Propuesta de Trabajo de Investigación

John Alexander Vargas
Estudiante de Maestría en Ingeniería
Universidad del Valle

Septiembre 2009

1. Resumen

La programación concurrente por restricciones son un formalismo para modelar computacionalmente sistemas de agentes que interactúan entre ellos a través de un almacén de restricciones. *ntcc* es una derivación de *ccp* para modelar sistemas reactivos adicionando el concepto de tiempo y no determinismo. La movilidad es una característica fundamental de algunos sistemas reactivos y *ntcc* y *ccp* han mostrado que modelar la movilidad en ellos no es una tarea que se pueda realizar de forma natural. El cálculo de ambientes modela sistemas móviles a través de jerarquías y configuraciones espaciales de procesos. Las lógicas espaciales son un formalismo para definir las propiedades de sistemas móviles como los modelados en π-cálculo y en ambientes. En este trabajo de investigación se pretende explorar el uso del cálculo *ccp* o una extensión de él, caracterizándolo con un sistemas de restricciones basado en la lógica espacial del cálculo de ambientes para modelar procesos con propiedades de sistemas móviles, propiedades definidas bajo la misma lógica.

2. Problema

La semántica de los cálculos de procesos *ccp*, *tcc* y *ntcc* no permiten modelar propiedades de movilidad de sistemas reactivos de manera natural pues las variables son canales y la unificación se utiliza para "ligar" le mensaje al canal. Esto es porque las variables lógicas solo se pueden asignar una vez y entonces, no se puede enviar dos mensajes distintos por el mismo canal. De lo contrario, el almacen de restricciones se vuelve inconsistente. *utcc* es un cálculo que adiciona movilidad al estilo del π-cálculo, pero no se ha verificado si sirva como modelo para las propiedades que cumplen sistemas móviles desde el punto de vista de localización jerárquica.

3. Objetivos

Objetivo General

Usar CCP para razonar sobre el comportamiento y estructura de sistemas móviles como los modelados en el cálculo de Ambientes.

Objetivos Específicos

1. Verificar si *ccp* satisface la lógica espacial o alguna parte de ella.
2. Verificar si *utcc* satisface propiedades espaciales como las satisface el π-cálculo
3. Definir un cálculo concurrente por restricciones parametrizado con un sistema de restricciones espaciales.
4. Explorar si usando un sistema de restricciones basado en la lógica espacial (la de ambientes), el *ccp* o alguna parte de él, podría modelar sistemas móviles.
5. Modelar el comportamiento del sistema de regulación hipotalámica de peso corporal con programación concurrente por restricciones.
6. Determinar si es necesario extender el cálculo para modelar propiedades de movilidad.

4. La Movilidad en Sistemas Reactivos

Un objeto (o proceso) se mueve cuando cambia sus conexiones de comunicación que puede ser visto como enlaces de interacción con otros procesos, o cuando cambia su localización o configuración espacial. Bajo estos dos puntos de vista existen formalismos como cálculos nominales para modelar sistemas móviles: el π-cálculo y el cálculo de ambientes. En el primero la localización esta determinada por una configuración de red circundante (nivel de conectividad con otros procesos). En el segundo la localización es modelada explícitamente por medio de entidades sintácticas que permiten modelar jerarquias de locaciones [Nestmann06]. Existen otros formalismos como $PiCO$, que es un cálculo con la noción de objeto localizado como una restricción y la movilidad aqui es modelada con el mecanismo de delegación, en el sentido que hay un cambio del objeto que recibe el mensaje. [DiazRueda01].

Un sistema reactivo puede ser considerado como un sistema de computación que esta en contínua interacción con su entorno, de manera que actua de inmediato ante los estímulos producidos por este[Riesco02]. Visto de esta manera, la movilidad es una propiedad relevante en estos sistemas porque los agentes pueden crear enlaces de comunicación con otros agentes de su entorno dependiendo del conocimiento que tengan de este. Por ejemplo, si estoy en una habitación y se que en ella no hay objetos con los que yo pueda interactuar, puedo intentar cambiar mi ubicación saliendo de la habitación, probablemente al salir pueda darme cuenta de la presencia de otros objetos con los cuales si puedo interactuar y establecer comunicación.

Las personas invidentes no tienen la capacidad de ver que hay en su entorno. Ellos deducen su entorno a partir de información parcial que han recogido sobre él. Dependiendo de los hechos que puedan deducir adquieren capacidad para interactuar con los objetos de su entorno. Es decir se establecen canales de comunicación como en π-cálculo y se puede cambiar de ámbito para interactuar con otros objetos como en el cálculo de ambientes. Teniendo que estas acciones se pueden realizar dependiendo del conocimiento que surge a partir de una información parcial del entorno.

Ambientes Móviles

El cálculo de ambientes es uno de los formalismos nominales para modelar sistemas móviles. Aquí se ve la movilidad como la evolución de configuraciones espaciales sobre el tiempo [CarGor97]. En este cálculo los ambientes cumplen las siguientes características: Un ambiente puede estar dentro de otro ambiente. Un ambiente es un lugar limitado donde ocurre una computación. Un ambiente puede moverse como un todo. Cada ambiente tiene un nombre y una colección de subambientes. Los nombres son muy importantes, pueden ser creados o proporcionados para nombrar ambientes algunas veces de cuyas acciones pueden ser extraídas. A continuación se presenta la sintaxis de este cálculo de procesos.

$P,Q,R ::=$	processes	
$\mathbf{0}$	void	spatial
$P \mid Q$	composition	spatial
$!P$	replication	spatial
$M[P]$	ambient	spatial
$M.P$	capability action	temporal
$(n).P$	input action	temporal
$\langle M \rangle$	output action	temporal
$M ::=$	messages	
n	name	names
$in\ M$	can enter into M	capabilities
$out\ M$	can exit out of M	capabilities
$open\ M$	can open M	capabilities
ε	null	paths
$M.M'$	composite	paths

Sintaxis del cálculo de Ambientes

Las operaciones de reducción que permiten expresar el comportamiento dinámico de ambientes en el sentido de poder entrar en un ambiente, poder salir de un ambiente o poder abrir el contenido del ambiente, son:

$$
\begin{array}{ll}
n[in\ m.\ P \mid Q] \mid m[R] \longrightarrow m[n[P \mid Q] \mid R] & \text{(Red In)} \\
m[n[out\ m.\ P \mid Q] \mid R] \longrightarrow n[P \mid Q] \mid m[R] & \text{(Red Out)} \\
open\ n.\ P \mid n[Q] \longrightarrow P \mid Q & \text{(Red Open)} \\
(n).P \mid \langle M \rangle \longrightarrow P\{n \leftarrow M\} & \text{(Red Comm)} \\
P \longrightarrow Q \ \Rightarrow\ n[P] \longrightarrow n[Q] & \text{(Red Amb)} \\
P \longrightarrow Q \ \Rightarrow\ P \mid R \longrightarrow Q \mid R & \text{(Red Par)} \\
P' \equiv P,\ P \longrightarrow Q,\ Q \equiv Q' \ \Rightarrow\ P' \longrightarrow Q' & \text{(Red } \equiv \text{)} \\
\longrightarrow^{*} \text{ is the reflexive and transitive closure of } \longrightarrow &
\end{array}
$$

Reglas de Reducción del cálculo de Ambientes

En la primera regla de reducción se modela la acción de un ambiente con nombre n y con procesos P y Q en su interior, para entrar como un todo a otro ambiente llamado m. En la segunda relga se muestra la acción opuesta, salir de un ambiente. La tercera regla modela la capacidad para abrir el contenido de un ambiente y su interior queda expuesto. Se presenta la regla de comunicación. Por ejemplo el proceso $a[p[out\, a.in\, b. < m >]]|b[open\, p.(x).x[]]$ representa un proceso p que viaja saliendo del ambiente a, entrando al ambiente b, donde es abierto. Contiene un mensaje m que es leido y usado para crear un nuevo ambiente.

$$
\begin{array}{ll}
a[p[out\ a.\ in\ b.\ \langle m \rangle]] \mid b[open\ p.\ (x).\ x[]] & \\
\longrightarrow a[] \mid p[in\ b.\ \langle m \rangle] \mid b[open\ p.\ (x).\ x[]] & \text{(Red Out)} \\
\longrightarrow a[] \mid b[p[\langle m \rangle] \mid open\ p.\ (x).\ x[]] & \text{(Red In)} \\
\longrightarrow a[] \mid b[\langle m \rangle \mid (x).\ x[]] & \text{(Red Open)} \\
\longrightarrow a[] \mid b[m[]] & \text{(Red Comm)}
\end{array}
$$

Lógicas Espaciales

Las lógicas espaciales son un formalismo para especificar propiedades espaciales como jerarquía de localizaciones en el cálculo de ambientes. Como una lógica modal el valor de verdad depende del estado. En este caso depende del aqui (espacio) y el ahora (tiempo). Cada fórmula habla sobre el tiempo actual como estado actual de ejecución y lugar actual como localización actual. Por ejemplo la formula $n[0]$ es leida: Aqui y ahora hay una lozalización vacia llamada n. También se ha usado para describir el comportamiento y estructura espacial de sistemas concurrentes [CairCard07].

η, μ	a name n or a variable x
$\mathcal{A}, \mathcal{B}, \mathcal{C} ::=$	
$\mathbf{T}$	true
$\neg\mathcal{A}$	negation
$\mathcal{A} \vee \mathcal{B}$	disjunction
$\mathbf{0}$	void
$\mathcal{A} \mid \mathcal{B}$	composition
$\mathcal{A} \rhd \mathcal{B}$	guarantee
$\eta[\mathcal{A}]$	location
$\mathcal{A}@\eta$	placement
η®$\mathcal{A}$	revelation
$\mathcal{A} \oslash \eta$	hiding
$\Diamond\mathcal{A}$	sometime modality
✧$\mathcal{A}$	somewhere modality
$\forall x.\mathcal{A}$	universal quantification

Formulas lógicas

Las siguientes son reglas de satisfacción para decir cuando los procesos del cálculo de ambientes cumplen con las propiedades descritas con formulas de la lógica espacial definida.

$\forall P{:}\Pi.$	$P \vDash \mathbf{T}$		
$\forall P{:}\Pi, \mathcal{A}{:}\Phi.$	$P \vDash \neg\mathcal{A}$	$\triangleq$	$\neg\, P \vDash \mathcal{A}$
$\forall P{:}\Pi, \mathcal{A},\mathcal{B}{:}\Phi.$	$P \vDash \mathcal{A} \vee \mathcal{B}$	$\triangleq$	$P \vDash \mathcal{A} \vee P \vDash \mathcal{B}$
$\forall P{:}\Pi.$	$P \vDash \mathbf{0}$	$\triangleq$	$P \equiv \mathbf{0}$
$\forall P{:}\Pi, n{:}\Lambda, \mathcal{A}{:}\Phi.$	$P \vDash n[\mathcal{A}]$	$\triangleq$	$\exists P'{:}\Pi.\ P \equiv n[P'] \wedge P' \vDash \mathcal{A}$
$\forall P{:}\Pi, \mathcal{A},\mathcal{B}{:}\Phi.$	$P \vDash \mathcal{A} \mid \mathcal{B}$	$\triangleq$	$\exists P',P''{:}\Pi.\ P \equiv P' \mid P''$ $\wedge P' \vDash \mathcal{A} \wedge P'' \vDash \mathcal{B}$
$\forall P{:}\Pi, x{:}\vartheta, \mathcal{A}{:}\Phi.$	$P \vDash \forall x.\mathcal{A}$	$\triangleq$	$\forall m{:}\Lambda.\ P \vDash \mathcal{A}\{x \leftarrow m\}$
$\forall P{:}\Pi, \mathcal{A}{:}\Phi.$	$P \vDash \Diamond\mathcal{A}$	$\triangleq$	$\exists P'{:}\Pi.\ P \rightarrow^{*} P' \wedge P' \vDash \mathcal{A}$
$\forall P{:}\Pi, \mathcal{A}{:}\Phi.$	$P \vDash$ ✧$\mathcal{A}$	$\triangleq$	$\exists P'{:}\Pi.\ P \downarrow^{*} P' \wedge P' \vDash \mathcal{A}$
$\forall P{:}\Pi, \mathcal{A}{:}\Phi.$	$P \vDash \mathcal{A}@n$	$\triangleq$	$n[P] \vDash \mathcal{A}$
$\forall P{:}\Pi, \mathcal{A},\mathcal{B}{:}\Phi.$	$P \vDash \mathcal{A} \rhd \mathcal{B}$	$\triangleq$	$\forall P'{:}\Pi.\ P' \vDash \mathcal{A} \Rightarrow P \mid P' \vDash \mathcal{B}$

Fórmulas Lógicas

Reglas de satisfacción para los procesos del cálculo de ambientes con la lógica espacial.

Esta lógica cuenta con reglas de inferencia, para poder deducir fórmulas a partir de un conjunto de fórmulas llamadas axiomas[CarGor03]. Un ejemplo simple de propiedades que pueden ser especificadas con esta lógica son movilidad, revelación, igualdad de nombres y ocultamiento

Propiedades espaciales de cálculos concurrentes

Estas lógicas espaciales también han sido usadas para describir propiedades de estructura espacial, de frescura, secrecidad, y revelación de nombres en cálculos concurrentes, donde un sistema restringe el uso de ciertos recursos a ciertos subsistemas, en particular se han estudiado estas propiedades en el πcálculo.

$$
\begin{array}{lll}
[\![\mathbf{F}]\!]_v & \triangleq & \emptyset \\
[\![A \wedge B]\!]_v & \triangleq & [\![A]\!]_v \cap [\![B]\!]_v \\
[\![A \Rightarrow B]\!]_v & \triangleq & \{P \mid \mathit{if}\ P \in [\![A]\!]_v\ \mathit{then}\ P \in [\![B]\!]_v\} \\
[\![\mathbf{0}]\!]_v & \triangleq & \mathbf{1} \\
[\![A|B]\!]_v & \triangleq & [\![A]\!]_v \otimes [\![B]\!]_v \\
[\![A \triangleright B]\!]_v & \triangleq & \{P \mid \mathit{Forall}\ Q.\ \mathit{if}\ Q \in [\![A]\!]_v\ \mathit{then}\ P|Q \in [\![B]\!]_v\} \\
[\![n \circledR A]\!]_v & \triangleq & \{P \mid \mathit{Exists}\ Q.\ P \equiv (\boldsymbol{\nu} n)Q\ \mathit{and}\ Q \in [\![A]\!]_v\} \\
[\![A \oslash n]\!]_v & \triangleq & \{P \mid (\boldsymbol{\nu} n)P \in [\![A]\!]_v\} \\
[\![m\langle n\rangle]\!]_v & \triangleq & \{P \mid P \equiv m\langle n\rangle\} \\
[\![\forall x.A]\!]_v & \triangleq & \bigcap_{n \in \Lambda} [\![A\{x \leftarrow n\}]\!]_v \\
[\![\text{И} x.A]\!]_v & \triangleq & \bigcup_{n \notin fn^v(A)} ([\![A\{x \leftarrow n\}]\!]_v \setminus \{P \mid n \in fn(P)\}) \\
[\![\Diamond A]\!]_v & \triangleq & \{P \mid \mathit{Exists}\ Q.\ P \rightarrow Q\ \mathit{and}\ Q \in [\![A]\!]_v\} \\
[\![X]\!]_v & \triangleq & v(X) \\
[\![\forall X.A]\!]_v & \triangleq & \bigcap_{\Psi \in \mathbb{P}_{\_}} [\![A]\!]_{v[X \leftarrow \Psi]}
\end{array}
$$

Reglas de satisfacción de procesos del π-cálculo para fórmulas de la lógica espacial

5. Programación Concurrente por Restricciones

La programación concurrente por restricciones CCP es un modelo computacional donde cambia la idea original de variables de asignación y lectura como mecanismo de sincronización por un almacén de restricciones donde se imponen restricciones a través de un operador *tell* y se ejecutan procesos si ciertas restricciones se pueden deducir de las guardadas en el almacén de restricciones con un operador *ask*, [SarRP90]

Syntax.

$$\begin{array}{l} P::=D.A \\ D::=\epsilon \mid p(X) :: A \mid D.D \\ A::=c \mid c \rightarrow A \mid A \wedge A \mid \exists X A \mid p(X) \end{array}$$

Semantic Equations.

$$\begin{array}{l} \mathcal{A}(c)e = \{d \in |D|_0 \mid d \geq c\} \\ \mathcal{A}(c \rightarrow A)e = \{d \in |D|_0 \mid d \geq c \Rightarrow d \in \mathcal{A}(A)e\} \\ \mathcal{A}(A \wedge B)e = \{d \in |D|_0 \mid d \in \mathcal{A}(A)e \wedge d \in \mathcal{A}(B)e\} \\ \mathcal{A}(\exists X A)e = \{d \in |D|_0 \mid \exists c \in \mathcal{A}(A)e . \exists_X d = \exists_X c\} \\ \mathcal{A}(p(X))e = \exists_\alpha(d_{\alpha X} \sqcup e(p)) \\ \mathcal{D}(\epsilon)e = e \\ \mathcal{D}(p(X) :: A.D) = \mathcal{D}(D)e[p \mapsto \exists_X(d_{\alpha X} \sqcup \mathcal{A}(A)e)] \\ \mathcal{P}(D.A) = \mathcal{A}(A)(\ \underline{\mathbf{fix}}\ \mathcal{D}(D)) \end{array}$$

Sintaxis y Semántica del Cálculo CCP

El almacen de restricciones en ccp es monotónico, es decir solo de pueden adicionar restricciones en él y estas no pueden modificarse o eliminarse. Pensando en definir un sistema de restricciones para modelar comportamiento de movilidad a través de propiedades espaciales, es necesario que el almacen de restricciones permita camdiar las restricciones o la información parcial sobre la configuración espacial. En especial al quere modelar cuando un proceso cambia de ubicación. Pensando en esto se debe intentar trabajar con un cálculo que permita estos cambios en el sistema de restricciones. *tcc* es un cálculo que deriva de *ccp* adiciona la noción de tiempo el cual es conceptualmente dividido en unidades de tiempo discretas. El almacen de restricciones no se mantiene a través del tiempo, a no ser que se especifice la imposición de una restricción como un proceso que replica en el tiempo De esta manera se intentará definir el sistema de restricciones basado en la lógica espacial para este tipo de cálculo.

Cálculo *utcc*

El cálculo *utcc* incrementa la expresividad de tcc permitiendo comportamiento infinito y movilidad imitando el mecanismo de paso de nombres como se hace en π-cálculo[OlPalVal07].

$$\begin{array}{rl} P, Q := & \mathbf{skip} \mid \mathbf{tell}(c) \mid (\mathbf{abs}\ \vec{x}; c)\, P \mid P \parallel Q \mid \\ & (\mathbf{local}\, \vec{x}; c)\, P \mid \mathbf{next}\, P \mid \mathbf{unless}\ c\ \mathbf{next}\, P \mid\ !\, P \end{array}$$

Sintaxis de utcc

El proceso **skip** no hace nada. El proceso **tell** (c) adiciona la restricción c al almacen de restricciones. (**abs** $x; c$) P puede ser visto como una λ-abstracción del proceso P de las variables x sobre la restricción c. Desde una perspectiva de lenguaje de programación (**local** $x; c$) P puede ser visto como variables locales de P y (**abs** $x; c$) P como parametros formales de P. **next** P ejecuta P en la siguiente unidad de tiempo y **unless** c **next** P ejecuta P en la siguiente unidad de tiempo si y sólo si c no se puede deducir del almacen de restricciones. $!P$ significa que P se ejecutara desde la unidad de tiempo actual en adelante.

utcc tiene una fuerte correspondencia con una lógica temporal de tiempo lineal (LTL) proporcionando una codificación de procesos *utcc* en fórmulas de LTL.

$$
\begin{array}{lcllcl}
[\![\mathbf{skip}]\!] & = & \mathbf{true} & [\![\mathbf{tell}(c)]\!] & = & c \\
[\![(\mathbf{abs}\ \vec{y}; c)\ P]\!] & = & \forall \vec{y}(c \Rightarrow [\![P]\!]) & [\![P \parallel Q]\!] & = & [\![P]\!] \wedge [\![Q]\!] \\
[\![(\mathbf{local}\ \vec{x}; c)\ P]\!] & = & \exists \vec{x}(c \wedge [\![P]\!]) & [\![\mathbf{next}\ P]\!] & = & \circ [\![P]\!] \\
[\![\mathbf{unless}\ c\ \mathbf{next}\ P]\!] & = & c \vee \circ [\![P]\!] & [\![!\ P]\!] & = & \Box [\![P]\!]
\end{array}
$$

Semantica de utcc

Así como existe la correspondencia de *utcc* con la lógica lineal temporal, en esta propuesta de investigación se prentende estudiar si existe correspondencia entre los procesos *utcc* y fórmulas de la lógica espacial definida para ambientes.

6. Biología de Sistemas

La biologia de sistemas es el estudio de un organismo visto como una red integrada e interactiva de genes, proteinas y reacciones bioquímicas las cuales dan origen a la vida. La teoría de la concurrencia y en particular los cálculos de procesos son herramientas emergentes para proporcionar fundamentos formales de la biología sistémica. Modelar es describir sistemas usando un lenguaje preciso y formal útil para reorganizar el conocimiento, simulación, predicción de propiedades y comportamiento de un sistema.

¿Qué se puede modelar en biología?

- Redes de interacción de proteinas
- Redes de regulación genética.
- Células
- Organos y organismos.

La biología de sistemas se plantea entonces el siguiente reto: desarrollar modelos matemáticos robustos y precisos cuya aplicación permita describir, comprender y realizar predicciones sobre sistemas y procesos biológicos complejos. Se han desarrollado formalismos inspirados en los procesos biológicos como el cálculo de membranas. Bioambients es un cálculo que combina procesos del cálculo de ambientes y del π-cálculo para modelar sistemas moleculares complejos donde las moleculas se mueven entre compartimentos.

Sistema hipotalámico de regulación de peso corporal.

Es el ejemplo escogido para hacer una abstraccion de compartimentos biológicos con bioambients [Reg03]. Involucra diversos niveles de organización biológica: molecular, celular y anatómica. El balance entre la energía tomada, gastada y almacenada es controlada por un sistema de retroalimentación para la regulación del peso corporal involucrando mecanismos moleculares y fisiológicos complejos. El principal controlador de este sistema esta ubicado en la región hipotalámica del cerebro. Esta región recibe señales de entrada en la forma de hormonas moleculares secretadas por varios tejidos en el cuerpo y utliza un circuito bioquímico y neuronal complejo que integra estos y envia señales de salida, el cual afecta varias funciones fisiológicas relacionadas con un conjunto de fenómenos de autoregulación de la energía.

La respuesta de estas hormonas es computada por el balance entre el sistema molecular orexigenico, el cual induce la acumulación de energía y anorexigenico, el cual induce la expulsión de energía.

El hipotálamo esta organizado en compartimentos distintos donde residen uno o mas tipos diferentes de células neuronales. Enfocandose en el nucleo ARC y limitandose por una vista relativamente simplificada de eventos., compuesto de dos grandes pasos. En el primer paso se distingue entre las neuronas NPY/AgRP que producen la neuropeptida orexigénica Y (NPY) , hormonas AgRP y neuronas POMC/CART que producen neuropeptidos anorexigénicas αMSH y CART. Ambas neuronas albergan receptores de las hormonas moleculares leptina e insulina.

Los niveles de leptina circulante le dan al cerebro una indicación del almacenamiento de energía con el propósito de regular el apetito y el metabolismo. La leptina funciona inhibiendo la actividad de las neuronas que contienen neuropéptido Y (NPY) y péptido relacionado con el agoutí (AgRP); y también incrementando la actividad de las neuronas que expresan la hormona estimulante de los melanocitos α (α-MSH). Las neuronas NPY son el elemento clave en la regulación del apetito; pequeñas dosis de NPY inyectadas dentro del cerebro de animales experimentales estimulan la alimentación, mientras que la destrucción selectiva de las neuronas NPY en ratones los convierten en anoréxicos. Por el contrario, el α-MSH es un mediador importante de la saciedad, y las diferencias en el gen del receptor sobre el que actúa en el cerebro están relacionadas con la obesidad.

Aunque la leptina es una señal que reduce el apetito, en general la gente obesa tiene una concentración inusualmente alta de leptina circulante; estas personas son resistentes a la leptina, de modo similar a la resistencia de los diabéticos a la insulina. Así, la obesidad se desarrolla cuando la persona ingiere más energía de la que usa durante un periodo prolongado de tiempo, y este exceso de ingesta de comida no es disparado por las señales de hambre, prolongándose a pesar de las señales anti-apetito de la leptina. Las concentraciones regularmente altas de leptina a partir de los grandes almacenes de grasa provocan que las células que responden a la leptina se desensibilicen.

7. Justificación

Muchos sistemas reactivos entre ellos los sistemas biológicos, de multimedia y de seguridad, presentan propiedades de movilidad que son relevantes en el comportamiento global del sistema, estudiándolos desde el punto de vista de procesos que intercambian canales de comunicación o de procesos que cambian de ubicación en una jerarquía de locaciones. Este trabajo pretende presentar un modelo computacional que integre estos enfoques basándose en formalismos especificados para cada uno de ellos.

Pensando en la Investigación científica como un saber hacer para producir modelos teóricos que contribuyan a una mejor comprensión de la realidad y facilite la detección y resolución de problemas concretos, este trabajo conduce a entender o analizar el funcionamiento de sistemas biológicos.

La leptina es un regulador hormonal de muchos procesos biológicos y como tal interviene en la fisiopatología de un amplio número de enfermedades. Entonces estudiar el sistema hipotalámico de regulación de peso corporal a partir de modelos teóricos puede dar luces para entender las causas de ciertas enfermedades y trastornos que tienen que ver con el apetito y el metabolismo del cuerpo.

8. Metodología de investigación.

1. Definir formalmente el sistema de restricciones basado en las lógicas espaciales definidas para el cálculo de ambientes.

2. Escoger un ejemplo sencillo modelado con el cálculo de ambientes y modelarlo con *ccp* y el sistema de restricciones definido.

3. Verificar si *utcc* satisface propiedades de movilidad especificadas en lógicas espaciales para ambientes.

4. Verificar si se satisfacen las propiedades descritas con lógicas espaciales para procesos del cálculo de ambientes.

5. Seleccionar un ejemplo biológico con propiedades de un sistema móvil y modelarlo con *ccp* y el sistema de restricciones definido.

9. Resultados esperados

1. Reglas de satisfacción para procesos del cálculo *ccp* como modelos de fórmulas lógicas espaciales.

2. Modelo de un ejemplo de sistema móvil con *ccp*
3. Modelo del sistema de regulación de peso corporal con propiedades móviles en *ccp*.
4. Reglas de satisfacción para procesos de *utcc* en lógicas espaciales.
5. Intento de respuesta a la pregunta sobre si es necesario extender el cálculo para expresar movilidad.

10. Cronograma

Actividad	**No. Semanas**
Definir del sistema de restricciones	1
Semántica del cálculo con el S.R.	1
Modelar un ejemplo sencillo	2
Modelar el sistema de regulación de peso corporal	4
Verificar propiedades de la lógica	2
Borrador de trabajo de investigación	4
Correcciones del trabajo de investigación	4

Referencias

[ArbGut06] Alejandro Arbelaez y Julian Gutierrez. *Estudio exploratorio de la aplicación de la programación concurrente por restricciones en bioinformática.* Tesis de grado Universidad Javeriana Cali, 2006

[CarGor97] Luca Cardelli y Andrew Gordón. *Mobile Ambients.* 1997.

[DiazRueda01] Juan F. Díaz y Camilo Rueda. *Modelos para la Computación Móvil.* Avispa, 2001

[CarGor03] Luca Cardelli y Andrew Gordon. *Ambient Logic.* 2003

[CairCard07] Luis Caires y Luca Cardelli. *A Spatial Logic for Concurrency (Part I).* 2007

[Nestmann06] Uwe Nestmann. *Welcome to the Jungle. A subjective guide to mobile process calculi.* Universidad de Berlin, 2006

[OlPalVal07] Carlos Olarte, Catuscia Palamidesi y Frank Valencia. *Universal Timed Concurrent Constraint Programming.* 2007

[Reg03] Aviv Regev, E. Panina, W Silverman, L Cardelli y E. Shapiro. *BioAmbients: An abstraction for biological compartments.* 2003

[Riesco02] Miguel Riesco Albizu. *Especificación de Sistemas Reactivos Distribuidos utilizando Estelle Síncrono.* Tesís Doctoral, Universidad de Oviedo.2002.

[SarRP90] Vijay A Saraswat, Martin Rinard y Prakash Pamangaden. *Semantic foundations of concurrent constraint programming. 1990*