargo, la última expresión (4.1.2) puede interpretarse de la siguiente manera. El valor del laplaciano en un celda se obtiene de tomar un promedio de los valores de las concentraciones en las celdas vecinas y su diferencia con

el valor del centro mismo. En otras palabras, se tiene una especie de regla de mayoría, de tal forma que si el promedio de las concentraciones en los vecinos es menor que en la celda central, el morfógeno saldría (flujo negativo), mientras que en el caso contrario, el flujo de morfógeno sería hacia dentro (flujo positivo).

Estas interacciones entre celdas centrales y sus vecinos ortogonales pueden representarse como procesos del cálculo- Π los cuales comparten canales de comunicación entre los mismos. La difusión de un morfógeno puede entenderse dentro del cálculo- Π como un nombre que se transmite entre procesos a través de los canales de comunicación entre celdas. Dependiendo de las interacciones y la movilidad de nombres que exista entre ellos, los procesos de la celda central evolucionarán en formas específicas todo de acuerdo a la regla de mayoría presentada. Más adelante se amplía esta propuesta.

4.2. Un sistema de reacción difusión clásico: El Activador-Inhibidor.

Uno de los mecanismos clásicos que cumplen las características postuladas por Turing es el llamado Activador-Inhibidor de [7]. Este sistema considera la existencia e interacción química y/o biológica de morfógenos llamados activador e inhibidor. Localmente el activador se autoinduce por medio de una autocatálisis y además promueve la producción de inhibidor, el cual a su vez inhibe la producción de activador. Esta descripción representa intuitivamente la cinética química del Activador-Inhibidor (Figura 4.2.1).

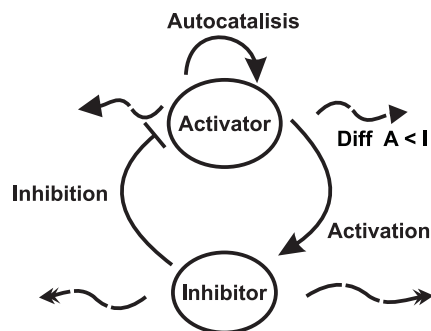


Figura 4.2.1: Sistema Activador-Inhibidor.

Es además usual suponer que la difusión del inhibidor es mayor que la del

activador, por lo que se habla de mecanismos de inhibición de largo alcance y de activación de corto alcance.

Inicialmente existen concentraciones distribuidas espacialmente de activador y/o inhibidor, ie., la condición inicial del sistema. La forma en que interactúan estos morfógenos dada una concentración inicial de activador en un lugar del medio específico es la siguiente: dado el mecanismo de activación de corto alcance y de inhibición de largo alcance, las zonas cerca del lugar donde inicialmente la concentración de activador es mayor se mantendrán activadas gracias a la autocatálisis o retroalimentación positiva, mientras que en lugares mas lejanos el inhibidor llegará más rápido y se mantendrán desactivadas (Figura 4.2.2). Con base en estas descripciones intuitivas es posible construir un modelo de Activador-Inhibidor por medio de un álgebra de procesos como el cálculo Π -estocástico de la siguiente sección, modelando la interacción de los morfógenos (cinética química) y su difusión en forma discreta utilizando la regla de mayoría presentada en la sección anterior. Mencionamos que, si bien este mecanismo ha sido sujeto a críticas, existen algunos sistemas biológicos en los que los mecanismos de activación e inhibición han sido identificados y este modelo ha sido usado de manera satisfactoria para explicar la aparición de patrones.

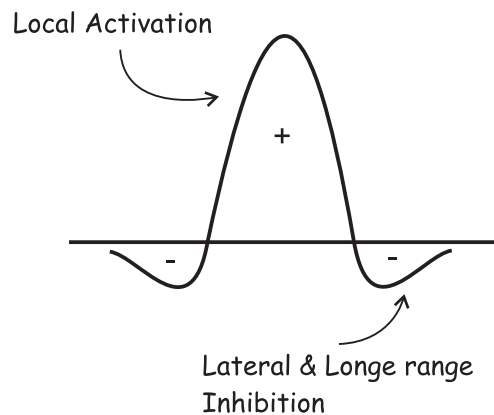


Figura 4.2.2: Activación lateral e inhibición a larga distancia.

4.3. Álgebras de procesos y sistemas de reacción difusión.

Las álgebras de procesos, en particular el cálculo Π , han sido ampliamente estudiadas y usadas en algunos procesos de comunicación (por ejemplo en telefonía celular) y en otros tipos de sistemas que involucran concurrencia como manejo de bases de datos y criptografía. Sin embargo, a pesar de haber sido utilizados en el estudio de fenómenos químicos y biológicos como reacciones químicas, redes de regulación genética, vías de señalización bioquímica, hasta donde tenemos conocimiento, no han sido usadas en el estudio de la formación de patrones y de la morfogénesis en biología. Es en este sentido la contribución fundamental de este trabajo, es decir, el uso de esta herramienta en el estudio de la emergencia de estructuras espacio-temporales. Dado lo anterior, referimos al lector no familiarizado con los elementos básicos del cálculo Π a las diversas referencias estándar [23, 15, 16].

El reproducir patrones espacio-temporales con álgebras de procesos requiere un tipo de semántica formal especializada con la cual se puedan describir correctamente sistemas de reacción difusión (SRD). Los mecanismos subyacentes en este tipo de sistemas consideran la cinética de las reacciones químicas de las especies involucradas y además las constantes de difusión de las mismas. Actualmente existen diversas extensiones del cálculo Π -estocástico (SPi) [23], como el cálculo Π -bioquímico [24], que modelan correctamente los sistemas de reacciones químicas como los encontrados en SRD; sin embargo no existe una semántica formal que pueda describir y modelar la difusión de sustancias químicas o biológicas. Además los agentes en SPi no poseen una referencia espacial específica¹. Por este motivo nuestro enfoque consiste en modelar estructuras espacio-temporales por medio de una aproximación distinta: usando cálculo Π -estocástico en un contexto discreto con índices en el espacio. Este enfoque tiene a su vez tiene su contraparte en ecuaciones diferenciales utilizando métodos de diferencias finitas (Sección 2).

Nos parece pertinente mencionar que desde el punto de vista de modelación, el SPi ofrece algunas ventajas sobre el enfoque tradicional de ecuaciones difer-

¹*“Unfortunately, the term mobility is overloaded with meaning and the notion of mobility supported by the Π -calculus encompasses only part of all the abstractions meaningful to mobility in a distributed system. For instance, the Π -calculus does not directly model phenomena such as the distribution of processes within different localities, their migrations, or their failures...”* [30]

enciales parciales. Específicamente, la posibilidad de incluir mecanismos de comunicación intercelular de manera mas realista que en el contexto de ecuaciones diferenciales parciales sólo se puede si se conoce la forma funcional explícita de tales mecanismos. Mientras que, en el contexto del SPI, la comunicación puede ser tomada en cuenta directamente justamente como canales adicionales, incluso de una manera análoga a como probablemente ocurran en la realidad, es decir por medio de canales en la membrana celular. En este sentido, cada enfoque ofrece ventajas y desventajas, sin embargo, creemos que a pesar de la dificultad inicial que pueda representar el familiarizarse con este formalismo, se pueden desarrollar modelos más realistas y flexibles.

El uso de SPI es justificado por la propuesta de que el Activador-Inhibidor (*AI*) puede ser visto como un sistema de regulación genética como los propuestos en [8] y modelados en Spi en [4]. Este sistema interactúa con sistemas análogos en distintas posiciones espaciales y dependiendo de estas interacciones, así como de su propio estado pueden generar estructuras espacio-temporales (Regla de Mayoría). Esta noción concuerda con la descripción de la discretización del laplaciano en el plano de la sección 2 cuando tomamos vecindades como la de Von Neumann de los autómatas celulares [18].

De este modo se pretende modelar un arreglo de $m \times n$ celdas representadas en el plano, cada una de las cuales interactúa con sus vecinos ortogonales y todas ellas contienen un sistema *AI*. Este arreglo se puede considerar como una abstracción de un tejido celular homogéneo. Según lo propuesto por Turing, necesitamos perturbaciones iniciales tales que el sistema global alcance un estado de inestabilidad y eventualmente se generen patrones estacionarios. Esta no homogeneidad la podemos alcanzar mediante la modificación del modelo *AI* en SPI produciendo las condiciones iniciales necesarias de manera aleatoria como lo veremos más adelante.

Un punto muy importante que es consecuencia de la descripción del sistema global con SPI al modelar un tejido celular, es que la difusión de un morfógeno *per se* puede ser reemplazada por una comunicación efectiva con celdas vecinas, lo que significa que lo que se transmite no es otra cosa sino información² y esto es suficiente para la formación de estructuras espacio temporales no triviales.

²Ya sea posicional como lo propuso Wolpert [29] o de otro tipo.

4.4. Modelo del sistema Activador Inhibidor usando Cálculo Π -Estocástico.

Tomando la descripción del Activador-Inhibidor [7] es posible modelar su comportamiento usando las nociones de compuertas lógicas positivas y negativas (*positive and negative gates*) encontradas en [16] con el fin de tratar el *AI* como un modelo de una red de regulación genética³ En este tipo de redes existe un número arbitrario de genes los cuales interactúan mediante regulaciones positivas y negativas, lo que eventualmente puede desencadenar la formación de alguna estructura no trivial en el organismo.

Nuestra propuesta contempla la existencia de dos genes o sitios de regulación con funcionamiento análogo al *AI*. Estos genes abstractos, pueden ser reprimidos o inducidos por distintos mecanismos. Sin embargo la forma de inhibición de los genes utilizada en el presente trabajo será el análogo a la represión por parte de algún elemento biológico o químico que se adhiere en sitios de regulación. Un ejemplo de ello es un oscilador biológico presentado en [6] de tres genes que se reprimen entre sí. La inducción por su parte será nuevamente el análogo al realizado en un promotor por parte de un factor de transcripción específico que desencadena una regulación positiva.

4.4.1. Modelo de una celda aislada

El primer modelo presentado describe una única célula con el sistema *AI* interno, es por ello que la difusión en el medio circundante no es considerada (4.4.1). En el modelo es necesario definir dos procesos principales en SPi: el *Activador* e *Inhibidor*. El proceso *Activador* puede ser modelado como una compuerta híbrida positiva y negativa; de este modo el activador recibe dos señales distintas, la propia (*act*) y la del inhibidor (*inh*). El proceso *Inhibidor* puede verse como una compuerta positiva donde la única señal recibida es la del activador (*act'*).

³El tratar de representar el Activador-Inhibidor como una red de regulación genética puede generar un debate en torno a la forma en la cual la interacción y difusión de los morfógenos preceden la diferenciación celular y de qué tan pertinente es esta descripción. Para una discusión más detallada véase también [5]

$$\begin{aligned}
\text{Activador} &\equiv \tau_{basA} \cdot (\text{SignA} \mid \text{SignA}' \mid \text{Activador}) \\
&\quad + \text{act} \cdot \tau_{ind} \cdot (\text{SignA} \mid \text{SignA}' \mid \text{Activador}) \\
&\quad + \text{inh} \cdot \tau_{unb} \cdot \text{Activador} \\
\text{SignA} &\equiv \overline{\text{act}} \cdot \text{SignA} \\
&\quad + \tau_{degA} \\
\text{SignA}' &\equiv \overline{\text{act}' } \cdot \text{SignA}' \\
&\quad + \tau_{degA} \\
\text{Inhibidor} &\equiv \tau_{basI} \cdot (\text{SignI} \mid \text{Inhibidor}) \\
&\quad + \text{act}' \cdot \tau_{ind} \cdot (\text{SignI} \mid \text{Inhibidor}) \\
\text{SignI} &\equiv \overline{\text{inh}} \cdot \text{SignI} \\
&\quad + \tau_{degI} \\
\text{AI} &\equiv (\text{Activador} \mid \text{Inhibidor})
\end{aligned} \tag{4.4.1}$$

Al ser visto el AI como una red de regulación genética, el proceso *Activador* mantiene un estado basal de producción de su propia señal. Esto es representado mediante una tasa denotada *bas* en τ_{bas} seguido de la producción de sus propias señales *SignA* y *SignA'*. Además puede recibir dos señales de regulación por los canales *act* e *inh* con ciertas tasas. Esta competencia entre producción basal o recepción de las dos señales adicionales es representada mediante una elección no determinista “+”. La función del canal *inh* es poder inhibir al *Activador*, el cual al recibir el estímulo se mantiene “ocupado” hasta que la señal del inhibidor pierda efecto, esto es representado por τ_{unb} . Cuando es completada una recepción a través del canal *act* (activación), el *Activador* comienza a producir sus señales *SignA* y *SignA'* con una mayor tasa de producción (inducción) dada por *ind* en la acción interna τ_{ind} . Las señales *SignA* y *SignA'* emiten la señal del activador en los canales *act* y *act'* respectivamente y al mismo tiempo existe la posibilidad de que se “degradaden” con tasa *degA* en τ_{degA} .

El agente *Inhibidor* por su parte posee una descripción parecida al proceso *Activador*. Una de las diferencias es que el proceso *Inhibidor* sólo puede recibir una señal externa, *act'* proveniente del *Activador* que representa una activación positiva del gen. La señal del *Inhibidor* es transmitida a través del canal *inh* la cual puede ser degradada al realizar la acción τ_{degI} con tasa *degI*.

En la actualidad existe una representación gráfica para el cálculo Π -estocástico llamada *GSII* [21], desarrollada para representar mediante una gráfica procesos del cálculo Π -estocástico pero con una sintaxis distinta. En la figura 4.4.1 se puede visualizar las capacidades de los procesos de interactuar más fácilmente. Para aclarar cuestiones de notación *!act* es igual a $\overline{\text{act}}$, e *?inh* es igual a *inh*; así como τ_{bas} por *bA* *degA* por *dA*

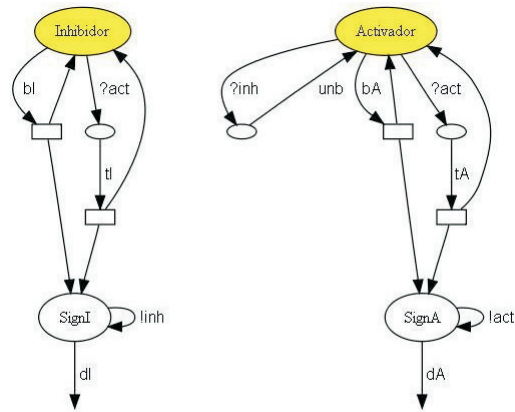


Figura 4.4.1: Sistema AI en GSI.

4.4.1.1. Resultados Numéricos

Para realizar simulaciones numéricas de nuestro modelo AI utilizamos un simulador desarrollado por A. Phillips, llamado *Stochastic Pi Machine* (SPiM) el cual fue desarrollado con base en una máquina abstracta que simula correctamente procesos del cálculo- Π [21]. El código de la simulación de nuestro modelo aparece en el Apéndice B.

Este resultado coincide con un modelo específico de red de una regulación genética en *Arabidopsis Thaliana* en [2]. En este trabajo se hace una aproximación discreta de la dinámica activador-inhibidor. Se parte del plano fase de las soluciones de las ecuaciones diferenciales que se mantienen alrededor de un punto de equilibrio y se considera que están bien descritas por un ciclo alrededor de dicho punto.

Luego de realizar varias simulaciones de nuestra versión del activador-inhibidor modelada en SPi, (Figuras 4.4.2 y 4.4.3), cuando no consideramos la comunicación con más celdas vecinas, obtenemos oscilaciones entre el Activador y el Inhibidor bajo los parámetros estándar tomados de otros ejemplos en la documentación del simulador. Esta observación es relevante, pues está relacionada con la aparición de ritmos y oscilaciones biológicas, pero es un tema que no abordamos en el presente trabajo y que estamos estudiando en la actualidad.

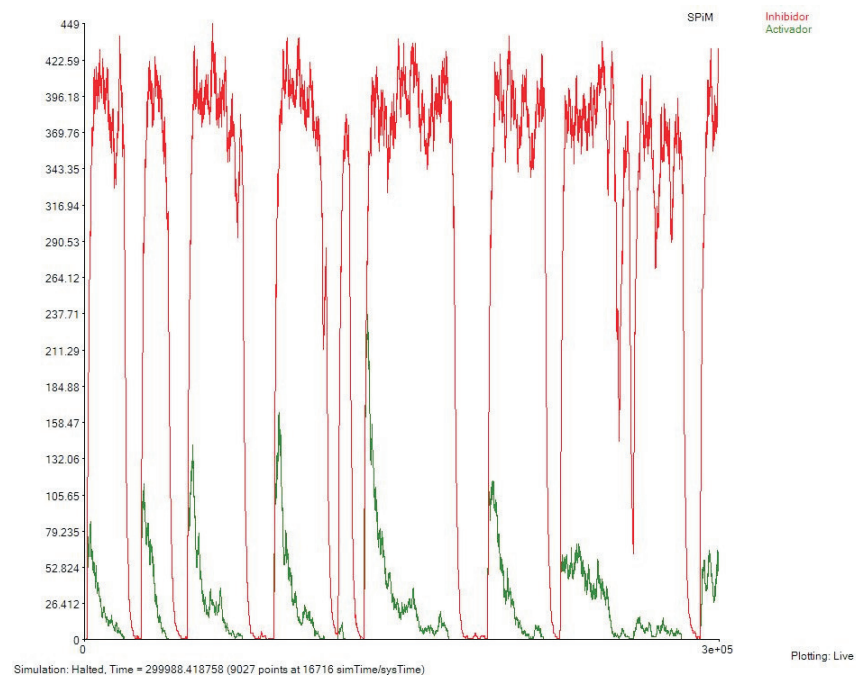


Figura 4.4.2: Simulación 1.

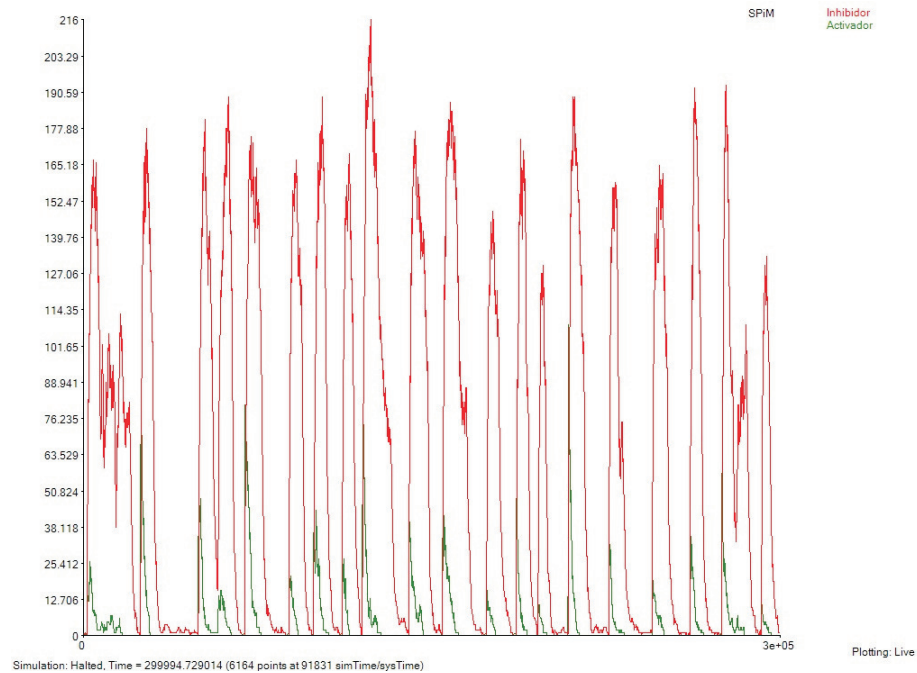


Figura 4.4.3: Simulación 2.

4.4.2. Modelo de un tejido celular, $m \times n$ celdas con un sistema AI cada una.

Con el fin de modelar el “tejido de células comunicándose entre sí”, i.e., con difusión, utilizamos un arreglo de $m \times n$ celdas, de forma análoga a un autómata celular con vecindad de Von Neumann (Figura 4.4.4). La celda central, así como sus vecinos poseen cada una un sistema AI interno. El modelo de $m \times n$ es una extensión de nuestro modelo anterior añadiendo canales de comunicación con los vecinos ortogonales (Figura 4.4.5).

En un tejido abstracto en el cual cada celda tenga un sistema AI interno y si no existe comunicación entre las celdas, el sistema se mantiene en punto de equilibrio estable [17]. Podemos interpretar biológicamente que el tejido abstracto se mantiene homogéneo. Por lo tanto, la condición interesante es cuando existe difusión entre las celdas.

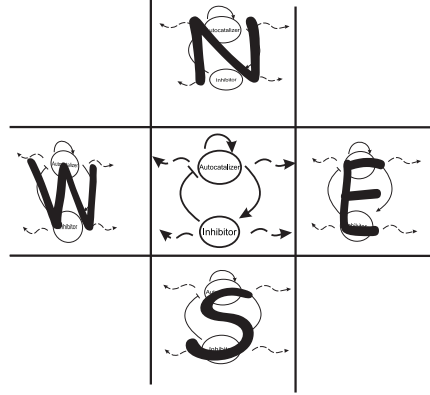


Figura 4.4.4: Vecindad de Von Neumann.

Partimos de nuestro primer modelo (4.4.1) basado en las nociones de compuertas lógicas positivas y negativas [16] del activador-inhibidor de [7] y añadimos un nuevo canal $diff_{ij}$ para cada celda donde ij es identificado de acuerdo a su posición en el arreglo. De este modo la comunicación de la vecindad con la celda i, j -ésima se realizará por medio del canal dif_{ij} .

El modelo que presentamos es una abstracción de la i, j -ésima celda, de tal forma que para simular un arreglo arbitrario de $m \times n$ es necesario renombrar los parámetros pA y pI especificando el nombre de los canales internos y los canales de comunicación de las celdas vecinas ortogonales. Para tener una referencia más fácil de la posición de estas últimas las ubicamos de acuerdo al conjunto $\{\text{North, South, East, West}\}$. Y de este modo establecemos la interconectividad del arreglo.

El proceso que representa al activador de AI es denotado por $Activador(pA)$ donde pA son un conjunto de parámetros que definen su posición espacial respecto de sus vecinos en el arreglo. Este proceso tiene cuatro capacidades distintas:

- $\tau_{basA} \cdot (Act(pA)|Activador(pA))$. El cual es una acción interna con tasa $basA$ que representa, como en sistemas de regulación genética, la actividad basal del gen activador. Cuando esta tasa se alcanza se ejecutan paralelamente los procesos $Act(pA)$ y $Activador(pA)$.
- $act_{ij} \cdot \tau_{transA} \cdot (Act(pA)|Activador(pA))$. Esto representa una recepción por el canal act_{ij} de la i, j -ésima celda del arreglo sin comunicación con vecinos. Esta recepción simula la retroalimentación positiva de nuestro sistema AI . Al suceder esta interacción,

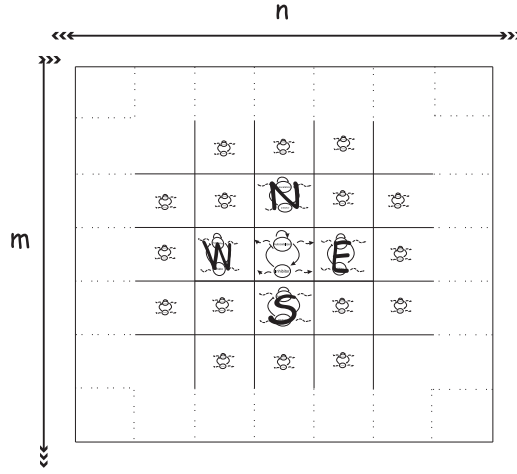


Figura 4.4.5: Arreglo de $m \times n$.

se da paso a una acción interna τ_{transA} con tasa $transA$ mucho mayor a $basA$ y luego se ejecutan paralelamente $Act(pA)$ y $Activador(pA)$.

- $inh\ ij$. τ_{unb} . $Activador(pA)$. La recepción por el canal $inh\ ij$ de la i, j -ésima celda representa la inhibición por parte del Inhibidor de AI de la célula misma. Al darse esta interacción el sistema se mantiene “ocupado” en la acción interna τ_{unb} . Biológicamente quiere decir que se reprime la acción del gen activador hasta que la tasa unb (muy baja) sucede, i.e., el represor pierde efectividad, lo que permite al gen estar listo para transcribir nuevamente, esto es regresa a la descripción $Activador(pA)$ original.
- $dif\ ij\ (com)$. $(Com(com) | Activador(pA))$. La recepción por el canal $dif\ ij$ representa la comunicación de los vecinos con la celda i, j -ésima. En la comunicación, los vecinos transmiten nombres como $act\ N$ los cuales sustituyen todas las ocurrencias de com en $(Com(com) | Activador(pA))$.

Por su parte el Inhibidor de AI es representado por el proceso $Inhibidor(pI)$, el cual tiene una descripción análoga en nuestro modelo al proceso $Activador(pA)$, pero posee sólo dos capacidades:

- τ_{basI} . $(Inh(pI) | Inhibidor(pI))$. Es el estado basal de transcripción del gen Inhibidor con tasa $basI$.
- $act\ ij$. τ_{transI} . $(Inh(pI) | Inhibidor(pI))$. La recepción por el canal $act\ ij$ representa nuevamente una retroalimentación positiva la cual da paso a la acción interna τ_{transI} con tasa $transI$ mucho mayor que $basI$ y luego se ejecuta $Inh(pI)$.

$$\begin{aligned}
\text{Activador}(pA) &\equiv \tau_{basA} \cdot (\text{Act}(pA) | \text{Activador}(pA)) \\
&\quad + \text{act } ij \cdot \tau_{transA} \cdot (\text{Act}(pA) | \text{Activador}(pA)) \\
&\quad + \text{inh } ij \cdot \tau_{unb} \cdot \text{Activador}(pA) \\
&\quad + \text{dif } ij(\text{com}) \cdot (\text{Com}(\text{com}) | \text{Activador}(pA)) \\
\text{Act}(pA) &\equiv (\text{IntSigA}(\text{act } ij) | \text{DiffA}(pdA)) \\
\text{IntSigA}(\text{act } ij) &\equiv \overline{\text{act } ij} \cdot \text{IntSigA}(\text{act } ij) + \tau_{degIntSigA} \\
\text{DiffA}(pdA) &\equiv \tau_{diffA} \cdot (\text{DiffSigA}(\text{difN}, \text{act } N) | \text{DiffSigA}(\text{difS}, \text{act } S) \\
&\quad | \text{DiffSigA}(\text{difE}, \text{act } E) | \text{DiffSigA}(\text{difW}, \text{act } W)) \\
\text{DiffSigA}(\text{difNeigh}, \text{act } Neigh) &\equiv \overline{\text{difNeigh}}(\text{act } Neigh) \cdot \text{DiffSigA}(\text{difNeigh}, \text{act } Neigh) \\
\\
\text{Inhibidor}(pI) &\equiv \tau_{basI} \cdot (\text{Inh}(pI) | \text{Inhibidor}(pI)) \\
&\quad + \text{act } ij \cdot \tau_{transI} \cdot (\text{Inh}(pI) | \text{Inhibidor}(pI)) \\
\text{Inh}(pI) &\equiv (\text{Sign}(\text{inh } ij) | \text{DiffI}(pdI)) \\
\text{DiffI}(pdI) &\equiv \tau_{diffI} \cdot (\text{DiffSig}(\text{difN}, \text{inh } N) | \text{DiffSig}(\text{difS}, \text{inh } S) \\
&\quad | \text{DiffSig}(\text{difE}, \text{inh } E) | \text{DiffSig}(\text{difW}, \text{inh } W)) \\
\text{IntSigI}(\text{inh } ij) &\equiv \overline{\text{inh } ij} \cdot \text{IntSigI}(\text{inh } ij) + \tau_{degIntSigI} \\
\text{DiffSigI}(\text{difNeigh}, \text{inh } Neigh) &\equiv \overline{\text{difNeigh}}(\text{inh } Neigh) \cdot \text{DiffSigI}(\text{difNeigh}, \text{inh } Neigh) \\
\text{Com}(\text{com}) &\equiv \overline{\text{com}} \cdot \text{Com}(\text{com}) + \tau_{degDiff} \\
pA &= (\text{act } ij, \text{inh } ij), (\text{difN}, \text{difS}, \text{difE}, \text{difW}), \\
&\quad (\text{act } N, \text{act } S, \text{act } E, \text{act } W) \\
pdA &= (\text{difN}, \text{difS}, \text{difE}, \text{difW}), (\text{aN}, \text{aS}, \text{aE}, \text{aW}) \\
pI &= (\text{act } ij, \text{inh } ij), (\text{difN}, \text{difS}, \text{difE}, \text{difW}), \\
&\quad (\text{inh } N, \text{inh } S, \text{ihn } E, \text{ihn } W) \\
pdI &= (\text{difN}, \text{difS}, \text{difE}, \text{difW}), (\text{inh } N, \text{inh } S, \text{ihn } E, \text{ihn } W) \\
Neigh &= \{\text{North}, \text{South}, \text{East}, \text{West}\} \\
AI \text{ } ij(pA, pI) &\equiv (\text{Activador}(pA) | \text{Inhibidor}(pI))
\end{aligned}$$

Los procesos auxiliares $Act(pA)$ e $Inh(pI)$ definen y ejecutan en paralelo las señales del AI . Estas señales las separamos en internas y difusibles por medio de los subprocesos, $IntSigA(act\ ij)$, $IntSigI(inh\ ij)$, $DiffA(pdA)$ y $DiffI(pdI)$. Los procesos $IntSigA(act\ ij)$ y $IntSigI(inh\ ij)$ representan la capacidad de interacción interna de la celda i, j al sustituir el canal interaccionar por los canales $act\ ij$ si es del activador ó $inh\ ij$ si es del inhibidor. Biológicamente significa que los genes activador y/o inhibidor interactúan internamente dentro de la célula en la que se encuentra o se puede degradar las señales con tasa $degInt$. Los subprocesos $DiffA(pdA)$ y $DiffI(pdI)$ tienen la capacidad de realizar las acciones internas τ_{diffA} y τ_{diffI} respectivamente, que representan mediante las tasas $diffA$ y $diffI$ la razón de difusión de las señales de la celda a sus vecinos. Cuando estas tasas son alcanzadas se ejecutan múltiples instancias (tantas como vecinos⁴) del subproceso $DiffSigA(difNeigh, act\ Neigh)$ si es activador o $DiffSigI(difNeigh, inh\ Neigh)$ si es inhibidor. Estos procesos usan como parámetros $difNeigh$ como canal emisor para la difusión y su segundo parámetro $act\ Neigh$ para la señal activadora ($inh\ Neigh$ para la inhibidora); con tasa de acuerdo al la regla de mayoría de la sección 2. pA significa “parámetros del proceso Activador”, pdA “parámetros de difusión de A ” y de forma análoga para pI y pdI del proceso Inhibidor. $Neigh$ es un conjunto auxiliar de nombres para denotar la dirección de los vecinos con los que se comunica la celda.

4.4.2.1. Resultados numéricos preliminares del modelo de $m \times n$ celdas.



Figura 4.4.6: 5×5 celdas con inducción central.

Tomando la descripción de nuestro modelo propuesto para la representación espacial del AI simulamos un arreglo de 5×5 celdas con dos distintas condiciones

⁴En nuestro caso con la vecindad de Von Neumann, sólo cuatro, pero es posible extenderlo a una vecindad hexagonal o de n vecinos alrededor de la celda.



Figura 4.4.7: 5×5 celdas con inducción alternada.

iniciales. Los parámetros fueron tomados de los usados en simulaciones estándar de la documentación del simulador. Los datos obtenidos de las simulaciones en el SPiM fueron visualizados espacialmente con la ayuda del software Wolfram Mathematica 7. Los códigos para las simulaciones en el SPiM se encuentran en el apéndice B.

En la primera simulación (Figura 4.4.6) la celda central se activa o induce inicialmente y paulatinamente difunde la información del morfógeno activador a sus vecinas. Luego de cierto tiempo en un radio de 1 a 2 celdas de la celda central, se encuentran activadas y más lejos, 3 a 4 celdas, están virtualmente reprimidas.

La segunda simulación (Figura 4.4.7) consiste en activar dos celdas, la información del morfógeno activador vuelve a fluir con las celdas vecinas hasta que luego de cierto tiempo las celdas cercanas a la inducción forman zonas activadas y las lejanas zonas reprimidas.

Una observación importante de estas simulaciones preliminares es el uso por cada celda de la regla de mayoría. Esta regla está codificada en la descripción interna de cada celda y se puede observar que una celda se llega a activar si el promedio de las vecinas aporta morfógeno dentro de ella.

Es claro que estos resultados muestran cualitativamente el comportamiento del sistema de $m \times n$ celdas, respaldando así nuestro modelo espacial, sin embargo son preliminares debido a que solo se representan 5×5 celdas en el arreglo. Para

poder aproximar resultados experimentales sobre sistemas de reacción difusión es necesario realizar simulaciones en las cuales se incluyan una mayor cantidad de celdas. Además los parámetros necesarios para la aparición de patrones de Turing son muy finos. El estudio del espacio de parámetros es un tema de nuestro interés y es por esto que en la actualidad nos encontramos trabajando en ello en simulaciones con una mayor cantidad de celdas en el arreglo de $m \times n$.

Capítulo 5

Conclusiones.

El cálculo-II es un formalismo matemático muy útil para representar sistemas que se ejecutan al mismo tiempo e interactúan entre sí. Las ventajas que posee son poder expresar intercambio de conexiones entre agentes por la movilidad, se pueden restringir conexiones entre ellos y además es posible decir por medio de la bisimulación si agentes son equivalentes o no.

La principal desventaja para su uso es que es sumamente técnico, para poder trabajar con él plenamente se debe de adquirir una gran cantidad de conocimiento en matemática y las ciencias de la computación.

En términos de modelado matemático en cálculo-II ha sido utilizado para realizar diversos modelos cuantitativos y que logran reproducir fenómenos biológicos con gran exactitud. En comparación con las ecuaciones diferenciales el cálculo-II también predice cualitativa y cualitativamente estos fenómenos.

En términos de redes de regulación es posible describir toda una red biológica de forma accesible y con un costo computacional bajo, es decir, es posible extender la cantidad de genes, moléculas o cofactores involucrados sin que el modelo crezca exponencialmente en complejidad.

Vale la pena mencionar que las extensiones actuales de este formalismo han logrado añadir mucha mayor expresividad en los modelos, por ejemplo modelando interacción entre membranas, lo cual muestra un panorama alentador para su uso y desarrollo futuro.

Como trabajo futuro se plantea desarrollar una extensión del cálculo-II con la cual sea posible modelar fenómenos continuos como la difusión, explorar el

estudio de mecanismos ya conocidos de morfogénesis y describirlos en términos de esta álgebra, analizar el espacio de parámetros en las simulaciones y finalmente establecer un criterio (como las bifurcaciones en ecuaciones diferenciales) en términos de lenguajes para la formación de patrones con cálculo- Π .

Bibliografía

- [1] William F. Ames. *Numerical methods for partial differential equations*. Academic Press, 1977.
- [2] Mariana Benítez, Carlos Espinosa-Soto, Pablo Padilla-Longoria, José Díaz, and Elena R Alvarez-Buylla. Equivalent genetic regulatory networks in different contexts recover contrasting spatial cell patterns that resemble those in arabidopsis root and leaf epidermis: a dynamic model. *Int J Dev Biol*, 51(2):139–155, 2007.
- [3] Jan A. Bergstra and Jan W. Klop. *Algebra of Communicating Processes*, volume 37, pages 89–138. North-Holland, 1986.
- [4] Ralf Blossey, Luca Cardelli, and Andrew Phillips. A compositional approach to the stochastic dynamics of gene networks. In Corrado Priami, Luca Cardelli, and Stephen Emmott, editors, *Transactions on Computational Systems Biology IV*, volume 3939 of *Lecture Notes in Computer Science*, pages 99–122. Springer Berlin / Heidelberg, 2006.
- [5] V. Castets, E. Dulos, J. Boissonade, and P. De Kepper. Experimental evidence of a sustained standing turing-type nonequilibrium chemical pattern. *Phys. Rev. Lett.*, 64(24):2953–2956, Jun 1990.
- [6] Michael B. Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, January 2000.
- [7] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Biological Cybernetics*, 12:30–39, 1972. 10.1007/BF00289234.
- [8] C. C. Guet, Michael B. Elowitz, Weihong Hsing, and Stanislas Leibler. Combinatorial synthesis of genetic networks. *Science*, 296(5572):1466–1470, 2002.
- [9] Charles A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.
- [10] Francois Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3(3):318 – 356, 1961.

- [11] János Z Kelemen, Prasuna Ratna, Simone Scherrer, and Attila Becskei. Spatial epigenetic control of mono- and bistable gene expression. *PLoS Biol*, 8(3):e1000332, Mar 2010.
- [12] Esther M. Lederberg. Lysogenicity in eescherichia coli strain k-12. *Genetics*, 36:560, 1951.
- [13] Robin Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.
- [14] Robin Milner. The polyadic pi-calculus. 1992.
- [15] Robin Milner. *Communicating and mobile systems : the [Pi]-Calculus*. Cambridge University Press, Cambridge, 1999.
- [16] Robin Milner, Joachim Parrow, and David. Walker. *A calculus of mobile processes : Part I & II*. University of Edinburgh, Edinburgh, 1989.
- [17] James D. Murray. *Mathematical biology*. Springer-Verlag, Berlin; New York, 1989.
- [18] John Von Neumann and Arthur W. Burks. *Theory of self-reproducing automata*. University of Illinois Press, Urbana, 1981.
- [19] Carl A. Petri. Communication with automata. *PhD thesis Institut für Instrumentelle Mathematik Bonn Germany*, 15(8):97, 1966.
- [20] Andrew Phillips. The stochastic pi-machine. Available from <http://research.microsoft.com/aphillip/spim/>, 2008.
- [21] Andrew Phillips and Luca Cardelli. A correct abstract machine for the stochastic pi-calculus. In *TCSB VII, 2006, (LNCS, 4230)*, 2006.
- [22] Andrew Phillips, Luca Cardelli, and Giuseppe Castagna. A graphical representation for biological processes in the stochastic pi-calculus. In *Transactions on Computational Systems Biology VII*, volume 4230 of *Lecture Notes in Computer Science*, pages 123–152. Springer Berlin / Heidelberg, 2006.
- [23] Corrado Priami. Stochastic pi-calculus. *The Computer Journal*, 38(7):578–589, 1995.
- [24] Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80(1):25 – 31, 2001.
- [25] Marilyn C. Roberts. Tetracycline resistance determinants: mechanisms of action, regulation of expression, genetic mobility, and distribution. *Microbiol. Rev.*, 19:1–24, 1966.

- [26] Michael B. Smith and Jerry March. *March's Advanced Organic Chemistry*. Wiley, 2001.
- [27] Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London*, 327 B:37–72, 1952.
- [28] Thomas Wilhelm. The smallest chemical reaction system with bistability. *BMC Systems Biology*, 3(1):90, 2009.
- [29] Lewis Wolpert. Positional information and the spatial pattern of cellular differentiation. *Journal of Theoretical Biology*, 25(1):1 – 47, 1969.
- [30] Silvano Dal Zilio. Mobile processes: a commented bibliography. In *MOVEP'2K 4th Summer school on Modelling and Verification of Parallel processes*, volume 2067 of *Lecture Notes in Computer Science*, pages 206–222. Springer-Verlag, 2001.

Apéndice A

Códigos de simulaciones.

Los códigos de simulaciones a continuación fueron tomados íntegramente de la documentación del simulador. Para mayor información y ejemplos adicionales consultar [4].

A.0.3. Simulaciones de reacciones químicas.

A.0.3.1. Decaimiento radiactivo.

```
(* Simulation time, samples, and plotting *)
directive sample 10.0 directive plot X() directive graph
(* Parameter *)
val c = 0.5 (* Molecule definition *)
let X() = delay@c (* Execute model *)
run 1000 of X() (*1000,5000,10000*)
```

A.0.3.2. NaCl.

A.0.3.3. (* Na + Cl <==> Na+ + Cl- *).

```
(* Simulation time, samples, and plotting *)
directive sample 0.03 directive plot Na(); Na_plus() directive graph
(* The circuit *)
new ionize@100.0 : chan new deionize@10.0 : chan
(* Model definition *)
let Na() = !ionize; Na_plus() and Na_plus() = ?deionize; Na()
let Cl() = ?ionize; Cl_minus() and Cl_minus() = !deionize; Cl() (*
```

```

Execute model *)
run (100 of Na() | 100 of Cl())

```

A.0.3.4. Formación Amidas.

```

(* RNH2 + RCOOH <==> RNHCOR + H2O *)

(* Simulation time, samples, and plotting *)
directive sample 0.05 directive plot RNH2(); RNHCOR() directive
graph
(* The circuit *)
new amine@10.0:chan new hydrolysis@1.0:chan

(* Model definition *)
let RNH2() = ?amine; RNHCOR() and RNHCOR() = !hydrolysis;
RNH2()

let RCOOH() = !amine; H2O() and H2O() = ?hydrolysis; RCOOH()

(* Execute model *)
run (100 of RNH2() | 100 of RCOOH())

```

A.0.4. Simulaciones de procesos biológicos.

A.0.4.1. Sistema Biestable.

```

(*Bistable network*)

(* Simulation time, samples, and plotting *)
directive sample 1000.0 directive plot A(); B() directive graph
(* Parameters *)
val tA = 0.20 val dA = 0.002 (*0.0085*) val tB = 0.37 val dB =
0.002 (*0.034*)
(* The circuit *)
new bind@0.72:chan(chan) val dAB = 0.53 new inhibit@0.19:chan(chan)
val unbind = 0.42 val tB' = 0.027

(* Genes definitions *)
let a() = delay@tA; ( A() | a() )
and A() = (
new u@unbind:chan do delay@dA or !bind(u); A_B(u) or !inhibit(u); A_b(u) )

```

```

and A_b(u:chan) = ?u; A() and A_B(u:chan) = delay@dAB
let b() = do delay@tB; ( B() | b() ) or ?inhibit(u); b_A(u)
and b_A(u:chan) = do !u; b() or delay@tB'; ( B() | b_A(u) )
and B() = do ?bind(u); B_A(u) or delay@dB and B_A(u:chan) =
  () (* Execute model *)

run (a() | b())

```

A.0.4.2. Represilador.

```

(* Reppressilator *)

(* Simulation time, samples, and plotting *) directive sample 200000.0
  1000 directive plot Protein(b); Protein(c); Protein(a)
directive graph val t = 0.1 val d = 0.001
(* Parameters *) val u = 0.0001 val bind = 1.0

(*The circuit*) new a@bind:chan new b@bind:chan new c@bind:chan

(*Gene definitions*) let Gene(a:chan,b:chan) = do delay@t; (Pro-
  tein(b) | Gene(a,b)) or ?a; delay@u; Gene(a,b)
and Protein(b:chan) = do !b; Protein(b) or delay@d

(*Execute model*)run ( Gene(a,b) | Gene(b,c) | Gene(c,a) )

```

A.0.5. Simulaciones de compuertas lógicas.

A.0.5.1. Compuertas negativas.

```

(*Negative logic gates*)

(* Simulation time, samples, and plotting *)
directive sample 90000.0 500 directive plot !a as "a"; !b as "b"; !c as
  "c"
(* Parameters *)
val dk = 0.001 val inh = 0.001 val cst = 0.1 val bnd = 1.0

(* Transcription factor *)
let tr(p:chan()) = do !p; tr(p) or delay@dk

(* Neg gate *)
let neg(a:chan(), b:chan()) = do ?a; delay@inh; neg(a,b) or delay@cst;
  (tr(b) | neg(a,b))

```



```
(* The circuit *)  
new a @ bnd: chan() new b @ bnd: chan() new c @ bnd: chan()  
run (neg(c,a) | neg(a,b) | neg(b,c))
```

Apéndice B

Códigos de simulaciones del activador-inhibidor

B.1. Código del *AI* de una celda.

```
directive sample 300000.0 10000
directive plot !inh as "Inhibidor"; !act as "Activador"
directive graph
(*Parameters*)
val bl = 0.001 val bA = 0.001 val tA= 1.0
val tl= 0.4 val dA = 0.001 val dl = 0.001
val inh = 0.3 val ind = 1.0 val ind' = 1.0
val unb = 0.1
(*Definition of channels for communication*)
new inh@inh:chan new act@ind:chan new act'@ind':chan
(*Activator definition*)
let Activator()= do ?inh();delay@unb;Activator()
or delay@bA;(SignA()|SignA'())|Activator()
or ?act;delay@tA;(SignA()|SignA'())|Activator()
(*Activator signals*)
and SignA()= do !act;SignA() or delay@dA
and SignA'()= do !act';SignA'() or delay@dA
(*Inhibitor definition*)
let Inhibitor()= do delay@bl;(SignI())|Inhibitor()
or ?act';delay@tl;(SignI())|Inhibitor()
(*Inhibitor signal*)
and SignI()= do !inh();SignI() or delay@dl
(*System definition*)
run(Activator())|Inhibitor()
```

B.2. Código del *AI* de $m \times n$ celdas.

```
(*Activador-Inhibidor & SPi. Modelo mxn*)
directive sample 10000.0 1000
directive graph
directive plot
!act1; !act2; !act3; !act4; !act5; !act6; !act7; !act8; !act9; !act10; !act11; !act12; !act13; !act14; !act15;
!act16; !act17; !act18; !act19; !act20; !act21; !act22; !act23; !act24; !act25; !inh1; !inh2; !inh3; !inh4; !inh5;
!inh6; !inh7; !inh8; !inh9; !inh10; !inh11; !inh12; !inh13; !inh14; !inh15; !inh16; !inh17; !inh18; !inh19; !inh20;
!inh21; !inh22; !inh23; !inh24; !inh25
```

APÉNDICE B. CÓDIGOS DE SIMULACIONES DEL ACTIVADOR-INHIBIDOR 63

```

(*Parameters*)
val basA=0.001 val basI=0.001 val transA=1.0 val transl=0.4 val degIntSigA= 0.001 val degIntSigI=
0.002
val unb= 0.1 val difrate= 0.2 val int_A=0.9 val int_I=0.4 val diffA=0.2 val diffI=0.3 val degDiff=
0.001

(*Definition of channels for communication*)
new act1@int_A:chan new act2@int_A:chan new act3@int_A:chan new act4@int_A:chan new act5@int_A:chan
new act6@int_A:chan new act7@int_A:chan new act8@int_A:chan new act9@int_A:chan new act10@int_A:chan
new act11@int_A:chan new act12@int_A:chan new act13@int_A:chan new act14@int_A:chan new act15@int_A:chan
new act16@int_A:chan new act17@int_A:chan new act18@int_A:chan new act19@int_A:chan new act20@int_A:chan
new act21@int_A:chan new act22@int_A:chan new act23@int_A:chan new act24@int_A:chan new act25@int_A:chan
new actr:chan

new inh1@int_I:chan new inh2@int_I:chan new inh3@int_I:chan new inh4@int_I:chan new inh5@int_I:chan
new inh6@int_I:chan new inh7@int_I:chan new inh8@int_I:chan new inh9@int_I:chan new inh10@int_I:chan
new inh11@int_I:chan new inh12@int_I:chan new inh13@int_I:chan new inh14@int_I:chan new inh15@int_I:chan
new inh16@int_I:chan new inh17@int_I:chan new inh18@int_I:chan new inh19@int_I:chan new inh20@int_I:chan
new inh21@int_I:chan new inh22@int_I:chan new inh23@int_I:chan new inh24@int_I:chan new inh25@int_I:chan
new inhr:chan

new dif1@difrate:chan(chan) new dif2@difrate:chan(chan) new dif3@difrate:chan(chan) new dif4@difrate:chan(chan)
new dif5@difrate:chan(chan) new dif6@difrate:chan(chan) new dif7@difrate:chan(chan) new dif8@difrate:chan(chan)
new dif9@difrate:chan(chan) new dif10@difrate:chan(chan) new dif11@difrate:chan(chan) new dif12@difrate:chan(chan)
new dif13@difrate:chan(chan) new dif14@difrate:chan(chan) new dif15@difrate:chan(chan) new dif16@difrate:chan(chan)
new dif17@difrate:chan(chan) new dif18@difrate:chan(chan) new dif19@difrate:chan(chan) new dif20@difrate:chan(chan)
new dif21@difrate:chan(chan) new dif22@difrate:chan(chan) new dif23@difrate:chan(chan) new dif24@difrate:chan(chan)
new dif25@difrate:chan(chan)
new difr:chan(chan)

(*Activator definition*)
let Activador((act_ij:chan,inh_ij:chan),dif_ij:chan(chan), (difN:chan(chan), difS:chan(chan), difE:chan(chan),
difW:chan(chan)), (actN:chan, actS:chan, actE:chan, actW:chan))=
do delay@basA; (Act(act_ij, (difN,difS,difE,difW), (actN,actS,actE,actW)) | Activador((act_ij,inh_ij),
dif_ij, (difN,difS,difE,difW), (actN,actS,actE,actW)))
or ?act_ij; delay@transA; (Act(act_ij,(difN,difS,difE,difW), (actN,actS,actE,actW)) | Activador((act_ij,
inh_ij), dif_ij, (difN,difS,difE,difW), (actN,actS,actE,actW)))
or ?inh_ij; delay@unb; Activador((act_ij,inh_ij), dif_ij, (difN,difS,difE,difW), (actN,actS,actE,actW))
or ?dif_ij(com);(Com(com)|Activador((act_ij,inh_ij),dif_ij,(difN,difS,difE,difW),(actN,actS,actE,actW)))
(*Auxiliary Activator process*)
and Act(act_ij:chan, (difN:chan(chan), difS:chan(chan), difE:chan(chan), difW:chan(chan)), (actN:chan,
actS:chan, actE:chan, actW:chan))=
(IntSigA(act_ij) | DiffA((difN,difS,difE,difW), (actN,actS,actE,actW)))
(*Activator diffusible signals for the Von Neumann Neighborhood*)
and DiffA((difN:chan(chan), difS:chan(chan), difE:chan(chan), difW:chan(chan)), (actN:chan, actS:chan,
actE:chan, actW:chan))=
delay@diffA; (DiffSigA(difN,actN) | DiffSigA(difS,actS) | DiffSigA(difE,actE) | DiffSigA(difW,actW))
(*Channel for external effect in the ij cell*)
and Com(com:chan)= do !com;Com(com) or delay@degDiff
(*Activator internal signals*)
and IntSigA(act_ij:chan)= do !act_ij;IntSigA(act_ij) or delay@degIntSigA
(*Auxiliar diffusible process*)
and DiffSigA(difNeigh:chan(chan),actNeigh:chan)=
!difNeigh(actNeigh);DiffSigA(difNeigh,actNeigh)

(*Inhibitor definition*)
let Inhibidor((act_ij:chan,inh_ij:chan), (difN:chan(chan),difS:chan(chan),difE:chan(chan), difW:chan(chan)),
(inhN:chan, inhS:chan, inhE:chan, inhW:chan))=
do delay@basI; (Inh(inh_ij, (difN,difS,difE,difW), (inhN,inhS,inhE,inhW)) | Inhibidor((act_ij,inh_ij),
(difN,difS,difE,difW), (inhN,inhS,inhE,inhW)))
or ?act_ij; delay@transI; (Inh(inh_ij, (difN,difS,difE,difW), (inhN,inhS,inhE,inhW)) | Inhibidor((act_ij,inh_ij),
(difN,difS,difE,difW), (inhN,inhS,inhE,inhW)))
(*Auxiliary Inhibitor process*)
and Inh(inh_ij:chan, (difN:chan(chan), difS:chan(chan), difE:chan(chan), difW:chan(chan)), (inhN:chan,
inhS:chan, inhE:chan, inhW:chan))=
(IntSigI(inh_ij) | DiffI((difN,difS,difE,difW),(inhN,inhS,inhE,inhW)))
(*Inhibitor diffusible signals for the Von Neumann Neighborhood*)
and DiffI((difN:chan(chan), difS:chan(chan), difE:chan(chan), difW:chan(chan)), (inhN:chan, inhS:chan,
inhE:chan, inhW:chan))=

```

APÉNDICE B. CÓDIGOS DE SIMULACIONES DEL ACTIVADOR-INHIBIDOR 64

```

delay@diffI;(DiffSigI(difN,inhN)|DiffSigI(difS,inhS)|DiffSigI(difE,inhE)|DiffSigI(difW,inhW))
(*Inhibitor internal signal*)
and IntSigI(inh_ij:chan)= do !inh_ij;IntSigI(inh_ij) or delay@degIntSigI
(*Auxiliar diffusible process*)
and DiffSigI(difNeigh:chan(chan),inhNeigh:chan)=
!difNeigh(inhNeigh);DiffSigI(difNeigh,inhNeigh)

(*Specification of the interconnectivity for the 5x5 array, i.e. Activator and Inhibitor
names for identifying the ij-cell and the neighbors for interaction in the array*)
run(
  Activador((act1,inh1), dif1,(difr,dif6,dif2,difr), (actr,act6,act2,actr)) | Inhibidor((act1,inh1), (difr,dif6,dif2,difr),
(inhr,inh6,inh2,inh8))
  | Activador((act2,inh2), dif2,(difr,dif7,dif3,dif1), (actr,act7,act3,act1)) | Inhibidor((act2,inh2), (difr,dif7,dif3,dif1),
(inhr,inh7,inh3,inh1))
  | Activador((act3,inh3), dif3,(difr,dif8,dif4,dif2), (actr,act8,act4,act2)) | Inhibidor((act3,inh3), (difr,dif8,dif4,dif2),
(inhr,inh8,inh4,inh2))
  | Activador((act4,inh4), dif4,(difr,dif9,dif5,dif3), (actr,act9,act5,act3)) | Inhibidor((act4,inh4), (difr,dif9,dif5,dif3),
(inhr,inh9,inh5,inh3))
  | Activador((act5,inh5), dif5,(difr,dif10,difr,dif4), (actr,act10,actr,act4)) | Inhibidor((act5,inh5), (difr,dif10,difr,dif4),
(inhr,inh10,inh4))
  | Activador((act6,inh6), dif6,(dif1,dif11,dif7,difr), (act1,act11,act7,actr)) | Inhibidor((act6,inh6), (dif1,dif11,dif7,difr),
(inh1,inh11,inh7,inh8))
  | Activador((act7,inh7), dif7,(dif2,dif12,dif8,dif6), (act2,act12,act8,act6)) | Inhibidor((act7,inh7), (dif2,dif12,dif8,dif6),
(inh2,inh12,inh8,inh6))
  | Activador((act8,inh8), dif8,(dif3,dif13,dif9,dif7), (act3,act13,act9,act7)) | Inhibidor((act8,inh8), (dif3,dif13,dif9,dif7),
(inh3,inh13,inh9,inh7))
  (*Uncomment next line for induction of the 9 cell in the spatial simulation 2*)
  (* | 500 of Act(act9,(dif4,dif14,dif10,dif8), (act4,act14,act10,act8)) *)
  | Activador((act9,inh9), dif9,(dif4,dif14,dif10,dif8), (act4,act14,act10,act8)) | Inhibidor((act9,inh9), (dif4,dif14,dif10,dif8),
(inh4,inh14,inh10,inh8))
  | Activador((act10,inh10), dif10,(dif5,dif15,difr,dif9), (act5,act15,actr,act9)) | Inhibidor((act10,inh10),
(dif5,dif15,difr,dif9), (inh5,inh15,inh8,inh9))
  | Activador((act11,inh11), dif11,(dif6,dif16,dif12,difr), (act6,act16,act12,actr)) | Inhibidor((act11,inh11),
(dif6,dif16,dif12,difr), (inh6,inh16,inh12,inh8))
  | Activador((act12,inh12), dif12,(dif7,dif17,dif13,dif11), (act7,act17,act13,act11)) | Inhibidor((act12,inh12),
(dif7,dif17,dif13,dif11), (inh7,inh17,inh13,inh11))
  (*Uncomment next line for induction of the 13 cell in the spatial simulation 1*)
  (* | 500 of Act(act13,(dif8,dif18,dif14,dif12), (act8,act18,act14,act12)) *)
  | Activador((act13,inh13), dif13,(dif8,dif18,dif14,dif12), (act8,act18,act14,act12)) | Inhibidor((act13,inh13),
(dif8,dif18,dif14,dif12), (inh8,inh18,inh14,inh12))
  | Activador((act14,inh14), dif14,(dif9,dif19,dif15,dif13), (act9,act19,act15,act13)) | Inhibidor((act14,inh14),
(dif9,dif19,dif15,dif13), (inh9,inh19,inh15,inh13))
  | Activador((act15,inh15), dif15,(dif10,dif20,difr,dif14), (act10,act20,actr,act14)) | Inhibidor((act15,inh15),
(dif10,dif20,difr,dif14), (inh10,inh20,inh8,inh14))
  | Activador((act16,inh16), dif16,(dif11,dif21,dif17,difr), (act11,act21,act17,actr)) | Inhibidor((act16,inh16),
(dif11,dif21,dif17,difr), (inh11,inh21,inh17,inh8))
  (*Uncomment next line for induction of the 17 cell in the spatial simulation 2*)
  (* | 500 of Act(act17,(dif12,dif22,dif18,dif16), (act12,act22,act18,act16)) *)
  | Activador((act17,inh17), dif17,(dif12,dif22,dif18,dif16), (act12,act22,act18,act16)) | Inhibidor((act17,inh17),
(dif12,dif22,dif18,dif16), (inh12,inh22,inh18,inh16))
  | Activador((act18,inh18), dif18,(dif13,dif23,dif19,dif17), (act13,act23,act19,act17)) | Inhibidor((act18,inh18),
(dif13,dif23,dif19,dif17), (inh13,inh23,inh19,inh17))
  | Activador((act19,inh19), dif19,(dif14,dif24,dif20,dif18), (act14,act24,act20,act18)) | Inhibidor((act19,inh19),
(dif14,dif24,dif20,dif18), (inh14,inh24,inh20,inh18))
  | Activador((act20,inh20), dif20,(dif15,dif25,difr,dif19), (act15,act25,actr,act19)) | Inhibidor((act20,inh20),
(dif15,dif25,difr,dif19), (inh15,inh25,inh8,inh19))
  | Activador((act21,inh21), dif21,(dif16,difr,dif22,difr), (act16,actr,act22,actr)) | Inhibidor((act21,inh21),
(dif16,difr,dif22,difr), (inh16,inh22,inh8,inh19))
  | Activador((act22,inh22), dif22,(dif17,difr,dif23,dif21), (act17,actr,act23,act21)) | Inhibidor((act22,inh22),
(dif17,difr,dif23,dif21), (inh17,inh23,inh8,inh19))
  | Activador((act23,inh23), dif23,(dif18,difr,dif24,dif22), (act18,actr,act24,act22)) | Inhibidor((act23,inh23),
(dif18,difr,dif24,dif22), (inh18,inh24,inh8,inh19))
  | Activador((act24,inh24), dif24,(dif19,difr,dif25,dif23), (act19,actr,act25,act23)) | Inhibidor((act24,inh24),
(dif19,difr,dif25,dif23), (inh19,inh25,inh8,inh19))
  | Activador((act25,inh25), dif25,(dif20,difr,difr,dif24), (act20,actr,actr,act24)) | Inhibidor((act25,inh25),
(dif20,difr,difr,dif24), (inh20,inh8,inh8,inh19))
)

```