APLICACIÓN DE LA PROGRAMACIÓN CONCURRENTE POR RESTRICCIONES EN EL MODELADO DE PROCESOS EN LA MEMBRANA CELULAR.

Jaime Arley Muñoz Franco
Henry Andrés Pérez Maturana

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
SANTIAGO DE CALI
2009

APLICACIÓN DE LA PROGRAMACIÓN CONCURRENTE POR RESTRICCIONES EN EL MODELADO DE PROCESOS EN LA MEMBRANA CELULAR.

Jaime Arley Muñoz Franco
Henry Andrés Pérez Maturana

Tesis de grado para optar al título de
Ingeniero de Sistemas y Computación

Director
CAMILO RUEDA CALDERÓN
Ingeniero de Sistemas y computación

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
SANTIAGO DE CALI
2009

Santiago de Cali, Julio 13 de 2009

Ingeniera
MARÍA CONSTANZA PABÓN BURBANO
Directora de la Carrera de Ingeniería de Sistemas y Computación
Pontificia Universidad Javeriana
Ciudad

Certifico que el presente proyecto de grado, titulado "APLICACIÓN DE LA PROGRAMACIÓN CONCURRENTE POR RESTRICCIONES EN EL MODELADO DE PROCESOS EN LA MEMBRANA CELULAR." realizado por Jaime Arley Muñoz Franco y Henry Andrés Pérez Maturana, estudiantes de Ingeniería de Sistemas y Computación, se encuentra terminado y puede ser presentado para sustentación.

Atentamente,

Dr. CAMILO RUEDA CALDERÓN
Director del Proyecto

Santiago de Cali, Julio 13 de 2009

Ingeniera
MARÍA CONSTANZA PABÓN BURBANO
Directora de la Carrera de Ingeniería de Sistemas y Computación
Pontificia Universidad Javeriana
Ciudad

Por medio de ésta, presentamos a usted el proyecto de grado titulado "APLICACIÓN DE LA PROGRAMACIÓN CONCURRENTE POR RESTRICCIONES EN EL MODELADO DE PROCESOS EN LA MEMBRANA CELULAR." para optar al título de Ingeniero de Sistemas y Computación.

Esperamos que este proyecto reuna todos los requisitos académicos y cumpla el propósito para el cual fue creado, y sirva de apoyo para futuros proyectos en la Universidad Javeriana relacionados con la materia.

Atentamente,

Jaime Arley Muñoz Franco Henry Andrés Pérez Maturana

ARTICULO 23 de la Resolución No 13 del 6 de Julio de 1946
del Reglamento de la Pontificia Universidad Javeriana.

"La Universidad no se hace responsable por los conceptos emitidos por sus alumnos en sus trabajos de Tesis. Sólo velará porque no se publique nada contrario al dogma y a la moral Católica y porque las Tesis no contengan ataques o polémicas puramente personales; antes bien, se vea en ellas el anhelo de buscar la Verdad y la Justicia"

Nota de Aceptación:

Aprobado por el comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y Computación.

JORGE FRANCISCO ESTELA URIBE
Decano Académico de la Facultad de Ingeniería

MARÍA CONSTANZA PABÓN BURBANO
Director de la Carrera de Ingeniería de Sistemas y Computación

CAMILO RUEDA CALDERÓN
Director de Tesis

Jurado

Jurado

A Dios y a nuestros padres por habernos ayudado a terminar otra etapa importante en nuestras vidas, simplemente gracias por su apoyo incondicional.

Agradecimientos

El más profundo agradecimiento a todas y cada una de aquellas personas que siempre estuvieron con nosotros y ayudaron de una u otra forma en la realización de este trabajo de grado, que pone fin a un largo camino de estudio, sacrificios, alegrías y tristezas, pero a su vez se abren nuevas puertas para próximos logros personales y profesionales.

- De manera especial a nuestro director de tesis Camilo Rueda quien nos guió y orientó a lo largo de este tiempo en la elaboración de este trabajo de grado, así como fue un gran apoyo cuando las cosas se ponían difíciles.
- Un especial agradecimiento a Carlos Olarte, quien desde un principio estuvo interesado en este trabajo de grado, sus comentarios y sugerencias mejoraron cada capítulo, fue no solo un asesor sino un amigo quien nos orientó a nivel personal y académico en el desarrollo de este trabajo de grado, Carlos ha sido testigo de la evolución de este trabajo de grado y los diferentes giros que tomó. De verdad Carlos que no existen palabras para agradecerte todo el apoyo brindado. Gracias Totales!
- A nuestro compañero Víctor Rivera, quien fue siempre un apoyo y estuvo en los momentos más críticos de este trabajo. Gracias por toda tu ayuda.
- A nuestras familias y amigos que fueron un apoyo constante durante este tiempo, muchas gracias.
- Queremos expresar nuestros agradecimientos a los miembros de Avispa, especialmente a Camilo Rueda, Carlos Olarte y Frank Valencia por sus constantes sugerencias y comentarios en cada presentación de nuestro proyecto, sus observaciones mejoraron notablemente los resultados teóricos y el modelo propuesto en esta tesis de grado.

- A la Pontificia Universidad Javeriana - Cali por brindarnos la oportunidad de formarnos académica y profesionalmente dentro de esta institución.

Nuevamente a todos los que de una u otra forma influenciaron en este trabajo de grado. Muchas Gracias!!

Índice general

Índice de cuadros xiv

Índice de figuras xv

Resumen xvii

Abstract xix

1. Introducción **23**

1.1. Motivación 23

1.2. Objetivos y Alcances 27

1.3. Contribuciones 29

1.4. Estructura del documento 30

2. Fundamentos Biológicos y Computacionales **32**

2.1. Cálculos de Procesos 32

2.2. Programación Concurrente por Restricciones 33

2.2.1. Sistema de Restricciones 34

2.3. Ntcc 35

2.3.1. Sintaxis de Ntcc 36

2.3.2. Semántica Operacional 36
2.4. Movilidad en el Cálculo π 37
2.4.1. Sintaxis del Cálculo π 38
2.5. Utcc 38
2.5.1. Sintaxis Utcc 39
2.5.2. Movilidad en Utcc 40
2.5.3. Semántica operacional de Utcc 41
2.6. Fundamentación Biológica 43
2.6.1. La Membrana como Estructura Dinámica 43
2.7. Procesos a Nivel de Membrana Celular: Endocitosis y Exocitosis . 45
2.7.1. Endocitosis 45
2.7.2. Exocitosis 46
2.8. Biología y Computación 46
2.9. Resumen 48

3. Cálculos Biológicos Inspirados en Interacciones entre Membranas 50

3.1. Modelo de computación con membranas 50
3.2. BioAmbients 51
3.3. Cálculo Brane 53
3.3.1. Sintaxis 55
3.3.2. Interacciones Básicas 57
3.3.3. Interacciones Derivadas 61
3.4. Resumen 68

4. Interpretación de Cálculos Biológicos en CCP **69**

4.1. Descripción de la Interpretación de Membranas 69

4.2. Descripción del Sistema de Restricciones 72

4.3. Interpretación del Cálculo Brane en Utcc 77

4.3.1. Traducciones Básicas 78

4.4. Interpretación de Interacciones Básicas entre Membranas: Fagocitosis, Exocitosis y Pinocitosis 81

4.4.1. Comportamiento Natural de una Membrana: CASE 83

4.4.2. Interpretación del Proceso de Fagocitosis 84

4.4.3. Interpretación del Proceso de Exocitosis 88

4.4.4. Interpretación del Proceso de Pinocitosis 90

4.5. Persistencia de la Información a lo largo del Tiempo 92

4.6. Secuencia de Acciones: Interacciones Derivadas 94

4.7. Discusiones Sobre la Interpretación en Utcc 97

4.8. Resumen 98

5. Conclusiones y Trabajo Futuro **100**

5.1. CCP en el Modelado de Sistemas Biológicos 100

5.1.1. Sistemas Concurrentes 100

5.1.2. Noción de Tiempo y la Membrana Celular como Sistema Dinámico 101

5.1.3. Almacén de Restricciones y Movilidad 102

5.2. Ventajas y Desventajas en el uso de CCP para el Modelado de Membranas Celulares 103

5.3. Trabajo Futuro 106

Bibliografía **107**

Índice de cuadros

2.1. Sintaxis de Ntcc 36

2.2. Sintaxis cálculo π 38

2.3. Sintaxis de Utcc 39

3.1. Sintaxis Brane 55

3.2. Congruencia Estructural Brane 56

3.3. Brane: Reacciones Básicas 56

3.4. Mate: acción y co-acción 64

3.5. Budding: acción y co-acción 66

3.6. Drip: acción y co-acción 67

4.1. Definición General CASE 83

Índice de figuras

2.1. Almacén de Restricciones (Store) 34
2.2. Semántica Operacional de Ntcc 37
2.3. Semántica Operacional de Utcc 42
2.4. Estructura de la membrana celular. 43
2.5. Flujo dinámico de la membrana a través de la membrana 44
2.6. Fagocitosis. 45
2.7. Exocitosis. 46

3.1. Ambientes en sintaxis BioAmbient 52
3.2. Interacciones en las Membranas 54
3.3. Acción y Coacción en interacciones básicas entre membranas. . . . 58
3.4. Fagocitosis 59
3.5. Exocitosis 60
3.6. Pinocitosis 61
3.7. Budding (Bud) y Dripping (Drip). 62
3.8. Interacciones derivadas 62
3.9. Interacción derivada: Mate 64

3.10. Interacciones derivadas 65

3.11. Interacción derivada: Dripping 67

4.1. Exocitosis entre σ y τ 70

4.2. Alcances en un sistemas conformado por membranas 71

4.3. Relación entre membrana σ y conjunto asociado S_σ. 73

Resumen

El análisis y estudio de diversos sistemas biológicos se hace cada vez un proceso más frecuente dentro de la comunidad científica, quienes buscan constantemente la exploración e interpretación de datos biológicos, para poder realizar análisis, predecir situaciones y estudiar en detalle comportamientos que ocurren a nivel biológico. Este hecho ha motivado a que computacionalmente también se busquen herramientas que complementen este estudio, inspirando de esta manera nuevos paradigmas de programación que permitan describir y modelar el funcionamiento de dichos sistemas.

Computacionalmente una de las alternativas más nombradas en la actualidad es el uso de Cálculos de Procesos basados en un modelo de programación concurrente por restricciones (CCP) [Sar93] , donde es posible establecer abstracciones entre los componentes reales de los sistemas y los elementos básicos del cálculo, logrando modelar información parcial a través de restricciones que se acumulan en un almacén que contiene toda la información del sistema (Store). Particularmente se utilizará el cálculo Utcc (Universal Timed Concurrent Constraint) [COV07] el cual es un lenguaje derivado de TCC (extensión de CCP) adicionando nuevas construcciones para la abstracción de procesos permitiéndole al cálculo representar la noción de movilidad.

En este trabajo abordamos el problema del modelo de Computación Celular con membranas [dJPJC06] uno de los principales problemas en bioinformática donde se estudian los mecanismos por los cuales la célula se comunica con su exterior a través de la membrana celular.

En el documento se explican dos cálculos centrados en el modelado de este tipo de procesos biológicos inspirados en la interacción entre membranas. El primero de

ellos es BioAmbients [AREPWSS04] el cual proviene del Calculo Ambient [CG98] y por lo tanto conserva sus principales características permitiendo la creación de procesos locales dentro de ambientes, definiendo de esta manera la configuración básica de una membrana. El segundo es Brane Calculi [Car98] en donde se define un conjunto de primitivas inspiradas biológicamente en la interacción entre membranas, alcanzando una fidelidad a la realidad biológica modelada en el cálculo.

En este trabajo de grado se utiliza un enfoque basado en la sintaxis, funcionalidad y reacciones modeladas en el cálculo Brane. Dicho cálculo es traducido como procesos Utcc, generando de esta manera un encoding que a través de procesos previamente establecidos permita traducir un proceso que se encuentre Brane a un proceso en Utcc que simula su comportamiento. Eventualmente se usan técnicas de razonamiento en CCP para analizar los comportamientos de los sistemas en Brane.

Finalmente se presentan los resultados obtenidos, junto con un análisis de los puntos más importantes del presente trabajo y se establece una serie de conclusiones que describen las principales ventajas y desventajas al definir un encoding entre los cálculos, así como también se proponen algunas orientaciones para trabajo futuro. Se detalla y se analizan las principales contribuciones de UTCC como un cálculo que permita modelar sistemas biológicos inspirados en interacción entre membranas, así como sus principales falencias. En este trabajo de grado los autores consideran que las conclusiones y los resultados obtenidos motivan e incentivan al estudio y el análisis de modelos de programación basados en CCP que permitan modelar cada vez sistemas biológicos más complejos.

Abstract

The functional behavior of biological systems has become a task of increasing interest to the scientific community of biologists, who constantly search to infer this behavior from biological data, in order to analyze, to forecast some situations and to do detailed studies about some behaviors which occur at biological level. This fact has also led to the exploration of computational tools that complement this study, inspiring new programming paradigms to describe and model the operation of these biological systems.

Computationally one of the alternatives in the current time is the use of Process Caluli based on a concurrent constraint programming model (CCP), where it is possible to model abstractions between the components of the systems and the basics elements of the calculus, allowing to model partial information through formalisms based on constraints in a store where all information about the system is kept. Particularly in this dissertation the Utcc Calculus (Universal Timed Concurrent Constraint) will be used as a language derived from TCC (extension of CCP) by adding new constructions like the abstraction of processes allowing the calculus to represent the notion of mobility.

This work is mainly focused on the "Mobile Computing with membranes", one of the main problems in bioinformatics, where the mechanisms which the cell communicates with the outside through the cell membrane are studied.

The document outlines two calculi focused on the modeling of these kinds of biological processes based on the interaction between membranes, the first one is BioAmbients derived from Ambient Calculus, allowing the creation of local processes within environments, thus defining the basic configuration of a membrane. The second one is "Brane calculi"which defines a set of biologically inspired primi-

tives in the interaction between membranes, with a fidelity to the biological reality modeled in the calculus. Then a detailed approach to the syntax, functionality and reactions modeled in Brane Calculi will be presented and then an interpretation of the processes in Brane Calculi to Utcc, generating an *encoding* which allows to translate processes in Brane Calculi to equivalent processes in Utcc.

Finally the results obtained are presented, with an analysis of the most important points of this work and establishing a set of conclusions that describe the main advantages and disadvantages to define an interpretation of the calculi, and also to suggest some guidelines for future work. The main contributions of Utcc to modeling biological membrane systems will be analyzed. The last part is intended to work through the findings and results, motivate and encourage the study and analysis of programming models based on CCP to model more complex biological systems

1 Introducción

En este trabajo de grado se estudia el uso de formalismos matemáticos diseñados para el modelado y verificación de sistemas concurrentes basados en el modelo de Programación por Restricciones (CCP) [Sar93]particularmente usando el cálculo Utcc para el modelado de sistemas biológicos. Específicamente se analizan problemas biológicos a un nivel superior (membrana celular), donde se estudian los mecanismos a través de los cuales la célula se comunica con su exterior.

1.1. Motivación

Los cálculos de procesos son formalismos matemáticos y computacionales que permiten describir de forma clara las características fundamentales de un sistema [JGR05]. Dichos sistemas se encuentran a su vez compuestos de otros subsistemas menos complejos generando así una construcción composicional de las especificaciones, que se encuentran basadas fundamentalmente en la descripción de los elementos que las componen y sus interacciones. Esto permite a los cálculos de procesos realizar abstracciones formales que modelen el comportamiento de los sistemas y sus componentes.

El constante avance en el estudio de los Sistemas Biológicos ha permitido que actualmente se hayan logrado notorios avances en el análisis de los componentes que conforman dichos sistemas. Este conjunto de componentes biológicos puede ser visto como un sistema dinámico, en donde los cálculos de procesos son considerados un instrumento de abstracción para analizar su funcionamiento [AG06].Estas características han motivado la exploración y el estudio constante del uso de los cálculos de procesos como una opción viable para describir el funcionamiento de

varios sistemas biológicos.

Dadas todas estas características y ventajas que ofrecen los cálculos de procesos para el modelado de sistemas, es lógico pensar que no se han centrado únicamente en el estudio de procesos biológicos, sino que también han explorado diversos campos, como por ejemplo para modelar Sistemas Reactivos [MNV02a], composición musical [RV05] y protocolos de comunicación segura [AA99], entre otros.

Particularmente, existen algunos Cálculos de procesos que han sido diseados específicamente para el modelado de ciertos sistemas biológicos, como por ejemplo el proceso de Interacciones entre membranas [dJPJC06]. La motivación está dada por la abundancia de interacciones de éste tipo en los procesos celulares, en donde se analiza el comportamiento de las células cuando éstas realizan intercambio de elementos (e.g., aminoácidos, oxigeno, agua) a través de la membrana, que son necesarios para el correcto desarrollo de los procesos celulares. Por lo tanto, se debe tener en cuenta que cualquier tipo de alteración en este tipo de sistemas puede descontrolar el funcionamiento de la célula, incluso al punto de llevarla a la muerte o a la generación de productos nocivos. [AG06]

Para este tipo de procesos Biológicos, como el descrito anteriormente surgen cálculos como Brane Calculi [Car98] que están enfocados precisamente en el proceso de interacción entre membranas, identificando tres tipos de reacciones básicas: fagocitosis, exocitosis y pinocitosis. La inspiración biológica está dada por el proceso de división celular y procesos donde se acepta o se expulsa un agente biológico del interior de la célula. A pesar de la facilidad con que Brane permite realizar modelos de este tipo es poco lo que se ha avanzado en el campo aplicativo, ya que aunque con Brane es posible verificar las propiedades de los sistemas modelados, no existe aún una herramienta que implemente las ideas formales del cálculo. Esto ha motivado otras formas de avanzar en esta dirección utilizando la programación concurrente por restricciones para modelar este tipo de sistemas.

Por consiguiente, en este documento se proponen modelos en CCP para el modelado de sistemas biológicos inspirados en la interacción entre membranas utilizando el cálculo Utcc, tomando como base las interacciones ya definidas inicialmente por el cálculo Brane. Es decir se plantea una interpretación del cálculo Brane en Utcc, logrando codificar el cálculo a través de procesos previamente establecidos que

permitan una traducción para un sistema en Brane, obteniendo como resultado procesos en Utcc que simulen los proceos en Brane.

En ocasiones la descripción de ciertos procesos biológicos es tan específica que se requiere de un cálculo de procesos especializado para realizar abstracciones sobre este tipo de sistemas como es precisamente el sistema que estamos analizando. Por tal motivo el objetivo de realizar esta interpretación, es analizar este tipo de modelos biológicos mediante la programación concurrente por restricciones, específicamente usando el cálculo Utcc que dispone de ciertas características interesantes para describir, simular y verificar propiedades de sistemas biológicos complejos. Entre estas características propias del cálculo podemos destacar: el concepto de movilidad como extensión de tcc, permitiendo la comunicación de nombres privados entre procesos. Representando así el alcance de sistemas a las membranas, la noción explícita del tiempo para describir la evolución de sistemas biológicos dinámicos, el uso de agentes concurrentes o procesos que interactúan entre sí, sincronizando información a través de un almacén de restricciones permitiendo modelar entidades biológicas y las ventajas derivadas de los fundamentos del modelo de programación concurrente por restricciones para expresar propiedades del sistema a modelar. Adicionalmente esta representación a través de un modelo de Programación Concurrentes por Restricciones, puede ser llevada a la práctica siguiendo un enfoque de programación declarativo mediantes lenguajes de programación como Mozart [Smo95] o a través del uso de librerías genéricas que implementan el modelo CCP como GECODE [Sch06] para observar y analizar el comportamiento del modelo propuesto.

Cabe resaltar que este trabajo de grado hace parte de un proyecto del grupo AVISPA (Grupo de Investigación en Ambientes Visuales de Programación Aplicativa); uno de cuyos objetivos es la aplicación del modelo de Programación Concurrente por Restricciones en la representación de sistemas biológicos. Por tal motivo los resultados de esta investigación se encuentran influenciados por las decisiones de los miembros del grupo, quienes con su conocimiento y aportes ayudaron constantemente a encaminar este estudio. El objetivo general expresado en la propuesta del trabajo de grado era: Aplicar la teoría y herramientas de la programación concurrente por restricciones para modelar procesos biológicos a nivel de membrana celular. Los respectivos objetivos específicos eran los siguientes:

1. Modelar con Programación Concurrente por Restricciones el proceso de traspaso de proteínas en la membrana celular.

2. Representar el Modelo matemático basado en Ecuaciones Diferenciales del proceso a nivel de membrana celular con sistemas continuos en Ntcc.

3. Implementar el modelo del proceso de traspaso de proteínas en la membrana para comparar resultados con otros modelos propuestos.

El proyecto de grado se concebía entonces como un trabajo de aplicación de un cálculo de procesos concurrente conocido (Ntcc) al modelamiento de un sistema biológico particular. Tales sistemas son el objeto específico de estudio para el cálculo Brane propuesto por Cardelli y su grupo. Se decidió entonces como punto de partida buscar representar en Ntcc [MNV02b] las operaciones básicas de la membrana celular, según ellas se definen en el cálculo Brane. Habiendo encontrado bastantes dificultades en esta representación, los autores decidieron presentar sus ideas en el coloquio internacional Forces, del grupo Avispa. Fruto de las discusiones en este coloquio, el grupo Avispa determinó que era necesario replantear los objetivos del trabajo. La razón primordial fueron los recientes resultados obtenidos por Carlos Olarte sobre limitaciones de expresividad del cálculo Ntcc. En especial la imposibilidad de modelar procesos móviles que, según se desprendía de las conclusiones del coloquio, iba a ser indispensable para representar sistemas celulares cuya estructura jerárquica cambia dinámicamente como resultado de las operaciones biológicas. Los autores aceptaron entonces la sugerencia de tomar como base el nuevo cálculo Utcc [Ola09], propuesto por Carlos Olarte para agregar la posibilidad de expresar movilidad en cálculos de procesos concurrentes de restricciones.

Para los autores, un resultado interesante, aunque negativo, del trabajo de grado en ese momento fue entonces la imposibilidad de modelar en Ntcc las operaciones básicas sobre membranas celulares. Tomar como nueva base el cálculo Utcc ofrecía mejores perspectivas, pero mantenía la incertidumbre de saber si el esquema de movilidad de Utcc sería suficiente para modelar de manera conveniente las operaciones celulares, en particular el cambiante alcance de los efectos de sus interacciones. El grupo decidió entonces que el trabajo debería enfrentar este

problema como su objetivo básico. Concretamente, saber si era posible o no una representación formal de las operaciones del cálculo Brane en Utcc. Por *formal* se entiende en la literatura de cálculos de procesos el encontrar una codificación exacta de los componentes y operaciones de uno en los del otro.

El otro objetivo específico de la propuesta que hubo que descartar fue la implementación. Por una parte, el grupo Avispa determinó que el trabajo teórico era ya de por sí suficientemente complejo para una tesis de pregrado y, por otra, para Utcc no se había definido en el momento un intérprete, etapa previa indispensable a cualquier implementación.

Los nuevos objetivos son entonces los que se exponen en la siguiente sección.

1.2. Objetivos y Alcances

Este trabajo de grado presenta los siguientes objetivos y alcances:

- **Objetivo General**
  Aplicar la teoría y herramientas de la programación concurrente por restricciones para modelar procesos biológicos a nivel de la membrana celular.
- **Objetivos Específicos**
  1. Modelar por medio de CCP diferentes procesos que ocurren en la membrana celular.
  2. Codificar en Utcc los sistemas y procesos propuestos en Brane Calculi.
  3. Analizar las ventajas y desventajas del uso de CCP para la representación de sistemas y procesos a nivel de la membrana celular.
- **Alcances**
  1. Una Interpretación generalizada de sistemas representados por membranas de acuerdo a los sistemas establecidos por Brane Calculi.
  2. Modelar en Utcc de cada una de las interacciones básicas entre membranas: fagocitosis, exocitosis y pinocitosis.

3. Identificar direcciones de cómo el modelo CCP puede contribuir en el estudio de los sistemas biológicos basados en interacción entre membranas.

1.3. Contribuciones

A continuación se presentan las principales contribuciones de este trabajo de grado:

1. Un estudio exploratorio de la aplicación del modelo CCP para representar sistemas biológicos que ocurren a nivel de la membrana celular, donde la inspiración biológica está dada por procesos de división celular, aceptación o expulsión de un agente biológico del interior de la célula a través de reacciones básicas como fagocitosis, exocitosis y pinocitosis.

2. Una interpretación en Utcc del cálculo Brane que permita representar procesos que se encuentran en Brane a procesos en Utcc. De esta manera logrando codificar un cálculo especializado en sistemas biológicos específicos, como es el caso del cálculo Brane y la interacción entre membranas , dejando camino abierto a que se explore y se avance más en el campo aplicativo para este tipo de sistemas ya que a pesar de que en Brane es posible verificar las propiedades de los sistemas modelados, aún no existe una herramienta que implemente las ideas formales del cálculo, lo que ha impedido que se avance en esta dirección ya que no es posible verificar la utilidad real del cálculo a nivel biológico.

3. Este trabajo de grado se constituye como un punto de partida para realizar las respectivas pruebas de verificación y corrección al modelo planteado. Adicionalmente se constituye en una base para futuras implementaciones.

1.4. Estructura del documento

La estructura de este trabajo se presenta a continuación:

- Capítulo 2: Este capítulo presenta un marco de referencia sobre los fundamentos biológicos y computacionales que inspiraran este trabajo. Inicialmente se introduce brevemente la programación concurrente por restricciones como un formalismo computacional para modelar sistemas concurrentes, presentando sus principales propiedades y características. Posteriormente se presentan algunos cálculos inspirados en este paradigma de programación (CCP), entre estos se introduce particularmente el cálculo Utcc (Universal Timed Concurrent Constraint) como una extensión de TCC, presentando una descripción de su sintaxis, semántica operacional y lógica asociada. Después se explican brevemente algunos conceptos de biología molecular, y se describen procesos biológicos que ocurren a nivel de la membrana celular detallando sus interacciones

- Capítulo 3: En este capítulo se propone el Modelo de Computación con membranas, describiendo cómo diferentes campos tanto científicos como computacionales han abordado el modelado de este tipo de sistemas, debido a la abundancia de interacciones de este tipo en diversos procesos celulares. Posteriormente se describen dos cálculos especializados en este tipo de procesos como lo son BioAmbient y Calculo Brane, para este último se realiza principalmente un enfoque exponiendo algunas propiedades básicas del cálculo, sus componentes fundamentales, sintaxis y reacciones básicas (e.g., fagocitosis, pinocitosis, exocitosis). Finalmente el capítulo concluye presentando reacciones más complejas derivables a partir de las reacciones básicas nombradas anteriormente.

- Capítulo 4: En este capítulo se propone una interpretación en Utcc de las membranas y sus interacciones definidas inicialmente en el Cálculo Brane. Estableciendo de esta manera un sistema de restricciones que definirá las relaciones entre los componentes reales del sistema y los elementos del cálculo. Logrando finalmente una codificación del cálculo describiendo sus principales componentes, así como las reacciones básicas para posteriormente

definir procesos más complejos. Finalmente el capítulo termina con una sección donde se discuten los aspectos más relevantes del *encoding* propuesto.

- Capítulo 5: Este capítulo está dedicado a presentar las conclusiones y puntos más relevantes relacionados con este trabajo de grado. El cápitulo inicialmente establece un conjunto de conclusiones que describen las principales ventajas y desventajas de usar Utcc para el modelado de procesos biológicos como interacciones entre membranas y también se proponen unas direcciones para trabajo futuro.

2 Fundamentos Biológicos y Computacionales

A continuación se introducen algunos fundamentos computacionales y biológicos que inspiran este trabajo. Estos conceptos representan un marco referencial para temas que abordaremos en los capítulos siguientes. Primero se introducirán los conceptos de cálculos de procesos, sistemas de restricciones, el modelo de Programación Concurrente por Restricciones y se presenta la sintaxis y semántica operacional del cálculo Utcc. Segundo se presentan algunos conceptos sobre sistemas biológicos, membrana celular, sus interacciones y el problema biológico a nivel celular.

2.1. Cálculos de Procesos

Los cálculos de procesos son una familia de formalismos propuestos para el modelado de sistemas concurrentes [EMR06]. Estos facilitan la abstracción de los diferentes agentes que componen un sistema y que interactúan entre sí para definir un comportamiento característico, por ejemplo, expresar la interacción entre dos agentes P y Q como la composición paralela de procesos ($P \parallel Q$). Una de las características de estos formalismos es que permiten la verificación de propiedades en los sistemas representados.

El fundamento matemático de los cálculos de procesos, hace de estos una herramienta importante en la descripción de los sistemas concurrentes [JGR05], ya que proporcionan reglas de inferencia y métodos de razonamiento que facilitan el análisis del comportamiento de los sistemas [AL06].

Actualmente existe una gran variedad de cálculos de procesos con distintas características como descripción de interacciones, comportamiento determinista y no determinista, comunicación (movilidad); algunos de ellos son extensiones de los ya existentes que permiten considerar algunos aspectos en el modelado de sistemas, como por ejemplo la noción de tiempo discreto [SJG94].

Existen diferentes ámbitos de estudio para los que algunos cálculos han sido propuestos, tomando como inspiración el tipo específico de sistemas en esos ámbitos. Se han utilizado los cálculos de procesos para representar modelos de sistemas biológicos[AG06], de improvisación musical [Tor09], de workflow, de protocolos de seguridad [ALRV05] y otros. Referente al campo de la biología han surgido cálculos inspirados en determinados tipos de sistemas biológicos, tales como BioAmbients [AREPWSS04] y el Cálculo Brane [Car98] los cuales se plantean como alternativa para el estudio de la Membrana Celular y otros procesos que ocurren a este nivel. Este tipo de cálculos inspirados en sistemas biológicos, presentan como característica propia la abstracción del tipo de comunicación entre procesos por analogía con el tipo de reacciones bioquímicas, en las que los cálculos mencionados se encuentran basados. Este tipo de sistemas son los que se consideran para el desarrollo de este trabajo, en el cual se pretende estudiar el uso de cálculos de procesos basados en restricciones y cómo este tipo de cálculos puede contribuir a la solución de problemas particulares en el campo biológico.

2.2. Programación Concurrente por Restricciones

La Programación Concurrente por Restricciones (CCP)[Sar93] es un formalismo computacional expresivo para la representación de sistemas concurrentes, cuyo concepto fundamental es el de «restricción», considerada como información parcial sobre un sistema. El modelo CCP combina el modelo declarativo basado en lógica, con la forma tradicional de los cálculos de procesos, lo cual permite tener en CCP gran variedad de mecanismos de razonamiento basados en lógica y cálculos de procesos [SJG94].

Figura 2.1: Almacén de Restricciones (Store)

En CCP los procesos o agentes que ocurren e interactan dentro de un sistema, se comunican mediante la adición y consulta de información en un almacén denominado store (ver Figura 2.1). La información almacenada en el store corresponde a información parcial (restricciones. e.g., $x + y < 50$) que determinan el comportamiento del sistema. Dicha información permite la comunicación y sincronización entre los procesos, los cuales pueden consultarla y agregar nueva información. En el modelo CCP se cuenta con operaciones para consulta (*ask c*), adición de información (*tell c*)[Sar93], además se puede ocultar información correspondiente a un proceso (*local x in P*). Los cálculos de procesos basados en CCP utilizan la adición de nuevos operadores para definir comportamientos no determinísticos (+), movilidad (abs x;c), comportamiento probabilístico, entre otros.

2.2.1. Sistema de Restricciones

Los cálculos de procesos basados en el modelo CCP se encuentran parametrizados por algún sistema de restricciones, el cual especifica la forma de las restricciones que pueden ser adicionadas o consultadas del almacén de información. Es decir determinan la forma de la información que se requiere para modelar un sistema.

Una restricción representa información parcial con la cual pueden interactuar los procesos dentro de un sistema. Luego un sistema de restricciones está conformado por una pareja ($\sum$,Δ), donde $\sum$ representa el conjunto de constantes, funciones y predicados, y Δ es una teoría consistente de primer orden sobre $\sum$. De esta forma, las restricciones se pueden considerar como fórmulas de primer orden sobre elementos de $\sum$[Sar93].

Adicionalmente, los sistemas de restricciones proveen la relación de deducción $\models$ sobre Δ, la cual especifica interdependencia entre las restricciones. Supongamos

que se tiene $c \models d$, lo que indica que la restricción d puede ser deducida de la información que representa la restricción c (e.g., $X > 20 \models X > 12$). Lo anterior llevado al modelo CCP podrá ser pensado como una restricción d que puede deducirse del Store representado por c. La relación $\models_\Delta$ se define como: $c \models_\Delta d$ si la relación de implicación $c \rightarrow d$ es válida en Δ.

2.3. Ntcc

Ntcc [MNV02b] es un cálculo basado en el modelo CCP y a su vez representa una extensión del mismo. Ntcc permite considerar las nociones de tiempo discreto, no determinismo y comportamiento asíncrono. Este cálculo permite controlar la ejecución de los procesos en el transcurso de las unidades de tiempo, debido a que considera la noción de tiempo entre sus características. Es decir, en Ntcc se puede indicar que un proceso se ejecuta en la unidad de tiempo actual, en la siguiente unidad de tiempo, en todas las siguientes unidades de tiempo o que se ejecuta en alguna unidad de tiempo futura. Lo anterior señala que el cálculo Ntcc permite modelar sincronización entre procesos que se ejecutan concurrentemente, también modelar comportamientos asíncronos por medio de operadores que retrasan la ejecución de los procesos y comportamiento infinito para ejecución de procesos que persisten a lo largo del tiempo.

Este cálculo se encuentra fundamentado en la lógica temporal lineal [MNV02b], lo cual en conjunto con su forma declarativa permite la verificación de propiedades representadas como fórmulas en la lógica temporal.

2.3.1. Sintaxis de Ntcc

A continuación se presenta la sintaxis de Ntcc (Cuadro 2.1):

P,Q. ... ::=	tell(c) | $\sum_{i \in I}$when c_i do P_i | P‖Q | local x in P
	| next P | unless c next P | !P | $\star$P

Cuadro 2.1: Sintaxis de Ntcc

El proceso *tell (c)* permite añadir la restricción c como información al Store. El proceso $\sum_{i \in I} when\ c_i\ do\ P_i$ representa una escogencia no determinista para un proceso P_i para el cual se debe deducir la guarda c_i del Store. Esta escogencia es de tipo excluyente. Para el caso de escogencia binarias se hace uso del operador + (e.g. P + Q). El operador ‖ denota la ejecución paralela entre procesos. $P \parallel Q$ indica que los proceso P y Q se ejecutan concurrentemente. El proceso *local x in* P establece la variable local x al proceso P. En este caso la información de x sólo puede ser accedida por P. El proceso *next* P, ejecuta P en la siguiente unidad de tiempo. El proceso *unless c next* P ejecuta P en la siguiente unidad de tiempo, si no se puede deducir del store la guarda c. El operador ! permite modelar el comportamiento persistente de procesos. Es decir, !P permite que P se ejecute en cada unidad de tiempo. El operador $\star$ modela el comportamiento asíncrono. El proceso $\star$P representa un retraso arbitrario y finito de la activación del proceso P. Es decir, el proceso P se ejecuta en una unidad de tiempo futura.

2.3.2. Semántica Operacional

A continuación se presenta la semántica operacional de Ntcc (Figura 2.2), mediante la cual se formaliza el comportamiento de los procesos, representando el estado inicial del Store y la ejecución de un proceso por medio de parejas $< P, c >$, donde P representa un proceso en Ntcc y c representa el Store.

La relación $\rightarrow$ denota las transiciones al momento de evolucionar un proceso, lo cual puede representar modificación en la información del Store. Es decir, $<$

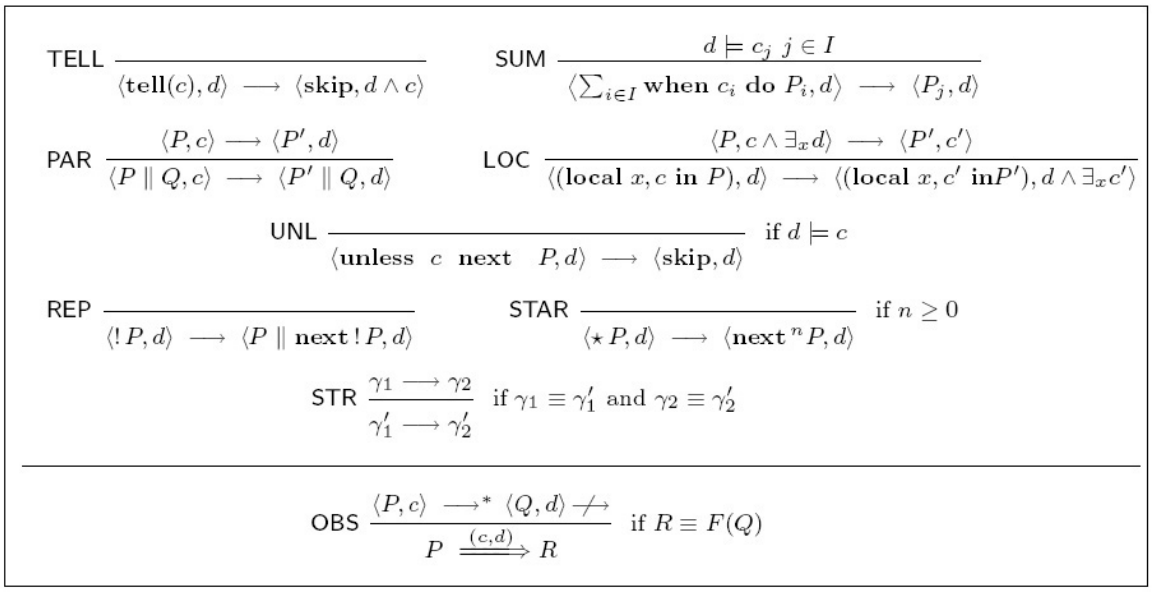

Figura 2.2: Semántica Operacional de Ntcc

$P, d >\!\!\rightarrow\!< P', d' >$ representa las transiciones no observables, y el cambio del Store que inicialmente es d y que se modifica durante la ejecución de P quedando en un paso interno como d' y con la transformación de P a P'. También se presenta la relación de transición observable $P \stackrel{(c,d)}{\Longrightarrow} R$, que se entiende como: un proceso P con estímulo c se reduce en una unidad de tiempo a R con salida d al ambiente.

2.4. Movilidad en el Cálculo π

El cálculo π se enfoca en la descripción de sistemas móviles, en los cuales la configuración puede cambiar durante la computación [Mil99]. Este cálculo representa una extensión de CCS [Mil89], caracterizado por ser uno de los cálculos más representativos utilizados para el modelado de sistemas en distintos ámbitos. La movilidad en el cálculo π se puede entender como la generación de nombres o links que pueden ser comunicados entre los procesos, cambiando la estructura de comunicación de un sistema.

2.4.1. Sintaxis del Cálculo π

El cálculo π representa la comunicación por medio de canales y variables. Utiliza los nombres como sus entidades primitivas. A continuación (Cuadro 2.2) se presenta la sintaxis del cálculo π. Se considera un conjunto finito de nombres (puertos, links o canales), los cuales se representan por $x, y, z \ldots$ y para cada nombre se asume la existencia de su correspondiente nombre complementario (co-name) denotado como $\overline{x}$.

P,Q ...	::=	$\sum_{i \in I}(\pi_i.P_i) \mid (vx)P \mid P \| Q$
α	::=	$\overline{(x)}y \mid x(y)$

Cuadro 2.2: Sintaxis cálculo π

P y Q representan procesos, x es nombre y π un prefijo. El prefijo denota acción atómica previa a un proceso. Esta acción representa la comunicación entre procesos señalando una acción de envío o recepción de un dato por algún canal (e.g. el envío de un dato z por el canal x se denota por $\overline{x} < z >$, y para representar la recepción que se tiene de algún dato a por un canal c se tiene $c(a)$). El proceso $P \parallel Q$ denota la ejecución paralela de los procesos Q y P. Al igual que en Ntcc el proceso !P representa la replicación del proceso P o ejecución infinita de P. El proceso $(vx)P$, liga el nombre x al proceso P. Finalmente, el cálculo cuenta con escogencias no deterministas entre una serie de procesos P_i mediante el proceso $\sum_{i \in I} \pi.P_i$, dicha escogencia depende de la comunicación entre dichos procesos.

2.5. Utcc

El cálculo Utcc (*Universal Timed Concurrent Constraint*)[Ola09] es un lenguaje derivado de tcc (extensión de CCP), el cual cuenta con un nuevo constructor para la abstracción de procesos. En Utcc al igual que en tcc, la noción de tiempo es la misma. El tiempo se encuentra conceptualmente dividido en intervalos de tiempo o unidades de tiempo. También se considera el concepto de información parcial (restricción) y un almacén en que reposa la información, por medio del cual los

procesos se sincronizan e interactúan. Este cálculo incrementa la expresividad del lenguaje tcc, permitiendo modelar comportamiento infinito y movilidad [COV07]. Utcc implica una nueva teoría para el modelo CCP, donde los procesos pueden cambiar sus mecanismos de interacción por medio de la comunicación de sus nombres locales, del mismo modo en que se utiliza la movilidad en el cálculo π. Para tener mejor comprensión de la extensión de tcc, consideremos el siguiente ejemplo,[Ola09] donde se tiene un predicado (restricción) de la forma $out(.)$ y un proceso $P = when\ out(x)$ do R. Suponga que se tiene la entrada $out(42)$, en P no ejecuta R ya que $out(42)$ no implica $out(x)$ (i.e., $out(42) \models out(x)$).

Para modelar movilidad, Utcc reemplaza el operador ask de tcc por una construcción más generalizada y parametrizada denominada $(abs\ x;c)P$. Este proceso puede ser visto como una lambda-abstracción del proceso P sobre la variable x bajo la restricción c. El proceso $Q = (abs\ x;c)P$ ejecuta $P[t/x]$ (i.e. P con las ocurrencias libres de x substituidas con t) en el intervalo de tiempo actual. Por ejemplo, $P = (abs\ x\ ;\ out(x))R$ con entrada $out(42)$ ejecuta $R[42/x]$.

Desde una perspectiva lógica, se puede hacer una analogía entre el operador de localidad y el nuevo operador introducido por Utcc. El proceso $(localx; c)P$ y $(abs\ x;c)P$ corresponden respectivamente a los cuantificadores existencial y universal de la lógica ($\exists x(c \wedge Fp)$ y $\forall x(c \rightarrow Fp)$) donde Fp corresponde al proceso P.

2.5.1. Sintaxis Utcc

La sintaxis de Utcc se presenta a continuación (Cuadro 2.3); recordemos que éste extiende a tcc permitiendo modelar movilidad.

P,Q ... ::=	$skip \mid tell(c) \mid P \| Q \mid !P\ (local\ \overline{x};c)P$
	$\mid (abs\ x; c)P \mid next\ P \mid unless\ c\ next\ P$

Cuadro 2.3: Sintaxis de Utcc

El proceso $skip$ no representa la ejecución de algún proceso, representa inacción.

Comunicación El proceso $tell\ (c)$ adiciona c al Store en la unidad de tiempo

actual. El proceso $(abs\ x; c)P$ que representa la operación ask, mediante el cual se realiza deducción del store para activar el proceso P. (el proceso $(abs\ x; c)P$ es utilizado para representar comunicación entre sistemas (movilidad)).

Concurrencia El proceso $P \parallel Q$ representa la ejecución concurrente en la unidad de tiempo actual de los procesos P y Q.

Información Local El proceso $local\ x\ in\ P$ establece la variable x como variable local de P, es decir que toda información de x producida por P sólo puede ser vista por P , y la información de x producida por otros procesos diferentes a P, no puede ser vista por P.

Constructores Temporales El proceso $next\ P$ puede ser visto como un retraso en la ejecución de P, debido a que ejecuta P en el siguiente intervalo de tiempo. La consulta negativa $unless\ c\ nextP$ representa también un retraso condicionado en la ejecución de P, debido a que P se ejecuta en la siguiente unidad de tiempo únicamente si la guarda c puede ser deducida del Store.

Comportamiento infinito El operador de replicación ! permite establecer una copia de un proceso en cada unidad de tiempo futura. Es decir, el proceso !P establece que P se ejecuta en la unidad de tiempo actual y repite su ejecución en las siguientes unidades de tiempo, puede verse como $P \parallel next\ P \parallel next^2\ P \ldots$, pero sólo puede determinar una ocurrencia de P por cada unidad de tiempo. En Utcc se puede modelar comportamiento infinito en la misma unidad de tiempo, a diferencia de Tcc y Ntcc (el proceso $(abs\ x; c)P$ se ejecuta durante todo el intervalo de tiempo en el que ocurre, logrando abstraccin sobre x de la guarda c).

2.5.2. Movilidad en Utcc

En Utcc la movilidad se obtiene mediante la interacción entre procesos que realizan abstracciones y procesos local [COV07]. La movilidad en Utcc es comprendida en el mismo sentido del cálculo π, como la comunicación de variables privadas entre procesos.

Ejemplo. Se tienen los procesos:

$P = (abs\ y; out_1(y))tell(out_2(y))$
$Q = (local\ z)(tell(out_1(z)) \parallel when\ out_2(z)\ do\ next\ tell(out_2(0)))$

Según lo anterior un link y es enviado por el canal out_1, P lo reenvia por out_2. Luego, Q envía un link privado z por out_1 y en caso de ser devuelto por out_2 entonces envía 0 por out_2. Si ahora se ejecuta $P \| Q$ se tiene $\gamma = < P \| Q, true >$

$$
\begin{aligned}
\gamma \quad & \rightarrow * < (local\ z)(tell(out_1(z)) \| when\ out_2(z)\ do\ next\ tell(out_2(0)) \| P), \\
& \qquad true > \\
& \rightarrow * < (local\ z; out_1(z))(skip \| when\ out_2(z)\ do\ next\ tell(out_2(0)) \| P), \\
& \qquad \exists_z(out_1(z)) > \\
& \rightarrow * < (local\ z; out_1(z))(when\ out_2(z)\ do\ next\ tell(out_2(0)) \| P'), \\
& \qquad \exists_z(out_1(z)) > \\
& \rightarrow * < (local\ z; out_1(z) \wedge out_2(z))(next\ tell(out_2(0)) \| P'), \\
& \qquad \exists_z(out_1(z) \wedge out_2(z)) >
\end{aligned}
$$

Donde $P' = (abs\ y; out_1(y) \wedge y \neq z)(tell(out_2(y)))$. Luego $d = \exists_z(out_1(z) \wedge out_2(z))$. Se concluye $P \| Q \overset{(true,d)}{\Longrightarrow} tell(out_2(0))$.

2.5.3. Semántica operacional de Utcc

La semántica operacional de Utcc considera las transiciones entre el *Store* y los procesos mediante la configuración $< P, c >$ donde P representa el proceso en ejecución y el *Store* es representado por la restricción c.

Transiciones Internas (no observables) y Transiciones Observables.

En la tabla de semántica operacional el proceso, las relaciones $\rightarrow$ y $\Longrightarrow$ indican transiciones internas y observables respectivamente. La transición interna $< P, d > \rightarrow < P', d' >$ se debe leer como "P con *Store* d reduce, en un paso interno no observable, a P'con Store d' ". Mientras que las transiciones observables $P \overset{(c,d)}{\Longrightarrow} R$ se interpretan como "P con entrada c, reduce en una unidad de

tiempo a R con salida al ambiente d". Las transiciones observables se obtienen de secuencias finitas de transiciones internas.

$$\mathrm{R_{TELL}}\ \frac{}{\langle \mathbf{tell}(c), d\rangle \longrightarrow \langle \mathbf{skip}, d \wedge c\rangle}$$

$$\mathrm{R_{PAR}}\ \frac{\langle P, c\rangle \longrightarrow \langle P', d\rangle}{\langle P \parallel Q, c\rangle \longrightarrow \langle P' \parallel Q, d\rangle}$$

$$\mathrm{R_{LOC}}\ \frac{\langle P, c \wedge (\exists \vec{x} d)\rangle \longrightarrow \langle P', c' \wedge (\exists \vec{x} d)\rangle}{\langle (\mathbf{local}\, \vec{x}; c)\, P, d\rangle \longrightarrow \langle (\mathbf{local}\, \vec{x}; c')\, P', d \wedge \exists \vec{x} c'\rangle}$$

$$\mathrm{R_{UNL}}\ \frac{d \models_\Delta c}{\langle \mathbf{unless}\ c\ \mathbf{next}\ P, d\rangle \longrightarrow \langle \mathbf{skip}, d\rangle}$$

$$\mathrm{R_{REP}}\ \frac{}{\langle !\, P, d\rangle \longrightarrow \langle P \parallel \mathbf{next}\, !\, P, d\rangle}$$

$$\mathrm{R_{ABS}}\ \frac{d \models_\Delta c[\vec{t}/\vec{x}] \qquad [\vec{t}/\vec{x}] \text{ is admissible.}}{\langle (\mathbf{abs}\ \vec{x}; c)\, P, d\rangle \longrightarrow \langle P[\vec{t}/\vec{x}] \parallel (\mathbf{abs}\ \vec{x}; c \wedge \vec{x} \neq \vec{t}\ )\, P, d\rangle}$$

$$\mathrm{R_{STR}}\ \frac{\gamma_1 \longrightarrow \gamma_2}{\gamma_1' \longrightarrow \gamma_2'}\ \text{if } \gamma_1 \equiv \gamma_1' \text{ and } \gamma_2 \equiv \gamma_2'$$

$$\mathrm{R_{OBS}}\ \frac{\langle P, c\rangle \longrightarrow^* \langle Q, d\rangle \not\longrightarrow}{P \overset{(c,d)}{\Longrightarrow} F(Q)}$$

Figura 2.3: Semántica Operacional de Utcc

A continuación se tiene como ejemplo la interpretación de algunas de las reglas establecidas en la semántica operacional:

- La regla R_{TELL} indica que el proces $tell(c)$ adiciona c al Store actual d, y que luego evoluciona al proceso *skip*.
- El proceso $P = unless\ c\ next\ Q$ ejecuta Q en la siguiente unidad de tiempo solamente si del store resultante (Store final) en la unidad de tiempo actual

no se puede deducir la guarda c. En la regla R_{UNL} el proceso P evoluciona en *skip* si es posible deducir c del store d.

- La regla R_{ABS} describe el comportamiento de $P = (abs\ x; c)Q$. Si el Store actual permite deducir $c[t/x]$, entonces el proceso $P[t/x]$ se ejecuta. Adicionalmente, la abstracción persiste en el intervalo de tiempo actual, lo que permite activar otras ejecuciones de P con la substitución de x. Se puede notar que el Store es incrementado con $x \neq t$ para evitar ejecuciones repetidas de $P[t/x]$. Cuando se menciona "$[t/x]$ es admisible"se quiere decir que ninguna de las variables en x debe ser sintácticamente igual a un elemento en t.

2.6. Fundamentación Biológica

La membrana celular es la parte externa de la célula que envuelve el citoplasma, permite el intercambio de elementos (agua, gases y nutrientes) entre la célula y el medio que la rodea, también se encarga de la expulsión de desechos hacia el exterior.

La membrana plasmática se encuentra constituida por dos capas lípidas, estando ubicadas en el centro otras dos capas fosfolípidas, con un espesor de tan solo una molécula. Los extremos de estas capas repelen el agua. (ver Figura 2.4).

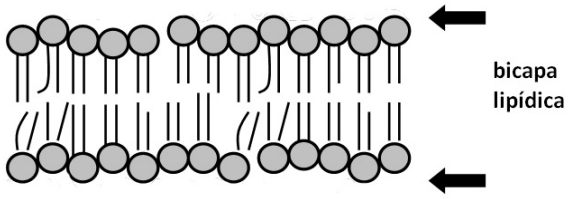

Figura 2.4: Estructura de la membrana celular.

2.6.1. La Membrana como Estructura Dinámica

En los procesos mediante los cuales se ingresan o expulsan agentes al interior y exterior de la membrana, algunas regiones de ésta se mueven dinámicamente a

través de la célula. Es decir, estos procesos permiten que se formen arreglos en la membrana para fluir de un compartimiento a otro. De acuerdo a lo anterior, las membranas se deben considerar como algo dinámico y no como una estructura estática. La membrana forma vesículas por brotes de ella misma. Ya formadas, estas vesículas viajan a través de la célula y posteriormente se fusionan con otros sistemas de membrana. De esta manera los materiales que se incluyen en las vesículas son transportados a diferentes lugares o expulsados del interior de la célula.

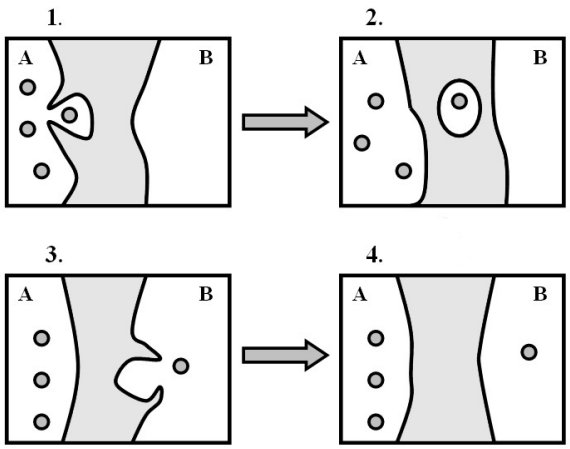

Figura 2.5: Flujo dinámico de la membrana a través de la membrana

En la Figura 2.5 se ve el flujo de una particula desde un compartimento A hasta otro compartimento B. Primero la membrana se invagina (1) rodeando la partícula, luego se forma la vesícula con la partícula (2), luego en (3) la vesícula se habre liberando la partícula en el compartimento B en (4).

Los procesos mencionados anteriormente, en los cuales los elementos son transportados al interior o exterior de la célula, se clasifican en dos tipos (Endocitosis y Exocitosis) y se explican en la siguiente sección.

2.7. Procesos a Nivel de Membrana Celular: Endocitosis y Exocitosis

Todas las funciones de transporte de elementos por medio de vesículas se dan gracias a las reacciones básicas que se denominan como endocitosis y exocitosis [Car08]. La primera reacción se refiere al proceso por el cual la célula introduce en su interior moléculas grandes o pequeñas partículas, también introduce fluidos. La segunda hace referencia al proceso celular por el cual se liberan elementos contenidos en el interior.

2.7.1. Endocitosis

La endocitosis es el movimiento de material del exterior hacia el interior de la célula, lo cual se hace por medio de vesículas que se forman en la membrana. Las membranas se invaginan para formar las vesículas, encerrando material que se encuentra en el exterior y transportándolo al interior. Pero este proceso se debe considerar en dos formas: cuando el material ingresado es algún agente no líquido (fagocitosis) y cuando se ingresa fluido (pinocitosis).

En el proceso de **fagocitosis** (Figura 2.6), la célula engulle desechos, bacterias u otros elementos grandes. Este tipo de proceso es realizado por células especializadas, las cuales se denominan fagocitos, en los que se incluyen los macrófagos, neutrófilos y otros glóbulos blancos de la sangre [EP03]. En este proceso, se le conoce a la vesícula producida por la invaginación como fagosoma, que por lo general se terminan fusionando con los lisosomas.

Figura 2.6: Fagocitosis.

En la **pinocitosis**, la célula engulle fluido extracelular, en el cual se incluyen moléculas como azúcar y proteínas. Dichos materiales son llevados al interior por

medio de una vesícula. Cabe aclarar que este fluido extracelular no se mezcla con el citoplasma a pesar de haber ingresado. Un ejemplo de células que realizan este proceso son las células epiteliales en los capilares, las cuales toman porción líquida de la sangre en la superficie capilar.

2.7.2. Exocitosis

La exocitosis es el proceso mediante el cual se transporta material que se encuentra en el interior de la célula hacia el exterior de la misma (ver Figura 2.7). Este proceso también se basa en la formación de vesículas membranosas para llevar a cabo la expulsión de los elementos. Este proceso se observa en diversas células secretora, tanto en la función de excreción como en la función endocrina.

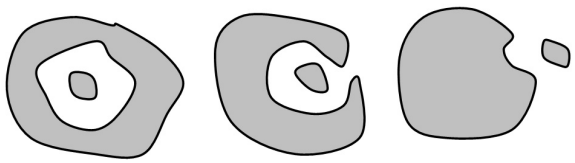

Figura 2.7: Exocitosis.

2.8. Biología y Computación

En la naturaleza existen muchos sistemas o fenómenos; de éstos, algunos que no se comprenden aún y otros que son entendidos de manera parcial debido a que se encuentran sujetos a suposiciones de acuerdo al comportamiento observado. Este tipo de problemas ha sido abordado por modelos computacionales propuestos para estudiar la interacción entre los componentes de un sistema. Algunos de estos modelos han permitido tener mejor comprensión de este tipo de sistemas por parte de la comunidad científica. Muchos de los formalismos computacionales basados en diferentes paradigmas, son considerados en la actualidad una herramienta de gran importancia para el estudio de sistemas biológicos.

Uno de los propósitos de la computación es la búsqueda de modelos que formalicen procesos que ocurren en la naturaleza. Estos procesos presentan cambios o evolu-

ción en el tiempo, lo cual sugiere que se establezca sobre cuáles elementos se debe razonar y qué operaciones actúan sobre dichos elementos. Es a partir de aquí que surge el modelo de la Computación Natural (natural computing), [Ghe04] una disciplina que trata de modelizar mecanismos computacionales inspirándose en la forma en que ocurren los procesos en la naturaleza.

En la Computación Natural se encuentran diversas áreas, entre ellas la Computación Molecular y la Computación Celular con Membranas [dJPJC06]. Por un lado, la computación molecular tiene como objetivo el estudio y uso de moléculas orgánicas como elemento base de cualquier sistema. Especialmente se centra en tratar las moléculas como chips electrónicos, los cuales pueden presentar ciertas ventajas como mayor rapidez y menor tamaño. Este campo intenta aprender de la biología molecular sobre los mecanismos y procesos que condicionan el comportamiento celular, que se encuentra representado en el ADN. Por otro lado, la Computación Celular con Membranas estudia el concepto de localidad en los sistemas. Es decir, considera el comportamiento de la célula como una máquina que realiza ciertos procesos de cálculo, desde el punto de vista biológico, [Ghe04]. Dichos procesos ocurren en una distribución jerárquica de membranas interiores, en donde se produce el flujo y modificación de los componentes químicos que la misma célula procesa.

La célula puede ser vista como la conformación de distintos sistemas biológicos, los cuales se encuentran delimitados por distintas clases de membranas. La membrana celular permite aislar el interior de la célula del medio exterior. Incluso en el interior existen membranas que delimitan el interior de las vesículas que se encuentran en la célula. El modelo de Computación Celular con Membranas fue introducido por Gh. Paun en 1998 [Ghe00], como un modelo de tipo distribuido y paralelo, inspirado en el funcionamiento de la célula como organismo vivo capaz de generar y procesar información.

Lo anterior motiva al desarrollo de modelos y aplicación de formalismos para describir el funcionamiento de los sistemas biológicos, de manera que se puedan representar con base en la información existente y poder estudiar su comportamiento. Para efectos de este trabajo de grado, nos vamos a enfocar en el fundamento biológico de las membranas celulares y las interacciones que ocurren en las mismas.

Es importante resaltar algunos de los formalismos existentes diseñados especialmente para el estudio las interacciones entre membranas, los cuales constan de una sintaxis que facilita el modelado de este tipo de sistemas. Entre estos formalismos tenemos el cálculo Brane de Luca Cardelli [Car98], del cual se toma como referencia el modelo de las interacciones entre membranas para aplicar formalismos basados en el modelo CCP en la interpretación de dichas interacciones, los cuales se explican en el siguiente capítulo.

2.9. Resumen

En este capítulo se presentó un marco de referencia, para estudiar los principales conceptos que trata este trabajo de grado. Inicialmente se introdujeron los Cálculos de Procesos como formalismos de gran importancia en el modelado y estudio de diversos sistemas. Seguidamente se presentó el modelo de la Programación Concurrente por Restricciones (CCP) como un paradigma para el estudio de sistemas concurrentes basados en información parcial, también se explicó el concepto e importancia de los sistemas de restricciones.

Uno de los formalismos basados en CCP y que se explica en éste capítulo, es el cálculo Ntcc. En Ntcc se introducen operadores para modelar comportamiento asíncrono y no determinista, considerando la noción de tiempo. Se presentó la sintaxis de dicho cálculo, al igual que su semántica operacional, explicando la noción de transiciones observables y no observables.

Uno de de los conceptos de gran importancia en el desarrollo de esta tesis, es la movilidad como característica de los cálculos para el modelado de sistemas basados en comunicación, como los sistemas biológicos. Para nuestro caso la representación de membranas como nombres locales a procesos, las cuales son comunicadas por medio del store para coordinar interacciones. Para explicar el concepto de movilidad, se hizo una breve introducción del cálculo Pi y su sintaxis, el cual cuenta con dicha característica. Seguidamente se explica el cálculo Utcc como una extensión de tcc que permite modelar comportamiento infinito y movilidad. La introducción al cálculo Utcc es de gran importancia en este trabajo de grado, ya que es el formalismo que se utilizará para lograr una interpretación de los modelos basados en

membranas propuestos por el cálculo Brane. En este capítulo se explica la sintaxis de Utcc en conjunto con su semántica operacional. También se hizo énfasis en la movilidad que proporciona este formalismo, la cual es posible por el operador de abstracción que introduce este cálculo.

Luego de presentar los fundamentos computacionales, se expuso el concepto de membrana celular y su estructura, con el fin de abordar los conceptos biológicos de importancia que son tratados en ésta tesis. Se expuso la membrana celular como un sistema dinámico y no estático. También se explicó los procesos que desarrolla la membrana celular denominados endocitosis y exocitosis (el primero de ellos comprendido en otros dos procesos: fagocitosis y pinocitosis), ya que son éstos el objeto de estudio de este trabajo de grado. Finalmente se expuso un marco referencial a la relación entre la biología y la computación (natural computing), en la que se explica brevemente cómo los procesos que ocurren en la naturaleza han servido de inspiración para el desarrollo de modelos que permitan estudiar su comportamiento.

3 Cálculos Biológicos Inspirados en Interacciones entre Membranas

Este capítulo es un marco de referencia que tiene como objetivo presentar de manera más detallada uno de los modelos de computación Bio-inspirados, en donde procesos que ocurren de forma natural inspiran nuevos paradigmas de computación. En este capítulo nos centraremos específicamente en el *Modelo de Computación con membranas* [Ghe04] y adicionalmente se presentarán algunos cálculos que surgen como herramientas alternativas de estudio, permitiendo modelar de manera más sencilla dichos procesos biológicos. Por último y por razones a fines a este trabajo de grado, nos centraremos en el Cálculo Brane [Car98], detallando su sintaxis, definición de interacciones básicas y derivadas para posteriormente analizar sistemas biológicos más complejos.

3.1. Modelo de computación con membranas

Actualmente uno de los principales problemas en Bio-informática es el estudio de los diferentes mecanismos a través de los cuales la membrana se comunica con el exterior. Este tipo de modelos siempre han generado gran interés por parte de científicos tanto en el área de la biología como en la computación. La primera aproximación a este tipo de Modelos ocurrió en Octubre de 1998 cuando Gh. Paun introduce un nuevo modelo de computación de tipo paralelo y distribuido conocido como P sistemas de transición [Ghe00] dando a conocer el primer modelo de un nuevo paradigma que con el tiempo se daría a conocer como Computación

Celular con membranas.

A diferencia de otros modelos de computación bioinspirados, cabe destacar que en este modelo no se proporciona una serie de operaciones básicas que actúan sobre un dato de entrada para generar un resultado final, sino que se crean diversos mecanismos que al ser ejecutados modifican el contenido de los diversos elementos que lo componen, logrando así una evolución del sistema a un próximo estado con diferentes características. Por ejemplo, la célula a través de procesos de asociación o división celular evoluciona a otro estado con diferentes propiedades.

De acuerdo a esto podemos decir que el Modelo de Computación con membranas está inspirado en un nivel biológico superior, donde se analiza el funcionamiento de la célula como organismo vivo capaz de procesar y comunicar información al exterior e interior a través de la membrana celular. Dichos procesos involucran el traspaso de elementos (e.g., oxígeno, aminoácidos, agua) necesarios para el correcto desarrollo de los procesos celulares. Por lo tanto, alteraciones en este tipo de sistemas pueden descontrolar o afectar notablemente el funcionamiento de la célula.

Son varios los estudios que computacionalmente se han realizado enfocados a este tipo de modelos, dando como resultados diversos Cálculos de Procesos y técnicas basadas en restricciones para el modelado de sistemas biológicos. Algunos inspirados en los cálculos tradicionales como el Cálculo Brane, otros planteando una sintaxis afín al sistema a modelar, o utilizando técnicas como lógica espacial que permitan modelar la noción de ambientes que ocurre a nivel celular. A continuación presentaremos dos de estos cálculos, enfocados a este tipo de interacción biológica que ocurre entre membranas: BioAmbients y Cálculo Brane para finalmente centrarnos en este último.

3.2. BioAmbients

BioAmbients [AREPWSS04] hace parte de los cálculos centrados en el modelo de interacciones a través de membranas. BioAmbients proviene del Cálculo Ambient y por esta razón conserva sus principales características, entre ellas la creación de ambientes independientes y móviles en los que se ejecutan procesos del cálculo pi

[Mil99].

Un Ambiente se define como un lugar delimitado donde ocurren procesos [Car98]. Es decir, existe una frontera que determina lo que está por dentro y lo que está por fuera. Cada ambiente alberga una serie de procesos, los cuales residen y se ejecutan en su interior. Cabe resaltar que es posible la anidación entre ambientes, cada ambiente con subambientes y su respectivo contenido.

Un ejemplo de ambiente puede ser n(P), donde n es el nombre que recibe el ambiente (es opcional, se utiliza para facilitar la lectura pero no tiene ninguna funcionalidad) y P es el proceso que está ocurriendo dentro de n. La Figura 3.1 representa un ejemplo de anidación de ambientes, en este caso el ambiente n contiene los procesos P y Q en ejecución paralela con el subambiente m que a su vez contiene el proceso R.

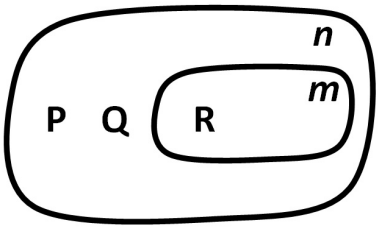

Figura 3.1: Ambientes en sintaxis BioAmbient

Uno de los principales objetivos de este cálculo es estudiar movilidad, donde los procesos pueden moverse a través de la jerarquía de ambientes y modificarlos, es decir que la movilidad es vista como un cambio en la configuración espacial a través del tiempo. A continuación se presentan las principales ventajas y desventaja del cálculo.

Ventajas:

- Permite ver el sistema como un conjunto de elementos que interactúan entre sí y estos a su vez tienen características particulares que lo definen y se encuentran en un determinado ambiente.
- La facilidad con la que se puede incluir un modelo dentro de otro, simplemente indicándole que se encuentra dentro de un ambiente y definiendo una interfaz que le permita interactuar con el medio exterior

Desventajas:

- Existe la necesidad de complementariedad entre los componentes biológicos para su interacción, algo que realmente no es siempre necesario a menos que se trate de una interacción a través de motivos moleculares [JGR05]. Esto debido que BioAmbients es un cálculo especializado en el modelado de ambientes y jerarquización entre procesos, razón por la cual para cualquier otro tipo de sistema es necesario definir la complementariedad entre los componentes biológicos.

BioAmbients define tres tipos de comunicación y seis primitivas para la manipulación de los ambientes. Básicamente los tipos de comunicación son: uno local entre procesos de un mismo ambiente (e.g., dos procesos en paralelo); uno entre procesos localizados en ambientes que se encuentren a un mismo nivel jerárquico (e.g., dos ambiente en paralelo) y uno entre procesos localizados en distintos ambientes en los que uno de ellos tiene un nivel jerárquico mayor que el otro (e.g., un ambiente se encuentra dentro de otro). Por otro lado, se tienen primitivas para entrar, aceptar, expulsar, salir y mezclar ambientes.

3.3. Cálculo Brane

Cálculo Brane es otro de los cálculos enfocados en el modelado de interacciones a través de membranas biológicas [Car98]. Enfocándose de esta manera en los mecanismos que permiten a la célula comunicarse con el exterior a través de la membrana. En Brane este tipo de comunicación no se limita únicamente a la realizada a través de la membrana citoplasmática sino que gracias a las operaciones definidas en el cálculo también es posible modelar interacciones a través de la membrana nuclear o de las membranas de los organelos.

Analizando de una manera más detallada la Membrana Biológica la podemos definir como una superficie cerrada que cumple con una propiedad de orientación y puede realizar diferentes funciones moleculares. De esta forma podemos ver la membrana no únicamente como contenedores sino también como coordinadores y sitios activos de mayor actividad, es decir que la inspiración fundamental del

cálculo se centra en el hecho de entidad activa, la cual reside sobre la membrana y no dentro de ella. A continuación hablaremos de las proteínas como moléculas funcionales que se encuentran inmersas en las membranas con una orientación consistente y pueden actuar en ambos lados de la membrana simultáneamente. La orientación consistente de dichas proteínas induce en ultimas una orientación en las membranas.

Una de las propiedades más notable de las membranas biológicas es que forman un fluido bidimensional (una bicapa lípida) inmerso en un líquido tri-dimensional (agua). De esta manera, tanto los componentes estructurales como la proteína inmersa pueden difundirse libremente en el plano bidimensional de la membrana. Adicionalmente las membranas pueden flotar en agua, la cual puede contener otras moléculas que se difundan libremente en ese liquido tridimensional [Car98]. Muchas membranas son altamente dinámicas y constantemente están cambiando: se mezclan o se separan, pero dichos cambios se ven limitados, parcialmente porque la orientación de las membranas debe ser conservada durante toda la reacción y las transformaciones deben ser justamente continuas.

Estas propiedades sirven como referencia para hablar de la estructura Fluido dentro de Fluido la cual inspira la organización básica del Calculo Brane y plantea ciertas restricciones que delimitarán las reacciones básicas del Cálculo. Una de estas restricciones es preservación de la orientación (e.g., membranas de diferente orientación no deberían mezclarse). Otra restricción que debe establecerse es *bitionality* la cual establece que las membranas anidadas deben tener orientaciones opuestas, siendo representadas por un sistema de colores en dos tonalidades

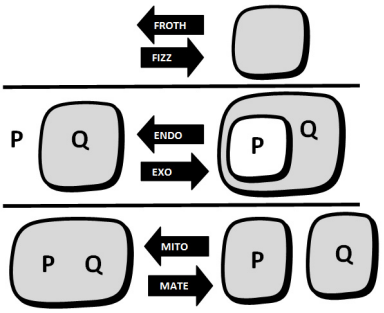

Figura 3.2: Interacciones en las Membranas

Como vemos en la Figura 3.2 , sean P y Q dos sistemas arbitrarios; la preservación de la orientación consiste en que las reacciones deben conservar la orientación con las cuales se encuentran anidadas dentro de la membrana. Nótese que P y Q finalizan el proceso con el mismo color inicial en cada reacción. Ahora podemos concluir que en una secuencia de reacciones bitonales nunca existirá una mezcla de los fluidos del exterior al interior, aunque cabe notar que fluidos externos pueden ingresar al interior de la membrana siempre y cuando estén encerrados en otra membrana.

3.3.1. Sintaxis

La estructura básica del Cálculo Brane está compuesta fundamentalmente por dos elementos: Sistemas y membranas. Un sistema en Brane es visto como una anidación de membranas y las membranas a su vez se definen como una composición de acciones, que son las que definen el comportamiento de la membrana. Adicionalmente se encuentran definidas unas serie de operaciones que permiten realizar interacciones entre los componentes del Calculo (e.g., replicación, composición).

Sistemas	P, Q $::= \diamond \mid P \circ Q \mid !\,P \mid \sigma(\!\mid P \mid\!)$
Membranas	$\sigma, \tau ::= 0 \mid \sigma \mid \tau \mid !\,\sigma \mid a.\sigma$
Acciones	$a \;::=\; \circledcirc(\sigma) \mid \circlearrowright_n^{\perp}(\sigma) \mid \circlearrowleft_n \mid \circlearrowleft_n^{\perp} \mid \circlearrowright_n$

Cuadro 3.1: Sintaxis Brane

El término $\diamond$ representa el sistema vacio; el operador que representa la composición paralela a nivel de sistemas está representado por el operador $\circ$. El operador de replicación ! denota la composición paralela de varias instancias de un sistema. El término $\sigma(\!\mid P \mid\!)$ denota la membrana que realiza un proceso σ y contiene el sistema P.

El término 0 denota el proceso vacío. Con el operador $\mid$ se denota la ejecución de varios procesos en paralelo; con $!\,\sigma$ se denota al igual que en sistemas la composición paralela de varias instancias del proceso σ. Finalmente el término a.σ es un

proceso compuesto, donde después de ejecutar la acción a el proceso se comporta como σ.

A continuación se definirá la relación de congruencia estructural en sistemas y procesos:

$P \circ Q \equiv Q \circ P$	$\sigma \mid \tau \equiv \tau \mid \sigma$
$P \circ (Q \circ R) \equiv (P \circ Q) \circ R$	$\sigma \mid (\tau \mid \rho) \equiv (\sigma \mid \tau) \mid \rho$
$P \circ \diamond \equiv P$	$\sigma \mid 0 \equiv \sigma$
$!\diamond \equiv \diamond$	$!\,0 \equiv 0$
$!\,(P \circ Q) \equiv !\,P \circ !\,Q$	$!\,(\sigma \mid \tau) \equiv !\,\sigma \mid !\,\tau$
$!\,!\,P \equiv !\,P$	$!\,!\,\sigma \equiv !\,\sigma$
$P \circ !\,P \equiv !\,P$	$\sigma \mid !\,\sigma \equiv !\,\sigma$
$0(\!\lvert \diamond \rvert\!) \equiv \diamond$	

Cuadro 3.2: Congruencia Estructural Brane

Como se ve en el Cuadro 3.3, las reglas (*par*) y (*brane*) son reglas contextuales, es decir que permiten respectivamente a un sistema ejecutar solo si está en paralelo con otro proceso, o si se encuentra dentro de otra membrana, respectivamente. La Regla (*strucong*) asegura que dos sistemas son estructuralmente congruentes cuando tienen las mismas reacciones.

$$(\texttt{par})\ \frac{P \to Q}{P \circ R \to Q \circ R} \qquad (\texttt{brane})\ \frac{P \to Q}{\sigma(\!\lvert P \rvert\!) \to \sigma(\!\lvert Q \rvert\!)}$$

$$(\texttt{strucong})\ \frac{P \equiv P' \quad P \to Q \quad Q \equiv Q'}{P' \to Q'}$$

Cuadro 3.3: Brane: Reacciones Básicas

3.3.2. Interacciones Básicas

Las interacciones básicas del cálculo Brane se encuentran inspiradas en las reacciones biológicas que ocurren a nivel de membranas. (i.e., Fagocitosis, Pinocitosis y Exocitosis). Este tipo de interacciones son básicamente la aceptación o la expulsión de un agente biológico.

Endocitosis y Exocitosis

La interacción mediante la cual una membrana acepta el ingreso de un agente es conocida como Endocitosis, mientras que Exocitosis se le denomina al proceso inverso, es decir, a la expulsión de algún agente al exterior de la membrana.

La endocitosis describe un proceso a través del cual la célula introduce en su interior moléculas o partículas a través de una invaginación de la membrana citoplasmática. Dicho proceso se refiere tanto al ingreso de rial líquido como material no líquido, por tal motivo y para facilidad del estudio, es necesario clasificar la Endocitosis en dos procesos los cuales son determinados por el tipo de agente que ingresa:

- **Fagocitosis:** - Ingreso de agentes no líquidos.
- **Pinocitosis:** - Ingreso de agentes líquidos.

Biológicamente el proceso de fagocitosis consiste en que las células rodean con su membrana citoplasmática a una sustancia extracelular. En este caso, un sólido es introducido al interior de la célula en una nueva membrana resultante en la interacción. Por otro lado la pinocitosis introduce líquidos orgánicos del exterior para ingresar nutrientes o para otra función.

En contraste a la Endocitosis, la Exocitosis es el proceso mediante el cual, las vesículas situadas en el citoplasma se fusionan con la membrana citoplasmática y liberan su contenido.

Para las interacciones Fagocitosis y Exocitosis, en base a lo establecido por Cardelli en el Cálculo Brane [Car98], se puede entender como la sincronización de dos procesos para lograr la interacción. Es decir, existe un sistema sobre el cual recae

una acción y otro sobre el cual recae la coacción como complemento para desencadenar la interacción entre ambos sistemas. En Brane el operador $\searrow_n^{\perp}(\sigma)$ se utiliza para denotar la coacción como vemos en la Figura 3.3. Cabe notar que la pinocitosis no necesita sincronizar con otra membrana para lograr la interacción, por lo tanto esta acción no cuenta con una coacción para su ejecución y puede ocurrir en cualquier momento sin necesidad de sincronización.

Acciones en Brane		
Acción		Descripción
Fagocitosis	Fago: $\searrow_n$	Acción que señala la membrana que ingresa.
	Co-Fago: $\searrow_n^{\perp}$	Co-acción que señala la membrana que recibe.
Exocitosis	Exo: $\nwarrow_n$	Acción que señala la membrana que sale.
	Co-Exo: $\nwarrow_n^{\perp}$	co-acción que señala la membrana expulsada.

Figura 3.3: Acción y Coacción en interacciones básicas entre membranas.

A continuación se analizarán en detalle cada una de las interacciones anteriormente nombradas.

3.3.2.1. Fagocitosis

La interacción mediante la cual una membrana permite el ingreso de otra, es coordinada en Brane por la acción y coacción que recaen sobre las membranas que van a interactuar. Podemos ver inicialmente en la Figura 3.4 la membrana $\searrow_n .\sigma|\sigma_0(\!|P|\!)$ con el subsistema P planteando acción de Fagocitosis a través del operador $\searrow_n$. Este sistema se encuentra interactuando con la membrana $\tau|\tau_0$ con subsistema Q quien plantea una coacción de Fagocitosis para sincronizar la interacción con la otra membrana. Adicionalmente como parámetro tiene la membrana ρ que indica la nueva membrana con la que se procede a realizar la invaginación de la membrana externa que ingresa a la célula.

A continuación las dos membranas realizan un proceso de sincronización a través de la acción con su respectiva co-acción, reduciendo a una sola membrana $\tau|\tau_0$

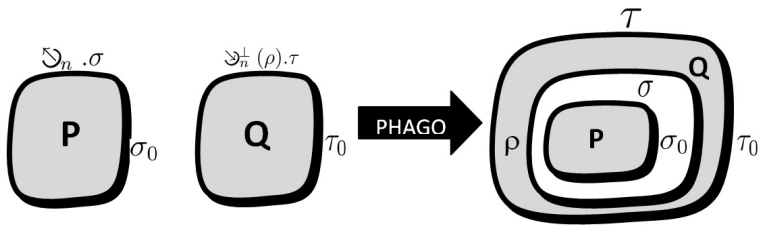

Figura 3.4: Fagocitosis

que ahora no solo contiene al sistema Q sino que internamente se encuentra la membrana ρ y esta a su vez tiene todo lo que contiene la membrana $\mathfrak{Y}_n\ .\sigma|\sigma_0(\!|P|\!)$ Figura 3.4.

La Fagocitosis entre dos membranas τ y σ, donde σ es la membrana que dese ingresar a τ, se define en Brane como:

Definition 1

$$\mathfrak{Y}_n\ .\sigma|\sigma_0(\!|P|\!) \circ\ \mathfrak{Y}_n^{\perp}\ (\rho).\tau|\tau_0(\!|Q|\!) \rightarrow \tau|\tau_0\ (\!|\rho(\!|\sigma|\sigma_0(\!|P|\!)|\!) \circ Q|\!)$$

Cabe aclarar que las acciones $\mathfrak{Y}_n$ y $\mathfrak{Y}_n^{\perp}$ () desaparecen como control en el momento de la reducción.

3.3.2.2. Exocitosis

En la Figura 3.3 se estableció que la acción y coacción en el caso de exocitosis son representados en Brane mediante los operadores $\mathfrak{Y}_n$ y $\mathfrak{Y}_n^{\perp}$ respectivamente. También recordemos que este tipo de interacción implica la mezcla de la membrana que expulsa y la membrana que es expulsada. La Figura 3.5 representa un ejemplo de exocitosis entre las membranas τ y σ, donde σ es la membrana expulsada por τ.

En la Figura 3.5 podemos ver la membrana $\tau|\tau_0$ quien contiene el subsistema Q y la membrana $\sigma|\sigma_0$ quien a su vez contiene al sistema P. Sobre la membrana $\sigma|\sigma_0$ se plantea la acción de exocitosis mediante el operador $\mathfrak{Y}_n$ indicando que la membrana requiere realizar una expulsión. Por otro lado la membrana $\tau|\tau_0$ plantea

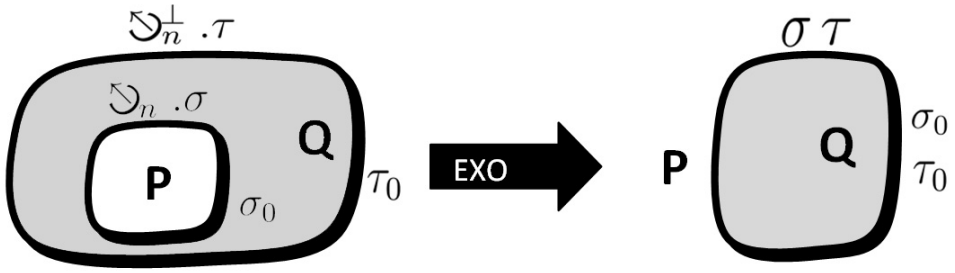

Figura 3.5: Exocitosis

la coacción de exocitosis estableciendo que desea realizar la expulsión de un agente biológico que se encuentre internamente realizando acción exo y sincronizando bajo la variable n, para este caso estamos hablando de la membrana $\sigma|\sigma_0$.

Como resultado de esta interacción se tiene que la membrana que rodeaba al sistema P sobre la cual se ejercía inicialmente la acción de exocitosis ha liberado el sistema P y se ha mezclado con la membrana $\tau|\tau_0$. Es decir que como consecuencia de la interacción de estas membranas tenemos el sistema P en paralelo con la $\tau|\tau_0|\sigma|\sigma_0$ quien contiene el subsistema Q.

El ejemplo de exocitosis explicado anteriormente es reflejado en Brane como:

Definition 2

$$\mathcal{Y}_n^{\perp} .\tau|\tau_0 \ (\!|\ \mathcal{Y}_n .\sigma|\sigma_0(\!|P|\!) \circ Q|\!) \rightarrow P \circ \sigma|\sigma_0|\tau|\tau_0(\!|Q|\!)$$

Se puede apreciar que luego de la reducción el sistema P se encuentra ahora en el exterior de σ. También se puede apreciar la mezcla entre σ y τ representada como la composición entre membranas $\tau|\tau_0|\sigma|\sigma_0$

3.3.2.3. Pinocitosis

En la figura 3.6 podemos ver inicialmente la membrana $\tau|\sigma$ con subsistema P, estableciendo una acción de pinocitosis mediante el operador $\circledcirc$. Cabe aclarar que debido a que este tipo de acción no necesita de una coacción para que ocurra la reacción, entonces siempre puede evolucionar. Adicionalmente vemos que la acción

⊚ tiene como parámetro la membrana ρ que indicará la nueva membrana que se incorporará a la célula.

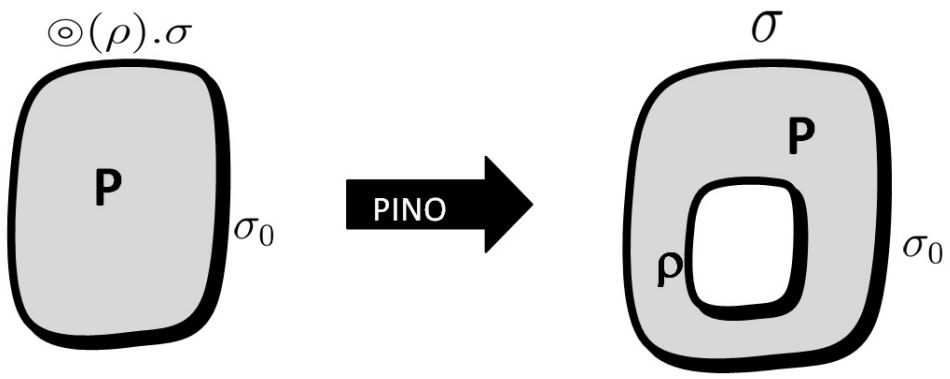

Figura 3.6: Pinocitosis

A continuación se presenta la representación de Pinocitosis en Brane y su respectiva reducción.

Definition 3

$$\circledcirc(\rho).\sigma|\sigma_0(\!|P|\!) \rightarrow \sigma|\sigma_0(\!|\rho(\!|\ |\!) \circ P|\!)$$

Como resultado de la reducción se puede ver la membrana $\sigma|\sigma_0$ quien contiene el subsistema P interactuando en paralelo con ρ (la membrana vacía que se crea en el interior de τ como resultado final de la pinocitosis).

3.3.3. Interacciones Derivadas

En esta sección se presentarán dos nuevas reacciones definidas en el Cálculo Brane, las cuales se derivan de las interacciones básicas ya expuestas anteriormente. Estas son llamadas interacciones derivadas debido a que pueden ser escritas en términos de las interacciones primarias del cálculo. Básicamente son dos tipos de interacciones a saber:

- **Mitosis:** Consiste en el proceso de división celular en donde las células resultantes contienen la misma información genética.

- **Mate:** Proceso por el cual dos células se unen en una sola compartiendo la misma información

Para el proceso de Mitosis, es necesario clasificarlo en 2 procesos para facilitar su estudio (de manera similar a la Endocitosis) y se denominarán Budding y Dripping como se ve en la Figura 3.7 donde la inspiración biológica está dada por un proceso de división celular donde se expulsa una membrana similar con algún tipo de contenido (Bud) y la generación de una nueva membrana en el exterior sin ningún contenido (Drip) respectivamente.

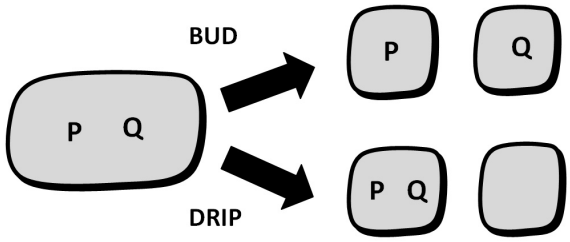

Figura 3.7: Budding (Bud) y Dripping (Drip).

En la Figura 3.8 se puede visualizar de forma completa las fases que ocurrirían para cualquiera de las dos acciones anteriormente planteadas. Cada una de estas fases se representa bajo una de las reacciones básicas (Fago, Exo, Pino).

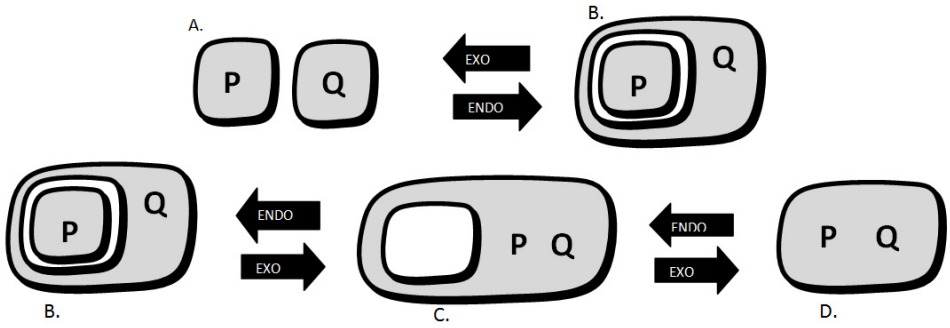

Figura 3.8: Interacciones derivadas

Si analizamos por ejemplo, el proceso de Mate que empieza en el punto A de 3.8 se observa:

1. En la primera interacción ocurre un proceso de Endocitosis (exactamente una Fago) en donde la membrana que contiene el subsistema Q (membrana 2) acepta a la membrana con subsistema P (membrana 1) invaginándola dentro de otra membrana que se especifica como parámetro de la reacción para cumplir las propiedades bitonales planteadas anteriormente, de esta manera se completa el sistema que se tiene en el punto B.

2. Luego en B ocurre una exocitosis entre la membrana que inicialmente denotamos como 1 y la nueva membrana que ahora la contiene, al realizar esta interacción el sistema P sale del sistema inmediatamente superior, logrando una única membrana vacía interactuando al mismo nivel de P y Q siendo contenidas bajo la inicial membrana 2 como se puede ver en el punto C.

3. Finalmente, en C ocurre nuevamente una exocitosis entre la membrana interna (que antes contenía a P) y la membrana 2 logrando finalmente la expulsión del agente biológico, de esta manera tenemos una única membrana con subsistemas P y Q interactuando al mismo nivel, completando así el proceso denominado Mate en el punto D.

De igual manera se puede observar el proceso de Mitosis, en este caso Budding, si partimos desde el punto D (una sola membrana con subsistemas P y Q), luego por los puntos C, B y finalmente terminando en el punto A, donde se tiene dos membranas independientes con características semejantes, una con subsistema P y la otra con subsistema Q.

3.3.3.1. Mate

Al igual que en las reacciones básicas descritas a lo largo de este capítulo, es necesario tener una coacción que permita realizar la sincronización las dos membranas. Para este caso, como vemos en la figura 3.9, tenemos inicialmente la membrana $\sigma|\sigma_0$ estableciendo acción de mate y en paralelo la membrana $\tau|\tau_0$ con la coacción de mate bajo el operador $mate^{\perp}$ y logrando sincronización mediante la variable n, lo cual se refleja en reducción que se presenta más adelante, al tener como resultado de esta interacción a una única membrana $\tau|\tau_0|\sigma|\sigma_0$ quien contiene a los sistemas P y Q.

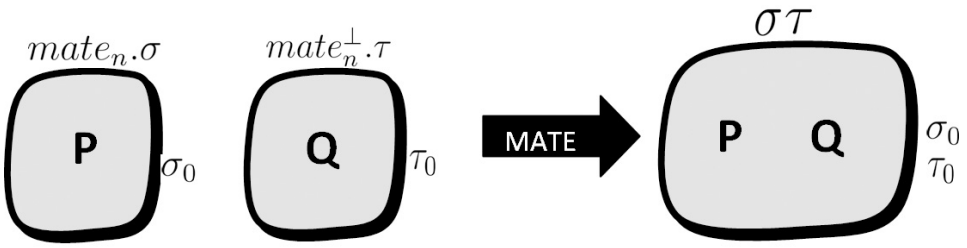

Figura 3.9: Interacción derivada: Mate

Para proceder a explicar el proceso Mate de acuerdo al Cálculo Brane, primero se presenta la definición de acción y coacción para Mate. Luego se presenta la reducción de dicha interacción como resultado de ocurrir un proceso de Fagocitosis seguido luego por un proceso de Exocitosis y finalmente otro proceso de Exocitosis.

Acción y Coacción Mate

Para definir la acción *mate* y $mate^{\perp}$, debemos tener en cuenta que esta se definió con base en las reacciones primitivas del cálculo, de esta manera tenemos:

$$\begin{array}{rcl} \text{mate}_n.\sigma & ::= & \circlearrowright_n .\circlearrowleft_{n'} .\sigma \\ \text{mate}_n^{\perp}.\tau & ::= & \circlearrowright_n^{\perp} (\circlearrowleft_{n'}^{\perp} .\circlearrowleft_{n''}). \circlearrowleft_{n''}^{\perp} .\tau \end{array}$$

Cuadro 3.4: Mate: acción y co-acción

Reducción del Proceso Mate

Cuando dos membranas se encuentren interactuando a un mismo nivel y tengan acciones de mate y co-mate respectivamente, debe ocurrir:

$$mate_n.\sigma|\sigma_0(\!|P|\!) \circ \ mate_n^{\perp}.\tau|\tau_0(\!|Q|\!) \rightarrow \sigma|\sigma_o|\tau|\tau_0(\!|P \circ Q|\!)$$

La reducción anterior se puede obtener mediante la reducción en secuencia de tres interacciones básicas, de acuerdo a las definiciones de mate y co-mate:

$$
\begin{aligned}
mate_n.\sigma|\sigma_0(\!|P|\!) \circ\ mate_n^{\perp}.\tau|\tau_0(\!|Q|\!) :=\ & \\
& \Im_n\ .\Im_{n'}\ .\sigma|\sigma_0(\!|P|\!) \circ \Im_n^{\perp}\ (\Im_{n'}^{\perp}\ .\Im_{n''})\ .\Im_{n''}^{\perp}\ .\tau|\tau_0(\!|Q|\!) \Rightarrow \text{Phago n} \\
& \Im_{n''}^{\perp}\ .\tau|\tau_0(\!|\Im_{n'}^{\perp}\ .\Im_{n''}\ (\!|\Im_{n'}\ .\sigma|\sigma_0(\!|P|\!)|\!) \circ Q|\!) \Rightarrow \text{Exo n} \\
& \Im_{n''}^{\perp}\ .\tau|\tau_0(\!|\Im_{n''}\ \sigma|\sigma_0(\!|\diamond|\!) \circ P \circ Q|\!) \Rightarrow \text{Exo n} \\
& \sigma|\sigma_0|\tau|\tau_0(\!|P \circ Q|\!)
\end{aligned}
$$

3.3.3.2. Budding

El proceso de Budding ocurre gracias a una ejecución secuencial de los procesos Pinocitosis, Fagocitos y por último una Exocitosis. Como se ha planteado anteriormente en todas las reacciones propias del Brane Calculi, primero debe ocurrir una sincronización entre la acción de bud_n y la coacción y a continuación la nueva membrana ρ acepta a la membrana interna $\sigma|\sigma_0$ y finalmente es expulsada de la membrana $\tau|\tau_0$ como podemos ver en 3.10

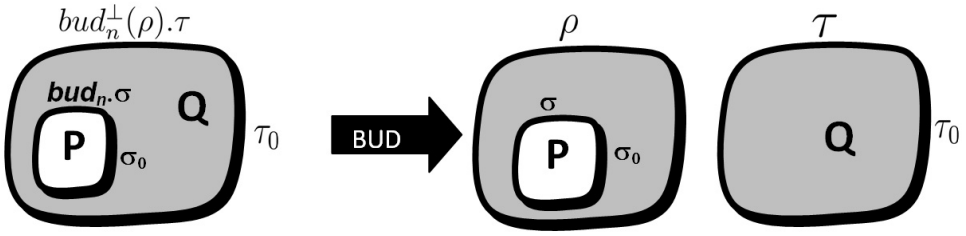

Figura 3.10: Interacciones derivadas

De igual manera como se definió en Mate (Cuadro 3.4), se presentará a continuación la definición de la acción y coacción correspondientes a la interacción Budding. Seguidamente se presenta la reducción en base a las interacciones básicas que conforman el proceso Budding.

Acción y Coacción Budding

A continuación se definen las acciones bud_n y la coacción ρ como:

En la coacción $bud_n^{\perp}\rho$, se puede notar el parámetro ρ que representa el nombre de la nueva membrana que resulta de la interacción.

$$\begin{array}{l} \mathrm{bud}_n.\sigma ::= \ \circlearrowright_n .\sigma \\ \mathrm{bud}_n^{\perp}(\rho).\tau ::= \circledcirc(\circlearrowright_n^{\perp}(\rho).\circlearrowleft_{n'}).\circlearrowleft_{n'}^{\perp}.\tau \end{array}$$

Cuadro 3.5: Budding: acción y co-acción

Reducción del Proceso Budding

Cuando dos membranas se encuentren interactuando a un mismo nivel y tengan acciones budding y co-budding respectivamente, debe ocurrir lo siguiente:

$bud_n^{\perp}(\rho).\tau|\tau_0(\!|bud_n.\sigma|\sigma_o(\!|P|\!) \circ Q|\!) \rightarrow \rho(\!|\sigma|\sigma_0(\!|P|\!)|\!) \circ \tau|\tau_0(\!|Q|\!)$

A continuación se presenta la reducción en base a las interacciones básicas (como resultado de la reducción en secuencia de Pinocitosis, fagocitosis y exocitosis, en el respectivo orden):

$$\begin{array}{l} bud_n^{\perp}(\rho).\tau|\tau_0(\!|bud_n.\sigma|\sigma_o(\!|P|\!) \circ Q|\!) \ := \\ \qquad \circledcirc(\circlearrowright_n^{\perp}(\rho).\circlearrowleft_{n'})\circlearrowleft_{n'}^{\perp}.\tau|\tau_0(\!|\circlearrowright_n .\sigma|\sigma_0(\!|P|\!) \circ Q|\!) \Rightarrow \text{Pino} \\ \qquad \circlearrowleft_{n'}^{\perp}.\tau|\tau_0(\!|\circlearrowright_n^{\perp}(\rho).\circlearrowleft_{n''}(\!|\diamond|\!) \circ \circlearrowright_n .\sigma|\sigma_0(\!|P|\!)Q|\!) \Rightarrow \text{Phago n"} \\ \qquad \circlearrowleft_{n''}^{\perp}.\tau|\tau_0(\!|\circlearrowleft_{n'}(\!|\rho(\!|\sigma|\sigma_0(\!|P|\!)|\!)|\!) \circ Q|\!) \Rightarrow \text{Exo n} \\ \qquad \rho(\!|\sigma|\sigma_0(\!|P|\!)|\!) \circ \tau|\tau_0(\!|Q|\!) \end{array}$$

3.3.3.3. Dripping

La interacción Dripping ocurre gracias a una ejecución secuencial de dos Pinocitosis, y finalmente una exocitosis. Cabe aclarar que esta reacción no requiere de una coacción para su ejecución, ya que no es necesaria la sincronización con otra membrana para poder evolucionar a un próximo estado.

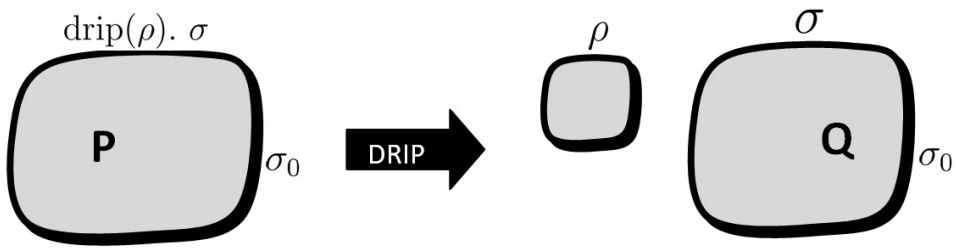

Figura 3.11: Interacción derivada: Dripping

En la Figura 3.11 se puede observar que como resultado de la interacción se tiene una nueva membrana de características similares, pero sin contenido alguno.

Acción y Coacción Dripping

A continuación la definición de la acción Dripping y su correspondiente reducción según el Cálculo Brane:

$$\text{drip}(\rho).\ \sigma := \circledcirc(\circledcirc(\rho).\mathfrak{S}_n).\mathfrak{S}_n^{\perp}\ .\sigma$$

Cuadro 3.6: Drip: acción y co-acción

Reducción del Proceso Budding

A continuación se presenta la reducción de la interacción Dripping en base a las reducciones de Pinocitosis y Exocitosis:

$$\begin{aligned} drip(\rho).\sigma|\sigma_o(\!|P|\!) \ :=& \\ & \circledcirc(\circledcirc(\rho).\mathfrak{S}_n).\mathfrak{S}_n^{\perp}\ .\sigma|\sigma_0(\!|P|\!) \Rightarrow \text{Pino} \\ & \mathfrak{S}_n^{\perp}\ .\sigma|\sigma_0(\!|\circledcirc(\rho).\mathfrak{S}_n\ (\!|\diamond|\!)\circ P|\!) \Rightarrow \text{Pino} \\ & \mathfrak{S}_n^{\perp}\ .\sigma|\sigma_0(\!|\mathfrak{S}_n\ (\!|\rho(\!|\diamond|\!)|\!)\circ P|\!) \Rightarrow \text{Exo n} \\ & \rho(\!|\,|\!)\circ\sigma|\sigma_0(\!|P|\!) \end{aligned}$$

3.4. Resumen

En este capítulo se presentó el Modelo de Computación con membranas, el cual se enfoca en todas las reacciones e interacciones que ocurren a nivel de la membrana celular, es decir analiza los mecanismos por los cuales la célula se comunica con su exterior a través de la membrana. Posteriormente se hizo una breve descripción de este tipo de modelos y de cómo procesos que ocurren de forma natural en la célula inspiran nuevos paradigmas de computación.

A continuación describimos BioAmbients, un cálculo bioinspirado y enfocado principalmente en la interacción entre membranas en donde los procesos residen dentro de una jerarquía de locaciones llamados ambientes, logrando así una configuración espacial que permite modelar movilidad a través del cambio de ambientes que pueden realizar los procesos durante el tiempo.

Posteriormente se presentó el cálculo Brane como otro de los cálculos centrados en el modelado de las interacciones a través de las membranas biológicas. Se plantearon algunas propiedades básicas del cálculo, sus componentes fundamentales, así como su sintaxis. Luego se establecieron las interacciones básicas del cálculo (fagocitosis, pinocitosis y exocitosis), inicialmente desde un contexto biológico para luego llegar a su equivalente computacional en el cálculo. Con base en estas interacciones básicas, se presentaron interacciones más complejas (Mate y Mito) como una derivación de las tres básicas.

Todo lo presentado en éste capítulo representa el objeto de estudio del presente trabajo de grado. Lo que se pretende es realizar un estudio exploratorio sobre este campo de la biología, utilizando como herramienta la Programación Concurrente por Restricciones para el modelado y representación de las interacciones que ocurren a nivel de membrana celular. De este modo, lo expuesto anteriormente es un marco de referencia para el siguiente capítulo en donde se presentará una interpretación en Utcc de los procesos ya descritos.

4 Interpretación de Cálculos Biológicos en CCP

En este capítulo se utilizará el cálculo Utcc para llevar a cabo la interpretación de membranas y sus interacciones con base en las definiciones del Cálculo Brane. Inicialmente se define un sistema de restricciones que está centrado en relaciones entre variables que representarán asociaciones de las membranas. El sistema de restricciones ya definido será la base para la representación del comportamiento de las membranas. Se definirán las interacciones básicas: fagocitosis, pinocitosis y exocitosis, que luego serán fundamentales para la representación de interacciones derivadas: Mate, Budding y Dripping.

4.1. Descripción de la Interpretación de Membranas

Las membranas se consideran como delimitantes de espacios o ambientes, donde pueden ocurrir procesos (de igual manera que en BioAmbients). Anteriormente se dió a conocer las interacciones que ocurren entre membranas, y dichas interacciones implican un cambio de configuración en este tipo de sistemas. Es decir, si un proceso P ocurre en alguna membrana τ anidada en otra membrana σ (ver Figura 4.1) y se desarrolla un proceso de exocitosis entre σ y τ, implicaría que P ya no ocurrirá en el interior de τ ni en el de σ como sucedía en un estado inicial. Estos cambios de configuración en este tipo de sistemas son el objeto de estudio del presente capítulo.

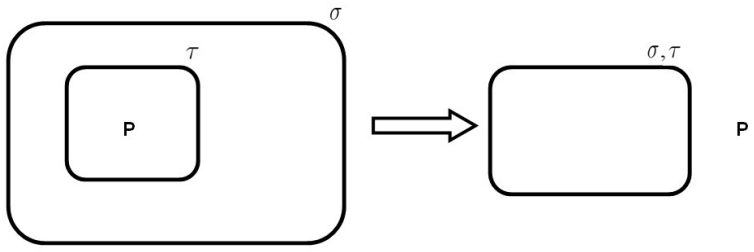

Figura 4.1: Exocitosis entre σ y τ

De ahora en adelante para hacer referencia a la configuración de un sistema, hablaremos de alcance de un sistema. El alcance de un sistema P estará comprendido por las membranas en las cuales se encuentre anidado. En el caso de la Figura 4.1, tendríamos que el sistema P tiene alcance a τ y σ, también podríamos considerar el alcance del sistema τ, donde τ solo tendría alcance a σ.

Ahora se procede a explicar detalladamente cómo se llevará a cabo la representación de membranas. Las membranas se representarán con letras del alfabeto griego y el alcance que tiene un sistema o membrana será representado por un *conjunto S asociado* (S es un conjunto de variables que representan membranas). Cabe resaltar que estas variables para identificar membranas son únicos. El conjunto S_x asociado a la membrana x estará conformado por membranas que se encuentran a un nivel superior de x. En la Figura 4.2 se presenta como ejemplo la representación de un sistema conformado por membranas. En este caso el conjunto asociado a la membrana α será $S_\alpha = \{\ \beta,\sigma\}$. Se puede notar que τ no hace parte del conjunto S_α, ya que se estableció que el alcance se tenía si la membrana en cuestión representaba una anidación para la membrana a incluir en el conjunto asociado. Si tomamos ahora como referencia la membrana τ, su conjunto asociado sería $S_\tau = \{\sigma\}$, ya que τ representa una anidación para σ.

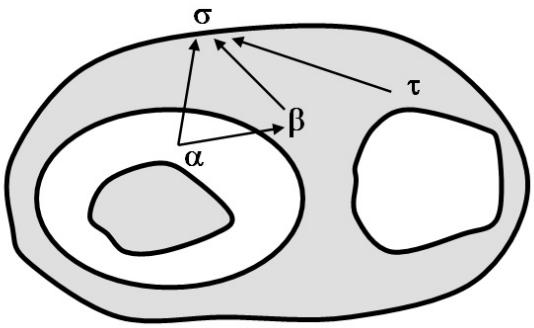

Figura 4.2: Alcances en un sistemas conformado por membranas

Luego de tener definido el concepto de alcance que se utilizará en éste trabajo, es necesario aclarar cuándo es posible la interacción entre dos sistemas. Para que un sistema o membrana A pueda interactuar con un sistema B, se necesita alguna de las siguientes condiciones:

1. Que ambos sistemas se encuentren a un mismo nivel, es decir que ninguno sea anidación del otro y se encuentren dentro de un mismo sistema (e.g., β y τ se encuentran al mismo nivel en Figura 4.2).

2. Que A sea un subsistema inmediato de B. Cuando se habla de subsistema inmediato se está haciendo referencia a una membrana que se encuentra a un solo nivel de anidación, por ejemplo, para el caso de la Figura 4.2, se tiene que τ es subsistema inmediato de σ, también se tiene β como subsistema inmediato de σ pero α no es subsistema inmediato de σ, debido a que α se encuentra a dos niveles de anidación de σ, es decir, se tendría primero que pasar por β para llegar al sistema α. La forma en que se representarán las relaciones entre sistema y subsistema será posible mediante la definición del sistema de restricciones en la siguiente sección.

Adicional a la relación entre la membrana y su conjunto S para representar el alcance, también se hace necesario otro tipo de asociaciones a las membranas. Cada relación entre membrana y su conjunto será asociada a un único identificador, es decir, existe un Id para cada sistema al cual se le asocia una variable que representa el nombre de la membrana y también se relaciona el conjunto asociado a

dicha membrana con el identificador, dicha relación se puede apreciar en la Figura 4.3.

Cabe aclarar que se asume que todo sistema que se defina, se encuentra contenido en una membrana global denominada *Membrana Universo U* (Observación 1).

Observación 1. *Membrana Universo U. Se establecerá la membrana universo U y su conjunto asociado S_U, donde U contiene cualquier sistema que se defina. No puede existir ningún sistema en el exterior de U y ningún sistema contenido en U estará en capacidad de interactuar con la membrana U.*

4.2. Descripción del Sistema de Restricciones

Las relaciones entre las membranas previamente mencionadas, serán establecidas mediante el sistema de restricciones que se define en esta sección. Anteriormente se mencionaba el concepto *subsistema inmediato* de un sistema cuando se hacía referencia a un sistema anidado en otro, ahora se podrá representar dicha relación mediante restricciones como *in(x,y)*, que indica que la membrana o sistema x se encuentra anidada en la membrana y. En el caso de la Figura 4.2, se tendrían las siguientes relaciones entre sistemas y subsistemas inmediatos: *in(α,β)*, *in(β,σ)* y *in(τ,σ)*. Se puede notar que para el caso anterior no es posible establecer la restricción *in(α,σ)*, pero es posible definir reglas o propiedades que permitan hacer deducciones, en este caso sobre las restricciónes *in(x,y)*, con las cuales se puede llegar a asegurar que α es un subsistema de σ partiendo de *in(α,β)* (α *es subsistema inmediato* de β) y *in(β,σ)* (β *es subsistema inmediato* de σ) lo que permite razonar que α al estar contenido en β y β al estar contenido en σ implica que α está contenido en un segundo nivel dentro de σ.

Se mencionaron también otro tipo de relaciones entre las membranas y algunas variables que son necesarias para coordinar las interacciones entre sí, ahora veamos cómo se establecen éstas relaciones basadas en el sistema de restricciones. Primero se asocia un identificador único al sistema (identificador asociado a la variable membrana y al conjunto asociado, Figura 4.3) mediante las restricciones *idmem(σ,Id)* y *idset(Id,S_σ)*, la primera asocia la variable *Id* con la membrana σ y la segunda asocia el mismo Id con el conjunto S_σ, donde S_σ representa el conjunto

asociado a σ. Según lo anterior podemos ver que la relación existente entre σ y su conjunto asociado S_σ es posible mediante un único identificador común entre la membrana y el conjunto asociado (ver Figura 4.3).

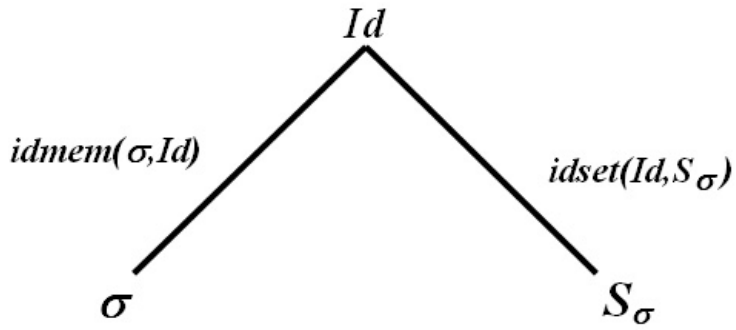

Figura 4.3: Relación entre membrana σ y conjunto asociado S_σ.

Ahora es necesario establecer otras relaciones entre variables y las membranas, donde dichas variables determinarán el comportamiento que adopta una membrana y la coordinación entre dos membranas que desean interactuar. Estas relaciones serán representadas mediante las restricciones *behavior(σ,r)* y *synchro(σ,n)* respectivamente.

La restricción *behavior(σ,r)* establece el tipo de comportamiento que adopta la membrana al ocurrir una interacción. Es decir, para el caso de una fagocitosis, determinará si σ es la membrana que ingresa o la membrana que recibe. Para este fin r será una variable de tipo booleano y su valor determinará el comportamiento de la membrana σ.

En restricción *synchro(σ,n)*, la variable n actúa como identificador de una acción y permite la sincronización entre dos membranas que van a interactuar. Por ejemplo, para que sea posible la interacción entre la membrana σ y la membrana τ, se necesita que existan las relaciones *synchro(σ,n_1)* y *synchro(τ,n_2)*, y también que pueda inferirse del store que $n_1 = n_2$. Es decir, la membrana σ solo podrá interactuar con otra membrana cuya variable asociada tenga el mismo valor que n.

También se establecen restricciones como *merge(σ,τ)*, *channel(σ,τ)* y $\sigma \in$ S. La restricción *merge(σ,τ)* hace alusión a la mezcla entre membranas como resultado de una Exocitosis, condición que ya se explicó en capítulos anteriores. La restricción *channel(σ,τ)* actúa de manera similar a *in(σ,τ)* solo que será necesaria en el momento de representar una Fagocitosis. Por otra parte la restricción $\sigma \in$ S, indica que σ pertenece al conjunto S, donde σ es membrana y S algún conjunto asociado.

Las restricciones que ya se mencionaron permiten de algún modo asociar características propias a cada membrana, que determinan cuál debe ser su comportamiento. Pero adicionalmente se hace necesario crear restricciones que permitan simular estímulos del medio u órdenes que determinen el momento justo para iniciar determinada interacción. Dichas restricciones son identificadas bajo los nombres de *phago, receive, finish, exo, leave* y *pino*, las cuales al poder ser deducidas del store activan el comportamiento de acuerdo al estímulo recibido. Este tipo de restricciones son explicadas con más detalle en el modelado de las interacciones básicas, donde se aclara la función que desempeña cada parámetro en cada restricción y en cada interacción.

A continuación se presenta resumido el sistema de restricciones en la *Tabla de Restricciones*, donde se define de manera breve el significado de cada restricción y algunas condiciones necesarias para lograr la representación de las membranas y sus interacciones.

Restricción	**Descripción**
idmem(σ,id)	Asocia el identificador id a la membrana σ. id es único y está compartido por la membrana y su conjunto asociado.
idset(id,S)	Asocia el identificador id al conjunto de membranas S.
behavior(σ,b)	Asocia la variable de comportamiento b a la membrana σ. La variable b es variable de tipo booleano y determina el tipo de comportamiento de la membrana, por ejemplo en caso de una fagocitosis, determina si σ es la membrana que ingresa o la que recibe.
synchro(σ,m)	Asocia la variable de coordinación m a la membrana σ. La variable m coordina la interacción entre dos membranas, para lo cual el valor de las variables de coordinación debe ser el mismo entre ambas membranas y no repetirse en alguna otra interacción.
channel(σ,τ)	Establece un canal entre la membrana σ que contiene a la membrana τ. Se utiliza para compartir la variable membrana en caso de una fagocitosis.
in(σ,τ)	Establece la relación de anidación entre membranas. Indica que la membrana τ es un subsistema inmediato de la membrana σ.
$\sigma \in$ S	Indica que la membrana σ hace parte del conjunto de membranas S.
finish(n)	Indica que la interacción coordinada mediante las varibles de coordinación cuyo valor es n ha finalizado
phago(n,ρ,b)	Estímulo del medio para desencadenar una fagocitosis entre dos membranas, cuyas variables de coordinación tienen como valor n. El parámetro ρ indica el nombre de la nueva membrana que resulta por el proceso de fagocitosis. El tercer parámetro es de tipo booleano, su importancia se explica más adelante en la interpretación de las interacciones básicas.
receive(σ,ρ',m,b)	Restricción que indica la señal para que se active la membrana que desea ingresar en un proceso de fagocitosis. El parámetro σ referencia a la membrana que recibe, para compartirla a la membrana que ingresa, de igual modo ρ' representa la nueva membrana que se origina y que también debe ser compartida. Finalmente m es la variable de coordinación de σ y b es de tipo booleano, su importancia se explica más adelante en la interpretación de las interacciones básicas.

Restricción	Descripción
exo(n,b)	Estímulo del medio para desencadenar una exocitosis entre las membranas cuya variable de coordinación tengan como valor n, b es de tipo booleano, su importancia se explica más adelante en la interpretación de las interacciones básicas.
leave(m,σ,b)	Activa a la membrana que expulsa, comunicando el valor de la variable de coordinación m de la variable que sale; también comunica la membrana que sale σ y b es de tipo booleano, su importancia se explica más adelante en la interpretación de las interacciones básicas.
merge(σ,τ)	Indica que la membrana τ que antes era un subsistema inmediato de la membrana σ, ahora está al mismo nivel de σ. Representa el resultado de una exocitosis.
pino(ρ, id_σ)	Estímulo que indica a la variable identificada por id_σ que debe realizar pinocitosis, y la nueva membrana que se crea en su interior debe ser ρ.

Tabla de Restricciones

Teniendo definido el sistema de restricciones, se puede tomar como base para establecer algunas afirmaciones que permitirán realizar deducciones sobre los sistemas, como por ejemplo, tener conocimiento de un sistema que sea subsistema inmediato o no inmediato de otro sistema.

Definition 4 *Subsistema Inmediato y Subsistema No Inmediato.*
x es subsistema inmediato de y, si se puede deducir la condición in(x,y) del store.
x es subsistema no inmediato de y, si $\exists_{z_1,z_2,\ldots,z_n}$ *tal que sea posible deducir del store: in(x,* z_1*) , in(z1,* z_2*), ,in(*z_n *,y).*

4.3. Interpretación del Cálculo Brane en Utcc

En este momento es importante resaltar la *movilidad* como una de las características de Utcc que hacen posible este estudio exploratorio, ya que la interacción entre membranas se basará en la comunicación entre distintos sistemas que trabajan sobre variables locales para coordinar la interacción entre sí. Más adelante se explicará la representación de las membranas por medio de variables locales para un conjunto de procesos que representan el comportamiento natural de una membrana, además mostraremos cómo la movilidad permitida por Utcc puede dar solución al problema de la coordinación entre sistemas.

Para lograr esta interpretación se hace necesario definir la función [[.]] la cual tomará sistemas definidos en Brane para lograr la representación de los mismos sistemas pero en Utcc (Definición 5).

Definition 5 *Enconding de Brane a Utcc.*
La función [[.]] mapea procesos modelados en Brane a representaciones en Utcc y se define como:

$$[[\ .\ ]] = \textbf{\textit{Sistema Brane}}\ X\ S\ X\ Id \rightarrow \textbf{\textit{Proceso Utcc}}$$

- ***Sistema Brane****: Sistema basado en membranas representado en sintaxis de Brane.*
- ***S****: Conjunto asociado de la membrana superior.*
- ***Id****: Identificador único del sistema.*

La *Definición 5* establece que [[.]] recibe como entrada tres parámetros: un sistema expresado en sintaxis del cálculo Brane, el conjunto asociado de la membrana superior y una llave única para identificar el sistema a interpretar; lo cual es información suficiente para retornar una representación de procesos en Utcc los cuales simulan el comportamiento de los procesos en Brane.

4.3.1. Traducciones Básicas

A continuación se definirán las traducciones básicas de sintaxis Brane a Utcc, se toma como base la definición de la sintaxis del cálculo Brane mencionada en el capítulo anterior.

1. Proceso Vacío

$[[\diamond, S, Id]] = skip$

El cálculo Brane define el operador $\diamond$ como un sistema vacío. Este operador como parámetro junto con cualquier sistema asociado S y un identificador reducirá a *skip* en Utcc que al igual que en Brane hace alusión a la ausencia de un proceso o sistema.

2. Proceso Simbólico

$$[[P , S, Id]] = \; tell\ idset(Id,S_P) \\ \|(abs\ x,y;idset(y,S) \wedge idmem(x,y))((!tell\ x{\in}S_P \ \|!Dentro(x,S_P)))$$

En la sintaxis de Brane se utilizan los símbolos P y Q entre otros, para simbolizar la existencia de un sistema cualquiera el cual no tiene una representación definida. En éste caso es necesario asociar a dicho sistema simbólico P un conjunto asociado S_P por medio de un identificador utilizando la restricción *idset(Id,S_P)*, donde la variable Id ingresa como parámetro de la función [[.]], luego se hace necesario construir el conjunto asociado S_P ya que P siempre va a ser un sistema anidado o contenido dentro de otro sistema, teniendo en cuenta que cualquier sistema a definir por más global o externo que sea, éste se encuentra contenido en el *Universo*. Para conformar el conjunto asociado S_P es necesario partir de la membrana inmediatamente superior a P, dicha membrana sería la que se encuentra asociada al conjunto S que ingresa como segundo parámetro en la traducción. La extracción de la membrana superior a P se logra mediante la consulta *(abs x,y;idset(y,S)* $\wedge$ *idmem(x,y))*, la cual abstrae la membrana que se encuentra asociada al identificador y que a su vez se asocia al conjunto S. Luego de tener el primer elemento de S_P, se utiliza el proceso recursivo *Dentro(x,S_P)* que se basa en una serie de

deducciones transitivas para conformar el conjunto de membranas que se encuentran por encima de P. La definición del proceso *Dentro(x,S_P)* se presenta más adelante en ésta sección.

3. Composición Paralela de Sistemas

[[P ∘ Q, S, Id]] = [[P ,S, id_P]] ‖ [[Q ,S, id_Q]]

En Brane dos membranas o sistemas P y Q interactuando a un mismo nivel serán representadas por medio del operador ∘. En términos de CCP se podría considerar los sistemas P y Q como dos procesos en ejecución paralela, para lo cual se establece que la reducción en este caso utilizará el operador ‖ de Utcc para representar dicha concurrencia. Para la definición de este caso, se puede observar que se utiliza *(2) Proceso Simbólico*.

4. Representación General de una Membrana

$$
\begin{aligned}
[[\ \sigma\ (\!|P|\!),S,Id\ ]] = {} & local\ \sigma\ in\ (\ (abs\ x,y;idset(y,S) \wedge idmem(x,y)) \\
& (\ tell(in(\sigma,x)) \parallel tell\ channel(\sigma,x)) \parallel\ !Dentro(\sigma,S_\sigma) \\
& \parallel tell\ idset(Id,S_\sigma) \parallel tell\ behavior(\sigma,B_\sigma) \parallel tell\ synchro(\sigma,A_\sigma) \\
& \parallel tell\ idmem(\sigma,Id) \parallel CASE_\sigma \parallel [[\ P,S_\sigma,Id_P\ ]])
\end{aligned}
$$

Ahora vemos cómo se logra la traducción de una membrana, generalizando el sistema como una membrana σ que contiene a su vez otro sistema simbolizado P. La variable σ es el nombre de la membrana, representándose como una variable local a una serie de procesos que obedecen a la descripción de la representación que se va a realizar (*mencionado en la Sección 4.1*). Recuerde que es necesario asociar un identificador común entre σ y un conjunto asociado S_σ mediante las restricciones *idset(Id,S_σ)* y *idmem(σ,Id)*, luego asociar las variables de estado y de coordinación. Recorde también que la variable de estado B_σ es de tipo booleano y su valor determinará el tipo de comportamiento de la membrana σ; la variable de coordinación A_σ determina con cuál membrana se llevará a cabo la interacción con σ. Para conformar luego el conjutno asociado S_σ, es necesario recurrir a los pasos que se utilizaron en *(2)*, primero abstraer la membrana superior, es decir la membrana para la cual σ es un subsistema inmediato (agregando al *store* la restricción

in(σ,x)) y luego a partir de dicha membrana empezar a construir S_σ con ayuda del proceso *Dentro(σ,S_σ)*. Para finalizar se hace llamado del proceso $CASE_\sigma$, donde dicho proceso modela el comportamiento natural de una membrana, es decir $CASE_\sigma$ refleja la capacidad que tiene una membrana de interactuar en el caso de fagocitosis, exocitosis o pinocitosis. La definición del proceso *CASE* es natural a toda membrana y se presentará más adelante.

En las definiciones anteriores *Proceso Simbólico (2)* y *Representación General de una Membrana (4)*, se hizo llamado del proceso recursivo *Dentro(σ,S_σ)* como apoyo para la construcción de los conjuntos asociados. Este proceso utiliza deducciones transitivas de la restricción *in(x,y)* para obtener cada una de las membranas que hace parte del conjunto asociado que ingresa como parámetro. A continuación se define el proceso *Dentro(σ,S_σ)* como:

Definition 6 *Dentro(σ,S_σ). Conformación del Conjunto Asociado.*

$$Dentro(\sigma,S_\sigma) \stackrel{def}{=} (absw; in(\sigma,w))(tell\ w \in S_\sigma \parallel Dentro(w,S_\sigma))$$

*El proceso recibe como parámetros la variable que representa una membrana y su correspondiente conjunto asociado. Para generar el conjunto asociado S_a (alcances de σ), se utiliza como punto de partida la abstracción de la membrana inmediatamente superior por medio del proceso (*abs w; in(σ,w)*), que se puede interpretar como: obtenga la membrana w tal que se cumpla la condición* in(σ,w)*. De esta manera se tiene acceso a la membrana que contiene a σ como* subsistema inmediato*, la cual es adicionada en los alcances de σ por medio de la restricción* $w \in S_\sigma$*. Para finalizar se utiliza el proceso recursivamente, pero cambiando el parámetro σ por la membrana w con el fin de continuar escalando en la jerarquía del sistema y obtener las otras membranas superiores a σ . El segundo parámetro en el llamado recursivo continúa siendo S_σ, ya que representa el los alcances de σ y la finalidad del proceso Dentro es completar el conjunto S_σ.*

Para facilitar la comprensión del proceso *Dentro(σ,S_σ)*, retomemos el caso de la *Figura 4.2* y como base las restricciones tipo *in(x,y)*. En esta ocasión se contaría en el *store* con las restricciones: *in(α,β)*, *in(β,σ)*, *in(τ,σ)* y *in(σ,U)*. Supongamos

ahora que se quiere construir el conjunto asociado S_α, para lo cual se hace llamado del proceso *Dentro(α,S_α)*. Primero se obtiene la membrana superior a α por medio de *(abs w; in(α,w))*, abstrayendo sobre la variable w a la membrana β y luego agregándola al conjunto mediante la restricción $w \in S_\alpha$, luego con el llamado recursivo *Dentro(β,S_α)* adicionaría la membrana σ al conjunto S_α ($\sigma \in S_\alpha$). Finalmente se haría nuevo llamado recursivo de la forma *Dentro(σ,S_α)*, lo que terminaría agregando la *membrana universo* al conjunto S_α luego de deducir *in(σ,U)*. Como resultado final se tendría el conjunto $S_\alpha = \{\beta,\sigma,U\}$, lo que indica que α tiene alcance a β, σ y U. Nótese que se obvió el llamado recursivo *Dentro(U,S_α)*, ya que en este caso no se podría deducir ninguna membrana superior a U, es decir, no existe *in(U,w)*.

Las representaciones de los sistemas conformados por membranas definidos en *(1)*, *(2)*, *(3)* y *(4)*, obedecen la descripción de la sección anterior. Vamos entonces a empezar a modelar las interacciones básicas que ocurren entre membranas: *fagocitosis*, *exocitosis y pinocitosis*. Con éste fin se procede a explicar la fagocitosis y la exocitosis en dos partes. Primero se define el modelo de las acciones, es decir de los procesos que producen estímulos en el ambiente para desencadenar una interacción y como segunda medida, se tiene la definición del comportamiento de las membranas de acuerdo a los estímulos del medio. Para la representación del comportamiento natural o respuesta de acuerdo a los estímulos, se define a continuación el proceso *CASE* ya mencionado con anterioridad en *(4)*, que simula el comportamiento de cualquier membrana.

4.4. Interpretación de Interacciones Básicas entre Membranas: Fagocitosis, Exocitosis y Pinocitosis

En la Sección 4.3.1 se establecieron las representaciones básicas de sistemas conformados por membranas, ahora se va a definir cómo se deben interpretar los operadores de acciones y coacciones del Cálculo Brane: $\circledS_n$, $\circledS_n^\perp(\sigma)$, $\circledcirc_n$, $\circledcirc_n^\perp$ y $\circledcirc(\sigma)$.

No olvidemos que las interacciones básicas se interpretan como la coordinación

entre dos membranas por medio de operadores de acción y coacción (*Acción y Coacción en interacciones básicas entre membranas, Tabla 4 del Capítulo 3*). Por tal motivo, de ahora en adelante para hacer referencia a las acciones que indican fagocitosis, éstas serán llamadas como *Fago* y *Co-fago*, para la exocitosis se denominarán *Exo* y *Co-exo*, y para la pinocitosis simplemente se denominará *Pino*.

Variables de Coordinación y Comportamiento

En los símbolos que denotan las acciones en Brane se pueden notar los subindices n (en el caso de fagocitosis y exocitosis), mediante los cuales se coordina una interacción entre dos membranas. Dicho subíndice será de gran importancia en la interpretación en Utcc ya que corresponde a la variable de coordinación que se usa para la representación de interacciones; recordemos que dicha variable se asocia al sistema mediante la restricción *synchro(x,m)*, donde m correspondería al subíndice n para determinar entre cuáles sistemas se desarrollará la interacción. Por otro lado el símbolo $\perp$ permite diferenciar entre *acción* y *co-acción*, lo que ayudará a determinar el valor de la variable de comportamiento (*true* o *false*); recordemos que la variable de comportamiento se asocia al sistema mediante la restricción *behavior(x,b)*.

Ahora es necesario definir un proceso que asigne los valores correspondientes a las variables de coordinación y comportamiento según corresponda. Dicho proceso se define a continuación.

Definition 7 *Action(Id,n,m) Asignación de valores a las variables de coordinación y comportamiento. Este proceso toma las variables de coordinación y comportamiento de la membrana x identificada por Id y les asigna los valores n y m respectivamente.*

$$Action(Id,n,m) \stackrel{def}{=} (absx, b, a; idmem(x, Id) \wedge\ behavior(x,b) \wedge\ synchro(x,a))$$
$$(tell\ b = m \parallel\ tell\ a = n)$$

Más adelante se define la interpretación de las interacciones fagocitosis y exocitosis. Dichas interpretaciones utilizan el proceso *Action(Id,n,m)* para la asignación de valores a las variables de comportamiento y de coordinación. El proceso recibe

como parámetros: un identificador, el valor de la variable de coordinación y un parámetro booleano que será el valor de la variable de comportamiento. Primero se abstrae la membrana correspondiente a Id, junto con sus variables de comportamiento y sincronización, mediante el proceso *(abs x,b,a ; idmem(x,Id) ∧ behavior(x,b) ∧ synchro(x,a))*; luego de obtener las variables correspondientes se procede a asignar los valores que entran como parámetros a las variables abstraídas mediante el proceso *tell* $(b = m) \parallel$ *tell* $(a = n)$.

4.4.1. Comportamiento Natural de una Membrana: CASE

En las definiciones anteriores se hacían llamados al proceso *CASE*; ahora vamos a dar una definición general de dicho proceso. Como ya fue mencionado en otra oportunidad, el *CASE* hace parte de la interpretación general de membranas *(4)*, y la función principal de este proceso es definir el comportamiento natural de una membrana, es decir que una membrana pueda desempear el rol de la membrana que ingresa, la membrana que recibe, la membrana que expulsa, la membrana que es expulsada y la membrana que realiza pinocitosis. La definición de *CASE* se presentará a lo largo de las interpretaciones que se hace de cada una de las interacciones básicas, ya que es necesario definir el comportamiento que adopta en caso de fagocitosis, exocitosis y pinocitosis.

wait phago recibir	espera phago	
...	ejecuta recibir en caso phago	⇒ CASE_phago
wait phago entrar	espera phago	
...	ejecuta entrar en caso de phago	⇒ CASE_phago
wait exo expulsar	espera exo	
...	ejecuta expulsar en caso de exo	⇒ CASE_exo
wait exo salir	espera exo	
...	ejecuta salir en caso de exo	⇒ CASE_exo
wait pino	espera pino	
...	ejecuta en caso de pino	⇒ CASE_pino

Cuadro 4.1: Definición General CASE

El Cuadro 4.1 permite tener una comprensión generalizada del proceso *CASE*, el cual se encuentra conformado por procesos *wait* que se encuentran a la espera de algún estímulo. Es decir, el proceso *CASE* se conforma de la ejecución paralela de cinco procesos *wait*, donde dos representan la activación para la acción y coacción en caso de *fagocitosis*; otros dos a la espera de la acción y coacción en caso de *exocitosis*; y finalmente un proceso en espera para la activación en caso de *pinocitosis*. Por tal motivo su definición se presentará en tres partes: *CASE_Phago, CASE_Exo* y *CASE_Pino*.

4.4.2. Interpretación del Proceso de Fagocitosis

Como ya fue mencionado, el proceso de Fagocitosis se encuentra coordinado por dos partes, es decir existe la *acción* para la fagocitosis y también existe su correspondiente *co-acción*. Para tales efectos es necesario modelar las dos partes. Para proceder, primero se determina que la membrana cuya variable de comportamiento tenga el valor de *true* será la membrana que permite el ingreso y por el contrario cuando tenga el valor *false* será la membrana que desea ingresar.

4.4.2.1. Acción Phago: Membrana que Ingresa

La sintaxis del cálculo Brane define la acción de la fagocitosis como: $\circledcirc_n \,.\tau\ (\!|P|\!)$, para lo cual se tiene la siguiente interpretación según la definición de la función [[.]] que podemos sumar a las ya definidas en *(1), (2), (3)* y *(4)*.

5. Acción Phago

$[\![\ \circledcirc_n \,.\tau\ (\!|P|\!), S, Id_\tau]\!] = Action(Id_\tau, n, false) \parallel [\![\ \tau\ (\!|P|\!), S, Id_\tau]\!]$

En *Acción Phago (5)* se tiene la acción $\circledcirc_n$ que recae sobre la membrana τ , indicando que está dispuesta a interactuar con la membrana sobre la que recae la co-acción con subíndice n ($\circledcirc_n^{\perp}\ (\rho)$). En esta ocasión el valor para la variable de comportamiento de τ es *false* y para la variable de coordinación es n. Dichos valores son asignados mediante el llamado al proceso $Action(Id_\tau, n, false)$ dejando preparada a la membrana τ en el momento de recibir algún estímulo del medio.

Finalmente se ejecuta en paralelo la *representación general de membrana (4)* para τ.

4.4.2.2. Co-Acción Phago: Membrana que Recibe

La sintaxis del cálculo Brane define la *co-acción* de la fagocitosis como: $\Theta_n^{\perp}(\rho).\sigma(\!|P|\!)$. La interpretación en Utcc según la definición de la función [[.]], se tiene:

6. Co-Acción Phago

$$[[\ \Theta_n^{\perp}(\rho).\sigma(\!|P|\!), S, Id_\sigma]] = \begin{array}{l} Action(Id_\sigma, n, true) \parallel tell\ phago(n,\rho,true) \\ \parallel [[\ \sigma(\!|P|\!), S, Id_\sigma]] \end{array}$$

En *(6)* el símbolo ρ representa el nombre de la nueva membrana que se crea y cuyo contenido será el sistema que desea ingresar, como ya se explicó en el proceso de fagocitosis. La variable ρ representará una anidación en σ y será la membrana que contiene el sistema que desea ingresar. Si tomamos como ejemplo la acción definida en *(5)*, la membrana en el interior de ρ sería la variable τ.

La interpretación hace llamado de $Action(Id_\sigma, n, true)$, con los parámetros n y *true*, en ésta acción *true* que la membrana debe adoptar el comportamiento de una membrana receptora en el proceso de fagocitosis. En este caso también se hace llamado a la *representación general de membrana (4)*, pero a diferencia de *(6)* se agrega información al *store*: *tell phago(n,ρ,true)*. La restricción *phago(n,ρ,true)* representa el estímulo del medio para desencadenar la interacción entre dos membranas para llevar a cabo el proceso de fagocitosis. Nótese que el estímulo utiliza las variables n, ρ y *true* para dar inicio a la interacción entre dos membranas cuya variable coordinación es n, mientras que *true* activa la membrana que recibe, quien a su vez activa a la membrana que ingresa como se muestra más adelante en la definición del *CASE*.

4.4.2.3. CASE_phago: Fagocitosis

Ya tenemos definidas las condiciones necesarias para llevar a cabo una fagocitosis mediante las interpretaciones *(5)* y *(6)*; ahora es necesario definir el comportamiento que adopta una membrana cuando se presentan las condiciones necesarias para interactuar en una fagocitosis. Para lograrlo es necesario definir la parte del *CASE* correspondiente a la fagocitosis donde se establece el comportamiento de la *membrana que recibe* y el de la *membrana que ingresa*.

El *CASE* para el proceso de fagocitosis se define como:

7. CASE_phago

$$
\begin{aligned}
CASE\_phago_{\sigma} \;=\; & (wait\ \rho;phago(A_{\sigma},\rho,B_{\sigma})) \\
& \quad ([[\ \rho\ (\!|\!|),S_{\sigma},x]] \parallel new\_id(x) \\
& \quad \parallel (abs\ \rho',x;channel(\rho',\sigma) \wedge idmem(\rho',x)) \\
& \quad\quad (tell\ receive(\sigma,\rho',A_{\sigma},false))) \\
& \parallel (wait\ \sigma',\rho';\ receive(\sigma',\rho',A_{\sigma},B_{\sigma})) \\
& \quad (tell\ in(\sigma,\rho') \parallel tell\ in(\rho',\sigma') \parallel tell\ finish(A_{\sigma}))
\end{aligned}
$$

La definición de *CASE_phago* en (7) se encuentra seccionada en dos partes como se puede ver: $wait(\rho;phago(A_{\sigma},\rho,B_{\sigma}))$ y $(wait\ \sigma',\rho';\ receive(\sigma',\rho',A_{\sigma},B_{\sigma}))$.

La primera hace que se active la membrana que recibe la cual actúa de la siguiente manera: se genera un identificador nuevo ($new\_id(x)$) el cual es utilizado para la identificación de la nueva membrana ($[[\ \rho\ (\!|\!|),S_{\sigma},x]]$), dicha membrana se crea para recibir al sistema entrante. Como segundo paso el proceso $(abs\ \rho',x;channel(\rho',\sigma) \wedge idmem(\rho',x))$ permite abstraer la variable ρ para poderla comunicar al sistema que ingresa y así éste pueda agregarlo a su conjunto asociado, con el fin de representar el nuevo alcance que adquiere el sistema entrante al anidarse dentro de ρ. Finalmente la variable ρ es comunicada por medio de la restricción $receive(\sigma,\rho',A_{\sigma},false)$, cuyos parámetros A_{σ} y *false* activan la membrana correcta, la membrana que desea ingresar.

La segunda parte corresponde a $(wait\ \sigma',\rho';\ receive(\sigma',\rho',A_{\sigma},B_{\sigma}))$ que al poder ser deducido del *store*, permite que se active la membrana adicionando información al

store de sus nuevos alcances al ingresar: *in(σ,ρ')* y *in(ρ',σ')*; donde ρ'corresponde a la variable ρ que es compartida por la membrana que permite el ingreso. Finalmente se adiciona la restricción *finish*(A_σ) al *store*, con el fin de poder tener control sobre la interacción que se está llevando a cabo. El uso de la restricción *finish*(A_σ) se explica más adelante en el modelado de secuencia de acciones.

Ejemplo *proceso fagocitosis.*

Consideremos la situación de fagocitosis entre τ y σ, expresada en Brane como: $\circledS_n \ . \ \tau \ (\!|Q|\!) \circ \circledS_n^{\perp} \ (\rho) \ . \ \sigma \ (\!|P|\!)$. Sobre la membrana τ recae la acción phago y sobre σ la co-acción phago, es decir, τ es la membrana que ingresa y σ es la que recibe. Luego la interpretación en Utcc de esta situación en base a las definiciones anteriores es:

$$\frac{[[\circledS_n \ . \ \tau \ (\!|Q|\!), S_U, Id_\tau]] \parallel [[\circledS_n^{\perp} \ (\rho) \ . \ \sigma \ (\!|P|\!) \ , S_U, Id_\sigma]]}{\text{composición paralela de sistemas (3)}}$$

Por parte de la membrana τ se tendría, acción phago (5):

$$\text{Action}(Id_\tau,\text{n},\text{false}) \| [[\tau \ (\!|Q|\!), S_U, Id_\tau]]$$

por parte de la membrana σ se tendría, co-acción (6):

$$\text{Action}(Id_\sigma,\text{n},\text{true}) \| \text{tell}(\text{phago}(\text{n},\rho,\text{true})) \| [[\sigma \ (\!|P|\!) \ , S_U, Id_\sigma]]$$

Inicialmente el proceso *Action(id,m,b)* asigna valores a las variables de coordinación y comportamiento para τ y σ. La interacción se coordina entre ambas, debido a que el valor de la variable de coordinación es el mismo para cada una (A_τ=A_σ=n). Luego la variables de comportamiento $B_\tau = false$ y $B_\sigma = true$ indican que τ es la membrana que ingresa y σ la membrana que recibe. Luego, al estar asignados los valores a las variables de coordinacíón y comportamiento, se activa el proceso CASE_phago para cada una de las membranas, desarrollando el comportamiento de la membrana que ingresa y de la membrana que recibe. Centrándonos en las restricciones $in(x, y)$ y los estímulos para las acciones, se tiene en el *store* luego de responder al estímulo $phago(n, \rho, true)$ en la

membrana que recibe: $\exists_\sigma(in(\rho,\sigma) \wedge receive(\sigma,\rho,n,false))$. Luego por el estímulo $receive(\sigma,\rho,n,false)$ por medio del cual se comunican las variable σ y ρ a τ, se activa la membrana que ingresa τ la cual modifica sus alcances, adicionando al *store* las restricciones $in(\tau,\rho)$ y $in(\rho,\sigma)$ (además de $finish(n)$), quedando el *store* como: $\exists_\tau(in(\tau,\rho) \wedge in(\rho,\sigma)) \wedge \exists_\sigma(n(\rho,\sigma) \wedge receive(\sigma,\rho,n,false))$. Lo cual indica ahora que τ está dentro de ρ y éste a su vez en σ (se llevó a cabo el proceso de fagocitosis, donde τ se encuentra dentro de σ contenido en la vesícula ρ). Cabe notar que las variables σ y ρ en $\exists_\tau(in(\tau,\rho) \wedge in(\rho,\sigma))$ son las mismas de $\exists_\sigma(in(\rho,\sigma) \wedge receive(\sigma,\rho,n,false))$ debido a que fueron comunicadas por el proceso CASE_phago de la membrana que recibe.

4.4.3. Interpretación del Proceso de Exocitosis

El proceso de exocitosis al igual que el de fagocitosis también es coordinado por dos partes; la membrana sobre la que recae la *acción* se determina como la membrana que expulsa y la membrana sobre la que recae la *co-acción* es la membrana que es expulsada. En este proceso coordinado por dos sistemas, es necesario establecer el papel que adopta cada uno de los sistemas, lo cual se logra mediante la asignación de valores a la variable de coordinación y de comportamiento. En ésta ocasión el sistema cuya variable de comportamiento tenga como valor *false* corresponderá a la membrana que realiza la expulsión mientras que el otro sistema tendrá su variable de comportamiento asociada con el valor *true*, lo que indica que es la membrana que se expulsa.

4.4.3.1. Acción Exo: Membrana Expulsada

En sintaxis del cálculo Brane se define la acción Exo como: $\circlearrowleft_n .\sigma\ (\!|P|\!)$. A continuación se presenta la interpretación de la acción Exo en Utcc:

8. Acción Exo

$$[[\ \circlearrowleft_n .\sigma\ (\!|P|\!),S,Id_\sigma]] = Action(Id_\sigma,n,true) \parallel [[\ \sigma\ (\!|P|\!),S,Id_\sigma]]$$

En *(8)* se tiene la *acción* $\circlearrowleft_n$ que recae sobre la membrana σ, indicando que está dispuesta a interactuar con la membrana sobre la que recae la *co-acción* con

subíndice n es decir con $\Im_n^{\perp}$. Dicha acción señala a σ como la membrana que sale, por lo que en ésta ocasión el valor para la variable de comportamiento de σ es *true* y para la variable de coordinación es n; dichos valores son asignados mediante el llamado al proceso *Action(Id_σ,n,true)* dejando preparada a la membrana σ en el momento de recibir algún estímulo del medio. Finalmente se ejecuta en paralelo la definición general de membrana para σ.

4.4.3.2. Co-Acción Exo: Membrana que Expulsa

En Brane se define la co-acción de la Exocitosis como: $\Im_n^{\perp}.\tau\ (\!|P|\!)$. La interpretación en Utcc según la definición de [[.]], se tiene:

9. Co-Acción Exo

[[$\Im_n^{\perp}.\tau\ (\!|P|\!)$*,S,$Id_\tau$]] = Actión($Id_\tau$,n,false)* $\|$ *tell exo(n,true)* $\|$ *[[* $\tau\ (\!|P|\!)$*,S,Id_τ]]*

En *(9)* la *co-acción* $\Im_n^{\perp}$ indica a la membrana τ que alguna membrana sobre la que recae la acción $\Im_n$ está dispuesta a salir de ella. En este caso el valor de la variable de comportamiento de τ es *false* según lo establecido anteriormente. También se asigna el valor n a la variable de coordinación para sincronizar la interacción y finalmente se agrega la restricción *exo(n,true)* al *store*, lo cual simboliza el estímulo del medio necesario para desencadenar el proceso de exocitosis. los valores de las variables de comportamiento y de coordinación se asignan mediante el proceso *Actión(Id_τ,n,false)* ya explicado en la definición de *(8)*.

4.4.3.3. CASE_exo: Exocitosis

Anteriormente se definieron *acción* y *co-acción* de la exocitosis, ahora se define el comportamiento de la membrana que expulsa y la membrana que es expulsada, dicha definición del comportamiento es la continuación del *CASE* referente al proceso de Exocitosis.

El *CASE* para el proceso de Exocitosis se define como:

10. CASE_exo

$CASE\_exo_{\sigma}$ = *(whenever exo(A_{σ},B_{σ}) do tell leave(A_{σ},σ,false))*
‖ (wait ρ';leave(A_{σ},ρ',B_{σ}))(tell merge(ρ',σ) ‖ tell finish(A_{σ}))

En *(10)*, en caso de poderse deducir la condición de *whenever exo(A_{σ},B_{σ})* se desencadenará el comportamiento de la membrana que sale, la cual impone la restricción *leave(A_{σ},σ,false)* que a su vez en algún momento en la misma unidad de tiempo dispara la ejecución de la membrana que realiza la expulsión.

La membrana que expulsa se activa mediante el proceso (wait ρ';leave(A_{σ},ρ',B_{σ})), para representar por medio del *store* que la membrana ρ'ha sido expulsada de σ y se ha mezclado con ella (*merge(ρ',σ)*). Finalmente, para tener control sobre la secuencia de acciones se agrega la restricción finish(A_{σ}) al *store* para indicar que la interacción coordinada mediante n, que es el valor de A_{σ}, ya ha terminado de ocurrir. Más adelante se explica la importancia de la restricción *finish(n)* en el control de las acciones en secuencia.

4.4.4. Interpretación del Proceso de Pinocitosis

El proceso de Pinocitosis difiere de los dos mencionados anteriormente (Fagocitosis y Exocitosis) debido a que no necesita de la coordinación entre dos sistemas para poderse llevar a cabo. Como ya se explicó en capítulos anteriores, el proceso de pinocitosis consiste en crear un subsistema que inicialmente no contiene ningún sistema en su interior, pero que es capaz de interactuar en algún momento con la membrana que la contiene o con cualquier otra que se encuentra a su mismo nivel.

Retomando la sintaxis de Brane, se tiene $\circledcirc(\rho).\sigma\ (\!|P|\!)$ para indicar que la membrana σ se dispone a crear un subsistema anidado identificado como la membrana ρ, lo que trae como resultado la reducción $\sigma\ (\!|P \circ \rho(\!|\diamond|\!)|\!)$ donde se puede apreciar que la membrana σ ahora tiene en su interior al sistema simbólico P interactuando en paralelo con la membrana ρ.

4.4.4.1. Acción Pino: Creación de una nueva Membrana Interna

Como se acaba de mencionar, la Pinocitosis no requiere de dos sistemas que se coordinen para llevar a cabo dicho proceso, lo que resume que en esta ocasión es suficiente definir la acción pino solamente ya que no se necesita de una co-acción. A continuación se presenta la interpretación en Utcc de la acción pino.

11. Acción Pino

$[[\ \circledcirc(\rho).\sigma\ (\!|P|\!),S,Id_\sigma]] = tell\ pino(\rho,Id_\sigma) \parallel [[\ \sigma\ (\!|P|\!),S,Id_\sigma]]$

La interpretación de la pinocitosis en *(11)* no es muy compleja, solo requiere de establecer el estímulo $pino(\rho,Id_\sigma)$ en el *store* y en paralelo se ejecuta la interpretación general de membrana, la cual con apoyo en la definición del *CASE* respecto a la pinocitosis es capaz de responder al estímulo.

4.4.4.2. CASE_pino: Pinocitosis

Ya definida la acción *pino*, se procede ahora a completar la definición del proceso *CASE* donde se modela el comportamiento de una membrana al responder al estímulo de la acción pionocitosis.

12. CASE_pino

$CASE\_pino_\sigma = (wait\ \rho;pino(\rho,Id_\sigma))([[\rho\ (\!|\diamond|\!),S_\sigma,Id_\rho]] \parallel tell\ finish(Id_\sigma))$

El modelo del comportamiento en caso de una pinocitosis es sencillo como se puede ver en *(12)*. Primero se espera el estímulo ($pino(\rho,Id_\sigma)$) el cual induce el proceso de pinocitosis, en caso de poder deducir dicho estímulo del store, se procede a crear la nueva membrana por medio de la definición general de membranas establecida en *(4)*. El llamado de $[[\rho\ (\!|\diamond|\!),S_\sigma,Id_\rho]]$ con S_σ como parámetro, establece que σ es la membrana que contiene a ρ y por lo tanto ρ se encuentra contenida en σ. Finalmente para mantener control sobre la ejecución de acciones en secuencia se ejecuta el proceso $tell\ finish(Id_\sigma)$ para indicar que la pinocitosis ha terminado de ejecutarse.

En las traducciones desde *(5)* hasta *(12)* se logró dar interpretación a las interacciones básicas básicas: *Fagocitosis, Exocitosi y Pinocitosis*. Además se modeló el comportamiento natural de una membrana mediante la definición del proceso *CASE*, que establece cómo se comporta una membrana a la hora de interactuar en alguna de las interacciones básicas. Las definiciones anteriores serán la base para dar interpretación a las interacciones *Mate, Mito y Budding*, las cuales son resultado de la combinación en secuencia de las interacciones básicas. Más adelante se presenta la interpretación de dichas interacciones derivadas de las básicas, pero antes es necesario tener en cuenta dos aspectos: el manejo de la información en el tiempo y el control sobre las acciones que se ejecutan en secuencia. Cuando se habla del manejo de la información en el tiempo, se hace referencia a la posibilidad de considerar diferentes unidades de tiempo en Utcc. Recuerde que al cambiar de unidad de tiempo la información en el store no se transmite de la unidad anterior a la siguiente unidad, por lo que es necesario establecer un proceso que permita modelar la propiedad de persistencia de la información a lo largo del tiempo. Por otro lado, el control de secuencia de acciones, hace referencia a algún mecanismo para mantener control sobre la posibilidad que considera el cálculo Brane de aplicar más de una acción sobre una sola membrana, ya que por ejemplo, una membrana no puede ingresar y ser expulsada al mismo tiempo, primero tiene que realizar una de las dos interacciones y luego de estar seguros de que ha finalizado la primera interacción se procede a iniciar la segunda interacción. Dicha problemática es abordada en la siguiente sección.

4.5. Persistencia de la Información a lo largo del Tiempo

Recordemos que la información en el store se pierde al cambiar de unidad de tiempo, por tal motivo es importante considerar la información que se debe conservar en las futuras unidades de tiempo. Esto es necesario ya que el store contiene información requerida para el funcionamiento de cualquier membrana. Cabe recordar que existen variables, conjuntos e identificadores asociados a las membranas, información que se necesita para el correcto funcionamiento de la membrana.

A continuación se define el proceso Persistencia, que hace posible el traspaso de la información de una unidad de tiempo a otra.

Definition 8 *Propiedad de Persistencia. Se debe asegurar la ejecución permanente del siguiente proceso para conservar la información necesaria en todas las unidades de tiempo.*

$$
\begin{aligned}
\textit{Persistencia} = {} & \textit{(abs x,s;idset(x,S))(next tell idset(x,S))} \\
& \parallel \textit{(abs x,b;behavior(x,b))(next tell behavior(x,b))} \\
& \parallel \textit{(abs x,b;synchro(x,b))(next tell synchro(x,b))} \\
& \parallel \textit{(abs x,Id;idmem(x,Id))(next tell idmem(x,Id))} \\
& \parallel \textit{(abs x,y;merge(x,y))(next tell merge(x,y))} \\
& \parallel \textit{(abs x,y;in(x,y))(unless merge(x,y) next tell in(x,y))}
\end{aligned}
$$

Según la *Definicion 8*, la Propiedad de Persistencia establece que cada variable e identificador asociados a un sistema debe perdurar en el tiempo, es decir que cualquier restricción de tipo *idset(x,S)*, *behavior(x,b)*, *action(x,b)*, *idmem(x,I)* y *merge(x,y)*, que pueda ser abstraída del store en la unidad de tiempo actual, debe también estar presente en la siguiente unidad de tiempo. Para las restricciones de tipo *in(x,y)* deben tenerse en cuenta con especial atención ya que en la presente unidad de tiempo pudo haber ocurrido una exocitosis, lo que implica que un sistema que se encontraba dentro de otro, ha salido. Por lo tanto no sería lógico tener una restricción de tipo *in(x,y)* que indica que el sistema todavía se encuentra anidado. Para el caso de la restricción *in(x,y)*, debemos considerar el proceso de exocitosis de acuerdo a como se define en Brane, donde la membrana expulsada se mezcla con la membrana que lo expulsa, aspecto que es reflejado en *(10)* mediante la restricción *merge(x,y)*. Por lo tanto podemos afirmar que si se puede deducir del *store* la información *merge(x,y)*, lo cual indica que y ha salido de x, sería contradictorio continuar en la siguiente unidad de tiempo con la información *in(x,y)*. En conclusión, si es posible deducir *in(x,y)* esta información debe pasar a la siguiente unidad de tiempo, a menos de que exista la condición *merge(x,y)*.

4.6. Secuencia de Acciones: Interacciones Derivadas

Las interacciones derivadas *Mate, Budding* y *Dripping*, son el resultado de la ejecución en secuencia de algunas de las interacciones básicas en un orden determinado, como ya se comentó en el capítulo anterior. Por ejemplo, en el caso de *Mate*, es posible representar dicho proceso ejecutando en secuencia primero una fagocitosis, luego una exocitosis y finalmente otra exocitosis. Pero antes de entrar a interpretar este tipo de interacciones es necesario establecer algún tipo de control sobre acciones que se ejecutan en secuencia para evitar cualquier comportamiento inadecuado durante las interacciones.

Para tal efecto debemos establecer cómo se debe interpretar las secuencias de acciones que se manifiestan en Brane. Más de una única acción que recaen sobre una membrana se expresa de la siguiente manera en sintaxis de Brane: $\circledcirc_n.\circledcirc_{\overline{n}}^{\perp}.\sigma\ (\!|P|\!)$. Para este ejemplo en particular se puede ver que sobre la membrana σ recae primero una acción *phago* coordinada por el subíndice *n* y luego una acción *exo* coordinada por el subíndice *n'*, leyendo de izquierda a derecha, lo que indica que la membrana σ primero debe ingresar en alguna otra membrana sobre la que recae la *co-acción fago*, y luego debe ser expulsada según la accióne *exo* que recae sobre la misma.

La interpretación general en Utcc de cualquier secuencia de acciones sobre una membrana se establece como:

13. Secuencia de Acciones

$$
\begin{aligned}
[\![\ A_{n_1}.A_{n_2}.\ \ldots\ .A_{n_n}.\sigma\ (\!|P|\!),S,Id_\sigma]\!] = & Accion_{A_{n_1}} \\
& \parallel (\textit{whenever finish}(n_1)\ \textit{do next}\ Accion_{A_{n_2}}) \\
& \parallel (\textit{whenever finish}(n_2)\ \textit{do next}\ Accion_{A_{n_3}}) \\
& \parallel \ldots \\
& \parallel (\textit{whenever finish}(n_{n-1})\ \textit{do next}\ Accion_{A_{nn}}) \\
& \parallel [\![\sigma\ (\!|P|\!),S,Id_\sigma]\!]
\end{aligned}
$$

En *(13)* la simbolización de A_{n_1}, A_{n_2} y A_{n_n} , hace alusión a los operadores que representan acciones en la sintaxis del cálculo Brane.

Esta interpretación de secuencias de acciones establece que primero se realiza la traducción correspondiente a la primera acción de izquierda a derecha en la secuencia de acciones, en paralelo se ejecutan proceso para preguntar si es posible deducir a lo largo de cada unidad de tiempo la restricción *finish(n)*, indicando que la interacción ha terminado para dar paso a la ejecución de la siguiente acción en la siguiente unidad de tiempo mediante el operador *next*, y así sucesivamente hasta completar la traducción de cada acción en diferentes unidades de tiempo, todas en paralelo con la interpretación general de membrana *(4)*, que es quien reacciona de acuerdo a la ejecución de las acciones. Es necesario aclarar que la traducción de cada acción omite el llamado a *representacion general de membrana (4)* ya que en el caso de traducir la secuencia de *n* acciones, se haría llamado repetitivo de *(4) n* número de veces, por tal motivo solo se hace llamado de *(4)* una sola vez ya que se necesita solamente una instancia en ejecución de cada membrana. Para mejor comprensión se presenta la interpretación del siguiente caso:

***Ejemplo**, secuencia de acciones.*

Considere los procesos $P \stackrel{def}{=}$ *Action*(Id_σ,n,*false*) *y* $Q \stackrel{def}{=}$ *Action*(Id_σ,n,*true*) *en la siguiente interpretación:*

$$[[\circledS_n.\ \circledS_n^{\perp}.\sigma\ \ (\!|P|\!),S,Id_\sigma]] = P \parallel \textit{(whenever finish(n) do } Q) \parallel [[\sigma\ (\!|P|\!),S,Id_\sigma]]$$

Se puede notar que la primera acción que recae sobre σ *es la* acción phago *por lo que se requiere de* (5) *para traducir dicha acción y señalar a la membrana* σ *como la membrana que desea ingresar mediante, la ejecución del proceso* P*. Luego de coordinarse la fagocitosis bajo el índice* n*, la membrana establece la restricción* finish(n) *(por medio de* (7)*) en el store para dar paso en la siguiente unidad de tiempo a la ejecución de la* acción exo *(por medio de* (8)*), representada por* Q*, en el momento de deducir la guarda* finish(n)*.*

Teniendo ya definido *Secuencia de Acciones (13)* podemos ahora retomar el caso de las interacciones derivadas, debido a que *(13)* permite tener control al ejecutar

en secuencia varias interacciones básicas de la siguiente manera:

Mate, Budding y Dripping

La interacción Mate resulta de la coordinación entre dos membranas que desean unirse en una única membrana con el contenido de ambas. Se necesita una acción y co-acción las cuales se definen según el cálculo Brane como:

mate: $mate_n\sigma = \mathfrak{Y}_n.\ \mathfrak{V}^{\perp}_{n'}.\ \sigma$

co-mate: $mate^{\perp}_n\tau = \mathfrak{Y}^{\perp}_n\ (\mathfrak{V}^{\perp}_{n'}\ .\ \mathfrak{V}_{n''})\ .\ \mathfrak{V}^{\perp}_{n''}\ .\ \tau$

La interpretación en Utcc de esta interacción al igual que en Brane es el resultado de ejecutar en secuencia los procesos fago, exo y exo (en este respectivo orden).

Interpretación Acción Mate en Utcc:

$$[[\mathfrak{Y}_n.\ \mathfrak{V}^{\perp}_{n'}.\ \sigma\ (\!|P|\!),S,Id_\sigma]] = Action(Id_\sigma,n,false) \parallel \textit{whenever finish(n) do next } Action(Id_\sigma,n',true) \parallel [[\sigma(\!|P|\!),S,Id_\sigma]]$$

Interpretación Coacción Mate en Utcc:

$$[[\mathfrak{Y}^{\perp}_n\ (\mathfrak{V}^{\perp}_{n'}\ .\ \mathfrak{V}_{n''}).\ \mathfrak{V}^{\perp}_{n''}\ .\ \tau\ (\!|Q|\!),S,Id_\tau]] = Action(Id_\tau,n,true) \parallel \textit{tell phago(n, } \mathfrak{V}^{\perp}_{n'}.\ \mathfrak{V}_{n''},true) \parallel \textit{whenever finish(n) do next } Action(Id_\tau,n'',false) \parallel \textit{tell exo(n'',true)} \parallel [[\tau\ (\!|Q|\!),S,Id_\tau]]$$

La interacción Budding, al igual que Mate resulta de la coordinación entre dos sistemas. Es decir, se necesita de un acción y su respectiva co-acción. Esta interacción se deriva de la ejecución de los procesos pino, fago y exo. Las respectivas definiciones de acción y coacción para Budding se definen en Brane como:

bud: $bud_n\sigma = \mathfrak{Y}_n.\ \sigma$

co-bud: $bud^{\perp}_n\rho = \ \circledcirc\ (\mathfrak{Y}^{\perp}_n\ (\rho)\ .\ \mathfrak{V}_{n'}).\ \mathfrak{V}^{\perp}_{n'}\ .\ \tau$

La interacción Dripping difiere de las dos anteriores, ya que no requiere de una acción y coacción por el hecho de que no es coordinada por dos sistemas. Se deriva de la ejecución de los procesos pino, pino y exo. en el mismo orden respectivo.

drip: $$bud_n\sigma = \circledcirc(\circledcirc (\rho). \circlearrowleft_n) \ . \circlearrowleft_n^{\perp} .\sigma$$

Las interacciones Budding y Dripping al ser una derivación de las interacciones básicas, se puede interpretar en Utcc de manera similar a como se hizo con la acción y co-acción de Mate.

4.7. Discusiones Sobre la Interpretación en Utcc

En esta Sección se presentan algunos aspectos desarrollados a lo largo del capítulo y que pueden ser considerados en discusión.

La interpretación propuesta de los sistemas conformados por membranas utilizando CCP, como ya se mencionó en la Sección 4.1 del presente capítulo, se fundamenta en el alcance de los sistemas, es decir, en la localidad de un sistema para reconocer cuales son los sistemas que lo contienen y cúales se encuentran a su mismo nivel, también para reconocer las nuevas configuraciones después de cualquier interacción. Se definió que cada sistema o membrana es representada por variables, donde cada una se toma como una variable local a una serie de procesos que se ejecutan para poder simular el comportamiento de una membrana. Es en este aspecto donde toma importancia la movilidad como característica de Utcc, ya que facilita el modelado de las interacciones permitiendo la comunicación entre las membranas, que son representadas como variables locales. Podemos notar la importancia que tiene la definición del sistema de restricciones para poder modelar las membranas, ya que con base en restricciones como *in(x,y)* es que se establece la posición de una membrana dentro de un sistema complejo y no por el manejo de variables locales. Por lo tanto, es muy probable que el presente estudio pueda ser concebido sin la necesidad de establecer las membranas como variables locales si se puede tener un sistema de restricciones que permita representar aspectos importantes en este tipo de sistemas.

Otro aspecto importante que se puede considerar es el uso del no determinismo, ya que biológicamente puede ocurrir que en determinado momento una membrana se encuentre interactuando con otras dos membranas que requieran realizar un tipo de interacción sobre ellas, es decir, dada la membrana x quien establece una acción inicial de *phago* y dos nuevas membranas y y z, respectivamente, las cuales establecen una *co-acción phago* indicando que desean entrar a la membrana x. Este tipo de situaciones fueron estudiadas por Cardelli en [Car98] a través de las secuencias de acciones que fueron detalladas en secciones anteriores. Para este caso Cardelli plantearía que inicialmente debe realizar *phago* una de las dos membranas para posteriormente realizar el proceso con la otra. Si se llegara a plantear este caso, sintácticamente en Brane se realizaría una de las dos reducciones, pero quedaría pendiente la otra reducción ya que la membrana x ha perdido la acción de *phago* al sincronizar con una de las otras dos membranas. Llevando este caso al modelo anteriormente expuesto, no sería posible lograr una representación para dicho caso en Utcc puesto que es no se cuenta con operadores no determinísticos.

4.8. Resumen

En este capítulo se presentó la descripción detallada de la representación de los sistemas conformados por membranas, junto con el sistema de restricciones que fue necesario para ilustrar diversos aspectos en el modelado de estos sistemas. El sistema de restricciones se presentó en Sección 4.2, con la descripción del significado que tenía cada restricción, además aclarando ciertas condiciones sobre cada una. Después se definió la interpretación para los sistemas propuestos por el cálculo Brane. Para llevar a cabo la interpretación de las membranas y los comportamientos de acuerdo a cada interacción, fue necesario dividir la representación en dos partes: el modelado de cada una de las interacciones básicas *fagocitosis, exocitosis*, y *pinocitosis*, y por otro lado el modelado del comportamiento que adquiere cualquier membrana en el caso de alguna interacción. Para cada una de las interacciones se presentó la interpretación en Utcc para cada *acción* y *co-acción* correspondiente. Seguidamente se definió el proceso *CASE*, que modela la naturaleza de toda membrana, ya que representa el trabajo que realiza una membrana en el momento de recibir el estímulo del ambiente para interactuar.

Luego de tener modeladas cada una de las interacciones básicas y el comportamiento natural de una membrana frente a cada interacción, se propuso una interpretación de las interacciones derivadas, las cuales son el resultado de la ejecución en secuencia de determinadas interacciones básicas. Para lograr modelar este tipo de interacciones conformadas por las básicas, fue necesario establecer algún tipo de control sobre acciones que se ejecutan en secuencia, para lo cual se definió una traducción generalizada en Utcc para cualquier secuencia de acciones en Brane. Teniendo definido el control y con base en las interacciones básicas se puede presentar la interpretación en CCP de las interacciones derivadas. Para finalizar se presentaron una serie de aspectos a discutir en cuanto a la representación que se hace de los sistemas conformados por membranas y a las características propias de Utcc. Especialmente se determinó que el uso de variables locales no era necesario, ya que la localidad como tal de un sistema está determinada por el sistema de restricciones ($in(x,y)$). También se discutió el tema del no determinismo al considerar casos reales en los que en la membrana se tiene más de una interacción, lo cual es asumido en Brane como una ejecución en secuencia, mientras en Utcc no se podría simular el comportamiento establecido en Brane , ya que el comportamiento determinista de Utcc permitiría más de una interacción en una sola membrana.

5 Conclusiones y Trabajo Futuro

En este capítulo se presentarán algunas conclusiones generales como resultado del desarrollo de este trabajo de grado. Primero se exponen algunas relaciones encontradas entre las características de los formalismos basados en CCP y las características propias de los sistemas biológicos. También se presentarán las ventajas que representa el uso de CCP en el modelado de membranas celulares, particularmente el uso de Utcc como formalismo para representar estos sistemas y sus interacciones.

5.1. CCP en el Modelado de Sistemas Biológicos

En este trabajo de grado se encontraron aspectos importantes del uso de la programación concurrente por restricciones en el modelado de sistemas biológicos (a nivel de membrana celular). Encontramos que existe una relación entre características de algunos formalismos en el modelo CCP y los sistemas biológicos que pretendemos modelar. Tales características de esos sistemas son: la ocurrencia de varios sistemas a un mismo nivel jerárquico (concurrencia), las membranas representan sistemas dinámicos y movilidad en los sistemas.

5.1.1. Sistemas Concurrentes

CCP es un formalismo computacional que nos permite representar la interacción entre sistemas de manera *concurrente*. Las características de este paradigma computacional lo han llevado a alcanzar un papel de gran importancia, ya que los modelos descritos logran una representación individual de cada agente que, en

interacción con otros, conforman un sistema. Esta representación de agentes y procesos permiten profundizar en el estudio de este tipo de sistemas, comprendiendo la función de cada componente y cómo interactúan entre sí dando forma y características a los sistemas.

El hecho de poder modelar procesos concurrentes perfila los modelos basados en la programación concurrente como mecanismos viables en el estudio de sistemas biológicos. Dichos sistemas que se presentan en la naturaleza, como en el caso de la membrana celular, se encuentran conformados por subsistemas (otras membranas en el interior de la célula), los cuales interactúan constantemente entre sí para llevar a cabo procesos que son necesarios para la subsistencia de la célula (*concurrencia entre sistemas*), como el transporte de elementos que se encuentran en un compartimento determinado (membrana) o hacia el exterior de la célula.

5.1.2. Noción de Tiempo y la Membrana Celular como Sistema Dinámico

En el Capítulo 2 se presentó la membrana celular como una estructura dinámica, desde el punto de vista de los procesos que ocurren a este nivel. Algunas regiones de la membrana se desplazan en el interior de la célula; esto ocurre cuando la membrana se invagina para cubrir algún elemento. Luego se desprende esta invaginación en forma de vesícula con el elemento en su interior y de esta manera logra transportar materia de un lugar a otro. Este tipo de comportamiento califica los sistemas conformados por membranas como sistemas dinámicos, que presentan cambio en su configuración en el transcurso de tiempo.

Algunos de los formalismos basados en el modelo CCP consideran la noción de tiempo por medio de primitivas propias a los mismos. La noción de tiempo es de gran importancia en el modelado de sistemas reactivos. Esta característica nos permitió modelar sistemas biológicos basados en membranas celulares por medio de la programación concurrente por restricciones, ya que teniendo en cuenta el comportamiento dinámico de estos sistemas, se puede detallar la evolución o cambios en la configuración de la membrana en el tiempo.

5.1.3. Almacén de Restricciones y Movilidad

La noción de restricción como representación de información parcial, ha jugado un papel importante en la aplicación de CCP para el estudio de sistemas biológicos. Partiendo del hecho de que no se tiene conocimiento sobre muchos de los procesos que ocurren en la biología y en algunos se tiene solo un conocimiento parcial sobre su comportamiento.

En CCP todos los procesos logran interactuar comunicándose por medio de un almacén de restricciones llamado *Store*. En este almacén se tiene toda la información generada por cada componente del sistema; la cual puede ser complementada por medio de procesos que adicionan información al almacén y a su vez también puede ser consultada por otros procesos y de esta manera representar su comportamiento de acuerdo a la información existente. Uno de los trabajos realizados [AG06], destaca el uso de la información parcial en el modelado de sistemas biológicos, en el cual se afirma que se puede distinguir dos tipos de información parcial: *cuantitativa y comportamental*. La primera hace referencia al estado de las variables, el cual usualmente se asocia a un estado de incertidumbre. Y la segunda, se relaciona a la incertidumbre del comportamiento de las entidades de un sistema.

En este trabajo de grado, el uso de restricciones desempeñó un papel muy importante, ya que por medio del sistema de restricciones que se propuso fue posible lograr una representación jerárquica de los sistemas y subsistemas (e.g. la restricción in(x,y), estableciendo que la membrana x se encuentra anidada en la membrana y). Teniendo en cuenta que en la membrana celular se encuentran otros tipos de membranas las cuales se encuentran interactuando constantemente a diferentes niveles en la jerarquía de la estructura del sistema (es decir, a distintos niveles de anidación).

La membrana celular es un delimitante, que aísla del exterior una serie de procesos que ocurren en su interior. De esta manera, se puede entender que existen mecanismos de comunicación entre este tipo de sistemas para coordinara las interacciones que pueden existir. Desde este punto de vista, el concepto de movilidad en los formalismos basados en CCP desempeña un papel importante en el estudio

de sistemas biológicos. En nuestro caso, se requirió de un cálculo que extendiera CCP, siendo capaz de expresar movilidad, razón por la cual se hizo uso de Utcc (extensión de CCP que permite modelar comportamiento infinito y movilidad) para modelar las membranas y sus interacciones, ya que fue necesario tener representadas las membranas como variables locales a un conjunto de procesos y a su vez debía ser posible comunicar estas variables a otros sistemas independientes, con el fin de representar las interacciones entre membranas.

5.2. Ventajas y Desventajas en el uso de CCP para el Modelado de Membranas Celulares

A partir del estudio realizado y del modelo propuesto en Utcc para la interacción entre membranas, se detallan a continuación las principales ventajas y desventajas del uso de CCP para el modelad de este tipo de sistemas.

Ventajas

- Una de las principales ventajas de modelar cualquier tipo de sistemas utilizando formalismos basados en CCP, es que cada uno cuenta con un sistema de pruebas que facilita la verificación de propiedades de los sistemas. Esto se debe a la visión declarativa de los procesos, y cómo éstos pueden ser expresados como fórmulas en cierta lógica.

- En la literatura se ha mostrado que CCP es un formalismo adecuado en el modelado de sistemas biológicos, debido a la relación existente entre las características de éstos y CCP. Encontramos que Utcc entonces aplica para el modelado de sistemas biológicos, ya que permiten abstraer cada componente de un sistema y estudiarlo como una única entidad, luego se puede considerar la interacción entre estos componentes y la conformación de subsistemas gracias al comportamiento concurrente permitido por Utcc (característico de los sistemas biológicos).

- En los cálculos bioinspirados (BioAmbients y Brane) se dificulta comprender si se desarrolla más de una acción en una unidad de tiempo. Por lo anterior,

se puede ver que estos cálculos no consideran la noción de tiempo, lo que hace difícil estudiar los cambios de configuración en determinados intervalos de tiempo. Por lo cual, al tener en cuenta los fenómenos que ocurren en la biología se hace difícil su comprensión. En este aspecto, Utcc al estar basado en la noción de tiempo discreto, facilita el estudio de casos en los que ocurre una interacción entre más de dos membranas, ya que sé puede detallar en cada unidad de tiempo cuál ha sido la evolución del sistema y que cambios de configuración se han presentado.

- Utcc extiende CCP para modelar *movilidad*, característica propia de un sistema conformado por membranas, como se ya se mencionó. El uso de este concepto facilitó la definición de las membranas en el cálculo, ya que como se expuso en el modelo, un sistema tiene alcance a las membranas que lo contienen, pero si llega a ocurrir un proceso a nivel biológico que implique la expulsión o la aceptación de un nuevo elemento el sistema deberá perder o ganar un alcance según sea el caso, debido al cambio de configuración que también implica cambios jerárquicos(*alcances*). Dicha representación fue posible gracias a la comunicación de nombres privados entre procesos utilizando el operador (abs x;c) de Utcc, logrando de esta manera modelar este tipo de sistemas.

- Previamente el grupo de investigación AVISPA ha realizado implementación de cálculos como Ntcc. Sería interesante contar con implementación de Utcc, con la finalidad de realizar simulaciones y pruebas bajo ciertas condiciones de los modelos desarrollados. De esta manera se pueden estudiar los resultados y poder compararlos con otro tipo de estudios propuestos.

- - Si bien Utcc a diferencia de Brane, no se diseñó específicamente para modelar membranas celulares, el concepto de restricción nos permitió superar dificultades al no contar con primitivas para representar la configuración de este tipo de sistemas. Las restricciones además de representar información parcial, en éste trabajo de grado fueron utilizadas para abstraer e interpretar la configuración jerárquica que existe en las membranas celulares. Fue posible utilizar la información del almacén de restricciones para representar la estructura de los sistemas representados (como se menciona en la Sección

5.1.3).

Desventajas

- - No existen formalismos basados en CCP que resulten como inspiración en los sistemas de membranas celulares. Es decir, no existe un cálculo como extensión de CCP cuyas primitivas permitan modelar con facilidad este tipo de sistemas. Operadores para representar la noción de membrana, ambiente o espacio. Lo cual implica un grado de dificultad al momento de abstraer este tipo de sistemas utilizando como herramientas los formalismos existentes que se basan en CCP.
- - Recuerde el tema del no determinismo que se discutió en el Capítulo 4, para situaciones en las que se puedan presentar varias interacciones al mismo tiempo sobre una membrana. De acuerdo a lo establecido en Brane, en el caso de tener tres membranas interactuando al mismo tiempo bajo la misma variable de coordinación, solo se lleva a cabo la reducción entre dos membranas quedando así en inanición la tercera membrana. Este caso en Utcc sí desencadenaría la interacción entre las tres membranas, algo que no se esperaría ya que el modelo propuesto tiene como objetivo simular los procesos establecidos en Brane.

5.3. Trabajo Futuro

En esta sección sugerimos algunas direcciones de trabajo futuro. Algunos aspectos que se discutieron en el Capítulo anterior se pueden considerar ahora como objetos de estudio, otros aspectos simplemente plantean diferentes alternativas como el uso de otros cálculos o sistemas de restricciones basados en otra lógica, como se explica a continuación.

1. Para los modelos propuestos sobre la representación de membranas y cada una de las interacciones básicas se hace necesario realizar pruebas de correctitud que permitan confirmar la adecuada representación de sistemas basados en membranas. Además este aspecto representaría el primer paso para realizar verificación de propiedades de este tipo de sistemas.

2. Implementar el modelo propuesto siguiendo un enfoque declarativo utilizando un lenguaje de programación basado en restricciones como Mozart-OZ, o alguna librería genérica como GCode. Dicha implementación es de gran importancia, ya que facilitaría el estudio de estos sistemas y permitiría realizar simulaciones del modelo bajo ciertas condiciones.

3. Utilizar la lógica espacial como alternativa para el desarrollo del sistema de restricciones. Este tipo de lógica ha servido como fundamentación de ciertos cálculos inspirados en la representación de membranas biológicas.

4. Evaluar otras alternativas diferentes a Utcc que incluyan elementos o herramientas probabilísticas en conjunto con operadores para representar no determinismo. Este aspecto permite aproximar el modelo propuesto a la realidad de este tipo de sistemas. Una de estas extensiones para el modelo CCP es Probabilistic Concurrent Constraint Programmming que permite introducir a los sistemas la noción de probabilidad para dar un acercamiento más real a los modelos propuestos.

5. La implementación de un simulador de Utcc y del sistema de restricciones propuesto. Este simulador ofrecería una herramienta útil para evaluar el comportamiento del modelo propuesto u otros modelos con el mismo enfoque.

Bibliografía

[AA99] M. Abadi and A.Gordon. A calculus for cryptographic protcols: The spi calculus. In *Information Computing*, 1999.

[AG06] Alejandro Arbeláez and Julian Gutiérrez. Estudio exploratorio de la aplicación de la programación concurrente por restricciones en bioinformática. 2006.

[AL06] A. A. Aristizabal and H. A. López. Using process calculi to model and verify security properties in real communication protocols. 2006.

[ALRV05] A. Aristizabal, H. A. Lopez, C. Rueda, and F. D. Valencia. Process calculi for the verification of security properties of communication protocols for peer-to-peer systems. 2005.

[AREPWSS04] L. Cardelli A. Regev E. Panina W. Silverman and E. Shapiro. Bioambients: An abstraction for biological compartments. In *Theoretical Computer Science*, 2004.

[Car98] L. Cardelli. Brane calculi. In *volume 3082 Lecture Notes in Computer Science*, 1998.

[Car08] Luca Cardelli. Bitonal membrane systems. 2008.

[CG98] L. Cardelli and A. Gordon. Mobile ambients. In *volume 1378 of Lecture Notes in Computer Science*, 1998.

[COV07] C. Palamidessi C. Olarte and F. Valencia. Universal timed concurrent constraint programming. In *Proc. of ICLP*, 2007.

[dJPJC06] Mario de J. Pérez Jiménez and Fernando Sancho Caparrini. La biología como alternativa computacional. 2006.

[EMR06] C. Olart E. Monfroy and C. Rueda. Exploring process calculi as a mechanism to define dynamic enumeration strategies in constraint programming. 2006.

[EP03] Oscar Rojas Espinosa and Adriana Arce Paredes. Fagocitosis: Mecanismos y consecuencias. 2003.

[Ghe00] Paun Gheorghe. Computing with membranes. In *Journal of Computer and System Science, 6, 1 - www.tucs.fi*, 2000.

[Ghe04] Paun Gheorghe. Introduction to membrane computing. 2004.

[JGR05] Jorge Andrés Pérez Julian Gutiérrez and Camilo Rueda. Modelamiento de sistemas biológicos usando cálculos de procesos concurrentes. 2005.

[Mil89] R. Milner. Comunication and concurrency. international series in computer science. 1989.

[Mil99] R. Milner. Communicating and mobile systems: Pi calculus. 1999.

[MNV02a] C. Palamidessi M. Nielsen and F. Valencia. Temporal concurrent constraint programming : Denotation, logix and applications. In *Nordic Journal of Computing*, 2002.

[MNV02b] C. Palamidessi M. Nielsen and F. D. Valencia. Temporal concurrent constraint programming: Denotation, logic and applications. 2002.

[Ola09] C. Olarte. Universal temporal concurrent constraint programing. 2009.

[RV05] C. Rueda and F. Valencia. A temporal ccp calculus as an audio processing. In *In SMC'05*, 2005.

[Sar93] Vijay A. Saraswat. In *Concurrent Constraint Pogramming*, 1993.

[Sch06] C. Schulte. Gecode. generic constraint development environment. 2006.

[SJG94] Vijay Saraswat, Radha Jagadeesan, and Vineet Gupta. Foundations of timed concurrent constraint programming. In *Ninth Annual IEEE Symposium on Logic in Computer Science*, 1994.

[Smo95] G. Smolka. The oz programming model. In *Computer Science Tdavy, colume 1000 of LNCS*, 1995.

[Tor09] M. Toro. Probabilistic extension to the concurrent constraint factor oracle model for music improvisation. 2009.