VNIVERSIDAD NACIONAL
AVTNOMA DE
MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"UNA LÓGICA DINÁMICA EPISTÉMICA PARA LA COMUNICACIÓN ASÍNCRONA POR CANALES INSEGUROS"

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS
(COMPUTACIÓN)

P R E S E N T A :

PEDRO ARTURO GÓNGORA LUNA

DIRECTOR DE TESIS: FRANCISCO HERNÁNDEZ QUIROZ

México D.F. 2008.

Agradecimientos

A mis compañeros y profesores del posgrado, por hacer de la maestría una experiencia sumamente grata. Al CONACyT, a la UNAM, sus facultades e institutos por permitirme estudiar y desarrollarme en mi país haciendo lo que más me gusta: computación.

A mi tutor, Francisco Hernández Quiroz, por atraerme a este mundo fascinante de la lógica y la computación teórica. A mis sinodales por regalarme algo de su tiempo leyendo mi trabajo y procurar su calidad.

En Amsterdam, a los doctores Johan van Benthem y Jan van Eijck y al *Institute for Logic, Language and Computation* por recibirme, escucharme y orientarme. A Fernando y Woyl, no sólo por su hospitalidad, sino también por su orientación y comentarios acerca de mi trabajo.

A mis padres y hermanas porque siempre puedo contar con su apoyo.

Por último, debo decir que no se puede emprender una tarea como esta sin restar atención a otros aspectos de la vida personal. A pesar esto, cuando siguen incondicionalmente a tu lado, sin restar apoyo, cariño e interés por tí, sólo puede ser gracias a un verdadero amor. Es con ese mismo amor que le dedico este pequeño esfuerzo a mi esposa: Selene.

Índice general

Resumen

La lógica dinámica epistémica agrupa una familia de lenguajes formales que buscan describir y razonar sobre el conocimiento que tienen los individuos de un sistema multiagente y cómo evoluciona éste por medio de acciones de comunicación. Por otra parte, el cálculo π es un álgebra de procesos que describe el comportamiento de procesos concurrentes, los cuales evolucionan por medio de intercambio de mensajes entre ellos.

En este trabajo se presenta la semántica formal de una lógica dinámica epistémica. Se busca describir los cambios en el conocimiento de los agentes de un sistema como consecuencia de la ejecución de programas basados en una variante del cálculo π con comunicación asíncrona.

El lenguaje presentado se traduce primero a un lenguaje más general. Para el segundo lenguaje se presenta la semántica formal en términos de las herramientas matemáticas más comunes para este tipo de lenguajes: las estructuras de Kripke como modelos. El resultado de la ejecución de un proceso se modela como una función que realiza los ajustes necesarios a un modelo para describir el cambio en el conocimiento de los agentes. Como resultado final se comprueba que las operaciones propuestas preservan las equivalencias de modelos y de procesos.

El lenguaje presentado puede ser útil para la tarea de verificación formal de propiedades de sistemas multiagente, sólo que agrupa en una herramienta las capacidades de las lógicas epistémicas y de las álgebras de procesos. También, utilizando las mismas herramientas matemáticas, puede extenderse el trabajo con otras variantes del cálculo π o lenguajes similares que abarquen otros tipos de comunicación.

Capítulo 1

Introducción

1.1. Motivación y antecedentes

Desde sus inicios, las ciencias computacionales han mantenido una relación estrecha con la lógica. Además de ser el fundamento de una buena parte de la teoría de la computación, la lógica tiene diversas aplicaciones, por ejemplo, en inteligencia artificial, especificación formal de lenguajes de programación, y muchas otras más. El éxito de todas estas aplicaciones se debe en parte al poder expresivo que tienen los lenguajes lógicos para modelar y razonar sobre múltiples situaciones, tanto del mundo real, como del universo de las computadoras.

Ejemplos de este tipo de situaciones hay muchos. Uno muy recurrido en la literatura del tema (de este trabajo) es el problema del consenso. Este problema, plantea un escenario en donde un conjunto de agentes deben acordar la realización de un evento propuesto por alguno de ellos, pero con limitaciones en la forma en que pueden comunicarse. La limitante es que los agentes de este sistema sólo pueden enviarse mensajes asíncronos por un medio de comunicación inseguro. Esto es, cuando un agente envía un mensaje no tiene la certeza de cuánto tiempo tardará en recibirse, más aún, tampoco tiene la certeza de que su mensaje sea recibido en última instancia.

La instancia más simple de este problema se presenta con dos agentes, por ejemplo 1 y 2. Si el primer agente propone p, el objetivo es que después de un cierto número de intercambio de mensajes, ambos agentes acuerden que p es la propuesta, pues su comportamiento va a depender de eso. Si el agente 1 envía un mensaje asíncrono a 2 con el contenido p, no puede decir si el segundo se enteró, es decir, no sabe si 2 sabe p. Si se supone que el mensaje llega a su destino, 2 podría enviar un acuse, pero esto no es suficiente, pues no sabe si

traducción que convierte sus enunciados a enunciados del lenguaje del capítulo 4. De esta manera, la semántica se define en términos del lenguaje anterior. Dada esta traducción, se procede a demostrar las propiedades básicas de esta semántica, las cuales son, básicamente, la preservación de bisimulación entre modelos y entre procesos del cálculo π.

Como es común, el trabajo termina con una discusión de sus límites, las relaciones que guarda con otros trabajos previos, además de presentar algunas líneas de mejora y aplicación.

1.4. Nomenclatura

A lo largo del texto se utilizarán los siguientes lineamientos:

- Se usarán letras mayúsculas (A, B, C, etc.) para nombrar tuplas y conjuntos de elementos simples;
- Los nombres de conjuntos de elementos sintácticos de un lenguaje se escribirán en estilo caligráfico ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, etc.);
- Las clases o conjuntos cuyos elementos son tuplas se nombrarán con mayúsculas negritas ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, etc.);
- Se usarán letras minúsculas del alfabeto griego (α, β, γ, etc.) para nombrar elementos arbitrarios de un conjunto o metavariables de un lenguaje;
- Las secciones particulares de texto como definiciones, teoremas o convenciones, comparten la misma numeración por capítulo. Por ejemplo, 2.6 se refiere a la definición de consecuencia lógica en el capítulo 2, y 2.8 al teorema que le sigue unos párrafos después;
- Se usará algunas veces “sii” como abreviatura de “si, y sólo si,”.

Capítulo 2

Lógica dinámica epistémica

En este capítulo se hará una breve presentación de los lenguajes epistémicos y dinámicos más importantes. La presentación comienza con la introducción de la herramienta semántica más importante en las lógicas modales, los marcos de Kripke/Hintikka [Kri63, Hin62]. Se presentan también las clases de estructuras más importantes y algunas de sus propiedades. En seguida, se presenta brevemente la lógica epistémica multiagente como una aplicación de la lógica modal, y a manera de introducción a los lenguajes epistémicos.

La lógica de anuncios públicos es el ejemplo más sencillo de los lenguajes dinámicos epistémicos, y conviene su presentación como introducción al tema. Seguido, se presenta el concepto de bisimulación entre modelos, útil como medida de equivalencia. La lógica de acciones epistémicas es un lenguaje elegante y expresivo, además de que introduce el concepto de estructuras de acciones, usado posteriormente en el lenguaje propuesto en el capítulo 4.

Para el tema de la lógica epistémica la referencia obligada es [Hin62], donde se puede leer una introducción de la discusión en el sentido filosófico sobre qué tipo de conocimiento es el que describe esta lógica. Sin embargo, debido a su poca disponibilidad puede consultarse una excelente introducción en [FHMV95] con un punto de vista computacional. Para una introducción a la lógica modal, incluyendo una discusión más detallada sobre modelos y bisimulación, se recomienda al lector la consulta del texto [BdRV01].

Algunas partes de este capítulo se basan en los textos ya mencionados, excepto cuando se señala explícitamente otra fuente.

2.1. Lógica modal y marcos de Kripke/Hintikka

Todos los lenguajes presentados en este trabajo son extensiones de la lógica modal. Por lo tanto, para una presentación ordenada es conveniente introducir primero esta lógica. En esta sección se presentan los conceptos básicos que servirán como fundamento para el resto de la tesis. También, con este primer lenguaje se introduce la dinámica de exposición que se mantendrá en todo el trabajo. Esto es, se presenta primero la sintaxis formal del lenguaje junto con una explicación intuitiva de ésta, para después presentar su semántica formal y algunas propiedades.

Sin más introducción, se presenta la definición de cómo se construyen las oraciones de la lógica modal, esto es, el conjunto de sus fórmulas.

Definición 2.1 (Sintaxis de $\mathcal{L}_{ML}$)**.** Sea $\mathcal{P}$ un conjunto numerable de símbolos de proposición. Se define *el conjunto $\mathcal{L}_{ML}$ de las fórmulas de la lógica modal* como el menor conjunto tal que:

(a) si $p \in \mathcal{P}$, entonces $p \in \mathcal{L}_{ML}$;

(b) $\bot \in \mathcal{L}_{ML}$ ($\bot \notin \mathcal{P}$);

(c) si $\varphi \in \mathcal{L}_{ML}$, entonces $\neg\varphi \in \mathcal{L}_{ML}$;

(d) si $\varphi, \psi \in \mathcal{L}_{ML}$, entonces $(\varphi \vee \psi) \in \mathcal{L}_{ML}$;

(e) si $\varphi \in \mathcal{L}_{ML}$, entonces $\Box\varphi \in \mathcal{L}_{ML}$. ∎

La definición inductiva anterior indica la estructura de las fórmulas de $\mathcal{L}_{ML}$. Los elementos más básicos de este lenguaje son las proposiciones atómicas. Para referirse a éstas se usarán las letras p, q, r, etc. Los operadores $\neg$ y $\vee$ mantienen su significado intuitivo de *negación* y *conjunción* lógicas.

El objetivo es que cada fórmula pueda evaluarse a un valor de verdad, esto es, *verdadero* o *falso*. De esta forma, cada proposición atómica está ligada directamente con uno de esos valores.

Por ejemplo, si la proposición atómica p está asociada a un valor de verdad *falso*, entonces la fórmula $\neg p$ se evaluará como *verdadera*. Adicionalmente, si el átomo q se asocia a un valor *verdadero*, entonces la evaluación de $(p \vee q)$ también resulta en un valor *verdadero*, y el resultado de $(p \vee \neg q)$ es *falso*.

La constante $\bot$ o falsedad, siempre estará asociada al valor de verdad *falso*. Para el operador modal $\Box$ se usará la intuición habitual de "*necesariamente es cierto que...*". Por ejemplo, la fórmula $((\Box p) \vee q)$ se puede parafrasear como "*o necesariamente es verdad que p, o es verdad que q, o ambas*".

En este lenguaje se mantienen las mismas equivalencias del cálculo de proposiciones. Por ejemplo, si p, q y r son proposiciones atómicas en $\mathcal{P}$, entonces la fórmula:

$$\neg(p \vee q) \vee \Box r$$

también puede escribirse como:

$$(p \vee q) \Rightarrow \Box r$$

Es decir: *"si p o q son el caso, entonces necesariamente r es el caso"*.

Como en el ejemplo anterior, los operadores $\wedge$ (conjunción), $\Rightarrow$ (implicación o condicional) y $\Leftrightarrow$ (bicondicional o equivalencia lógica) se definen de la forma usual, además de $\diamond$ (posibilidad), dual de $\Box$, y la constante $\top$ (verdad). A lo largo del texto se asumirán estas convenciones (para este y otros lenguajes presentados posteriormente) resumidas a continuación junto con la precedencia de los operadores:

Convención 2.2. (a) Se definen el resto de los operadores proposicionales de acuerdo con los siguientes esquemas:

$$(\varphi \wedge \psi) \stackrel{\text{def}}{=} \neg(\neg\varphi \vee \neg\psi)$$
$$(\varphi \Rightarrow \psi) \stackrel{\text{def}}{=} (\neg\varphi \vee \psi)$$
$$(\varphi \Leftrightarrow \psi) \stackrel{\text{def}}{=} ((\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi))$$

(b) Se definen la constante $\top$ (verdad) y el operador, $\diamond$ ("*es posible que...*"), dual de $\Box$ como sigue:

$$\top \stackrel{\text{def}}{=} \neg\bot$$
$$\diamond\varphi \stackrel{\text{def}}{=} \neg\Box\neg\varphi$$

(c) Para evitar el exceso de paréntesis y facilitar la lectura de las fórmulas se establece la precedencia de los operadores como sigue: el operador $\neg$ es el de más precedencia, le siguen $\vee$ y $\wedge$ con igual precedencia, luego $\Rightarrow$ y $\Leftrightarrow$ también con igual precedencia, y, finalmente $\diamond$ y $\Box$. ∎

Para definir formalmente el significado intuitivo de los operadores $\diamond$ y $\Box$ se hará uso de la semántica más común para los lenguajes modales: la semántica de *mundos posibles*. Los elementos de esta familia de semánticas se construyen a partir de estructuras relacionales, donde se tiene un universo de posibilidades (mundos posibles) relacionadas entre sí, y donde las proposiciones pueden tomar diferentes valores de verdad.

La satisfacción de las fórmulas es relativa al mundo actual (real), el cual está relacionado con otros mundos posibles. De esta forma,una noción como: "*es posible que* φ", se concretiza en: "*existe al menos un mundo posible y accesible desde el mundo actual, donde* φ *es el caso*".

Por ejemplo, durante el día, dentro de un cuarto cerrado sin ventanas, la afirmación "afuera está soleado" puede ser verdad o no. Si se tiene acceso a una ventana, y efectivamente está soleado, entonces pueden existir otras posibilidades donde, por ejemplo, esté soleado o no en otra ciudad, pero en todas ellas está soleado aquí, por lo que se puede afirmar que *necesariamente* está soleado afuera.

Se comienza la exposición del método semántico de mundos posibles con la definición de las estructuras mencionadas y la interpretación de las fórmulas del lenguaje $\mathcal{L}_{ML}$.

Definición 2.3 (Marco de Kripke/Hintikka)**.** Se define *un marco de Kripke / Hintikka* como la tupla $\mathfrak{K} = \langle \mathcal{W}, R \rangle$, donde

- $\mathcal{W}$ es un conjunto numerable de mundos posibles;
- $R \subseteq \mathcal{W} \times \mathcal{W}$ es una relación binaria (de accesibilidad) entre elementos de $\mathcal{W}$. ∎

Definición 2.4 (Modelos y satisfacción para $\mathcal{L}_{ML}$)**.** *Un modelo para* $\mathcal{L}_{ML}$ es una estructura

$$\mathfrak{M} = \langle \mathfrak{K}, V \rangle$$

donde $\mathfrak{K}$ es un marco de Kripke/Hintikka y

$$V : \mathcal{W} \times \mathcal{P} \to \{\mathbf{V}, \mathbf{F}\}$$

es una valuación que asigna un valor de verdad a cada proposición atómica en cada mundo posible. Dados un modelo, $\mathfrak{M}$, y un elemento w en $\mathcal{W}$, se define *la relación de satisfacción* $\models$*, entre modelos apuntados* $(\mathfrak{M}, w)$ *y las fórmulas de* $\mathcal{L}_{ML}$, como la menor relación tal que:

(a) $\mathfrak{M}, w \models p$ sii $V(w, p) = \mathbf{V}$ (para $p \in \mathcal{P}$);

(b) $\mathfrak{M}, w \not\models \bot$;

(c) $\mathfrak{M}, w \models \neg\varphi$ sii no es el caso que $\mathfrak{M}, w \models \varphi$ (i.e., $\mathfrak{M}, w \not\models \varphi$);

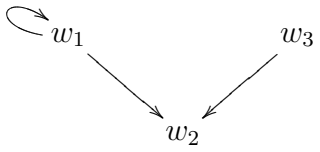
(d) $\mathfrak{M}, w \models (\varphi \vee \psi)$ sii se tiene que $\mathfrak{M}, w \models \varphi$ ó (no exclusivo) $\mathfrak{M}, w \models \psi$;

(e) $\mathfrak{M}, w \models \Box\varphi$ sii para todo $w' \in \mathcal{W}$, si $(w, w') \in R$, entonces $\mathfrak{M}, w' \models \varphi$. ∎

Un modelo puede representarse con una gráfica dirigida, donde los nodos son los mundos posibles y los arcos indican la relación de accesibilidad. Por ejemplo, el modelo $\mathfrak{M}$ con los componentes:

- $\mathcal{W} = \{w_1, w_2, w_3\}$,
- $R = \{(w_1, w_1), (w_1, w_2), (w_3, w_2)\}$,
- $V(w, p) = \mathbf{V}$ para $w = w_1 = w_2$, $\mathbf{F}$ de otra manera,
- $V(w, q) = \mathbf{V}$, para $w = w_3$, $\mathbf{F}$ de otra manera.

puede representarse gráficamente como:



Si se apunta el modelo anterior con w_1, entonces se cumplen las afirmaciones:

$$\mathfrak{M}, w_1 \models p$$
$$\mathfrak{M}, w_1 \models \Box p$$

Sin embargo, si se apunta el modelo con w_3, se tiene que:

$$\mathfrak{M}, w_3 \models q$$
$$\mathfrak{M}, w_3 \not\models \Box q$$

En este ejemplo se resalta que si un modelo apuntado satisface φ, no necesariamente también satisface $\Box\varphi$. Sin embargo, si se tiene que φ es satisfecha por cualquier modelo apuntado, entonces $\Box\varphi$ también lo es, y se puede tomar como axioma. Para esto, hacen falta las definiciones de validez y consecuencia lógica modal.

Definición 2.5 (Validez)**.** Sea φ una fórmula cualquiera en $\mathcal{L}_{ML}$, entonces:

(a) Dado un marco $\mathfrak{K}$ y un elemento w en $\mathcal{W}$ (en $\mathfrak{K}$), se dice que φ *es válida en* w *en el marco* $\mathfrak{K}$ si para toda valuación V, se tiene que el modelo apuntado $(\langle \mathfrak{K}, V\rangle, w)$ satisface φ. En este caso se escribe $\mathfrak{K}, w \models \varphi$;

(b) Dado un marco $\mathfrak{K}$, se dice que φ *es válida en el marco* $\mathfrak{K}$ si para toda w en $\mathcal{W}$ (en $\mathfrak{K}$), se tiene que φ es válida en w en $\mathfrak{K}$. En este caso se escribe $\mathfrak{K} \models \varphi$;

(c) Dada una clase $\mathbf{C}$ de marcos, se dice que φ *es válida en* $\mathbf{C}$ si para todo $\mathfrak{K}$, φ es válida en $\mathfrak{K}$. En este caso se escribe $\mathbf{C} \models \varphi$;

(d) Se dice simplemente que φ *es válida*, si es válida en la clase $\mathbf{K}$ de todos los marcos. En este caso se escribe $\models \varphi$ (ver cuadro 2.1 más adelante para otros nombres de clases comunes). ∎

Definición 2.6 (Consecuencia lógica modal)**.** Sean $\mathbf{C}$ una clase de marcos y Γ un conjunto de fórmulas. Se dice que *una fórmula* φ *es consecuencia (local) de las fórmulas en* Γ si:

$$\mathfrak{M}, w \models \gamma \text{ implica que } \mathfrak{M}, w \models \varphi$$

para todo modelo $\mathfrak{M}$ basado en los marcos en $\mathbf{C}$, para todo w en $\mathfrak{M}$, y para toda γ en Γ. Si esto es el caso, entonces también se puede escribir:

$$\Gamma \models_C \varphi$$

∎

Así como se usa $\mathbf{K}$ para la clase de todos los marcos, para otras clases comunes se tiene una nomenclatura estándar. Estas clases agrupan marcos de acuerdo con las propiedades de su relación de accesibilidad. Los nombres de las clases de marcos más importantes se muestran en el cuadro 2.1.

Diferentes clases de marcos permiten diferentes axiomatizaciones, dependiendo de las propiedades de la relación de accesibilidad. El sistema básico es el sistema $\mathbf{K}$, al que se le pueden agregar los axiomas asociados a las distintas clases de marcos.

Definición 2.7 (Sistema de demostración $\vdash_{\mathbf{K}}$)**.** Se define el *sistema de demostración* $\vdash_{\mathbf{K}}$ como el conjunto formado por los siguientes esquemas y reglas deductivas:

$$\vdash_{\mathbf{K}} \Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi) \qquad \text{(K)}$$

$$\frac{\vdash_{\mathbf{K}} \varphi \qquad \vdash_{\mathbf{K}} \varphi \Rightarrow \psi}{\vdash_{\mathbf{K}} \psi} \qquad \text{(Modus Ponens)}$$

$$\frac{\vdash_{\mathbf{K}} \varphi}{\vdash_{\mathbf{K}} \Box\varphi} \qquad \text{(Generalización)}$$

∎

K	Clase de todos los marcos.
K4	Clase de todos los marcos con relaciones transitivas.
T	Clase de todos los marcos con relaciones reflexivas.
B	Clase de todos los marcos con relaciones simétricas.
KD	Clase de todos los marcos con relaciones seriales.
S4	Clase de todos los marcos con relaciones reflexivas y transitivas.
S5	Clase de todos los marcos con relaciones reflexivas y euclidianas (i.e., de equivalencia).
K4,3	Clase de todos los marcos con relaciones transitivas y no seriales.
S4,3	Clase de todos los marcos con relaciones reflexivas, transitivas y no seriales.
KD45	Clase de todos los marcos con relaciones transitivas, euclidianas y seriales.

Cuadro 2.1: Las clases de marcos de Kripke/Hintikka más comunes

Teorema 2.8 (Sistema de demostración completo y correcto para $\mathcal{L}_{ML}$)**.** *El sistema demostración formado por todas las instancias de tautologías de la lógica proposicional, el axioma K y las reglas Modus Ponens y Generalización, es completo y correcto para $\mathcal{L}_{ML}$ en los modelos basados en la clase* **K**.

De acuerdo con la construcción de la sintaxis del lenguaje en esta tesis, además de las tautologías de la lógica proposicional (que incluyen las definiciones de $\wedge$, $\Rightarrow$ y $\Leftrightarrow$) se debe recordar la dualidad del operador $\diamond$. Si se incluye el operador $\diamond$ directamente en la sintaxis del lenguaje y se define $\Box$ como derivado, entonces debe considerarse un axioma más para el sistema **K**:

$$\vdash_{\mathbf{K}} \diamond\varphi \Leftrightarrow \neg\Box\neg\varphi \qquad \text{(Dual)}$$

Conviene resaltar que la regla (Generalización) se refiere a fórmulas válidas en cualquier marco posible, esto es, anteponer $\Box$ a una fórmula preserva su validez, mas no su satisfacción. Esto no ocurre igual en el caso de la regla (Modus Ponens), la cual sí preserva tanto validez como satisfacción.

Para finalizar esta sección se presentan en el cuadro 2.2 varios axiomas correctos, asociados a algunas de las clases conocidas. Por ejemplo, el axioma (4) es correcto para la clase **K4** y el axioma (.3) es correcto para las clases **K4**,**3** y **S4**,**3**. En conjunto con el sistema **K**, los axiomas del cuadro 2.2 forman lógicas correctas y completas para sus clases (cf. cuadro 2.1). Las lógicas resultantes se nombran de la misma forma que las clases para las que son correctas. Por

$\Box\varphi \Rightarrow \Box\Box\varphi$	(4)
$\Box\varphi \Rightarrow \varphi$	(T)
$\varphi \Rightarrow \Box \diamond \varphi$	(B)
$\Box\varphi \Rightarrow \diamond\varphi$	(D)
$\diamond \varphi \wedge \diamond\psi \Rightarrow \diamond(\varphi \wedge \diamond\psi) \vee \diamond(\varphi \wedge \psi) \vee \diamond(\psi \wedge \diamond\varphi)$	(.3)
$\Box(\Box\varphi \Rightarrow \varphi) \Rightarrow \Box\varphi$	(L)

Cuadro 2.2: Axiomas asociados a distintas clases de marcos

ejemplo, la lógica que contiene los axiomas (4), (T) y (D), forman la lógica **KD45**, y si se le añade el axioma (B), forma la lógica **S5**.

2.2. Lógica epistémica

Una de las múltiples aplicaciones posibles del operador $\Box$ y las estructuras relacionales es la descripción del conocimiento de un agente. Bajo esta perspectiva, dada una relación R de accesibilidad entre mundos posibles, si una pareja (w, w') está en R, entonces se dice que el agente no puede distinguir (epistémicamente) entre los mundos w y w'. De esta forma, se puede afirmar que un agente *conoce* φ, si φ es el caso en cualquier mundo posible indistinguible de la referencia actual.

Definición 2.9 (Sintaxis de $\mathcal{L}_{EL}$)**.** Sean $\mathcal{P}$ un conjunto numerable de símbolos de proposición y $\mathcal{A}$ un conjunto finito de agentes. El *conjunto $\mathcal{L}_{EL}$ de las fórmulas de la Lógica Epistémica* se define como el menor conjunto tal que:

(a) si $p \in \mathcal{P}$, entonces $p \in \mathcal{L}_{EL}$;

(b) $\bot \in \mathcal{L}_{EL}$;

(c) si $\varphi \in \mathcal{L}_{EL}$, entonces $\neg\varphi \in \mathcal{L}_{EL}$;

(d) si $\varphi, \psi \in \mathcal{L}_{EL}$, entonces $(\varphi \vee \psi) \in \mathcal{L}_{EL}$;

(e) si $\varphi \in \mathcal{L}_{EL}$ e $i \in \mathcal{A}$, entonces $K_i\varphi \in \mathcal{L}_{EL}$;

(f) si $\varphi \in \mathcal{L}_{EL}$ y $G \subseteq \mathcal{A}$, entonces $C_G\varphi \in \mathcal{L}_{EL}$. ∎

A este lenguaje se agregan dos nuevos elementos —además de intercambiar el uso de $\Box$ por K de la palabra inglesa *know*, acorde con su interpretación intuitiva—.

El primer elemento nuevo es la capacidad para describir escenarios con más de un agente. Cada agente i del conjunto de agentes posee conocimiento posiblemente diferente al de los demás. Por lo tanto, se tiene un operador K_i para cada agente i. Incluso, un agente puede tener conocimiento acerca del conocimiento de los otros agentes, lo que puede expresarse con fórmulas del tipo $K_iK_j\ldots$

El segundo elemento añadido es el *conocimiento común*. Aquí, la intuición es: φ es del conocimiento común entre los agentes en G, si todo agente i del conjunto G conoce φ, todo agente j en G conoce que todo agente i en G conoce φ, y así sucesivamente para cualquier nivel arbitrario de profundidad.

La afirmación del conocimiento común de φ sobre los agentes en $G \subseteq \mathcal{A}$ es más fuerte que la de afirmar que todos los agentes en G conocen φ. Esto es, la siguiente conjunción:

$$E_G\varphi \stackrel{\text{def}}{=} \bigwedge_{i\in G} K_i\varphi$$

no es equivalente a $C_G\varphi$, pues no implica que para cualquier par de agentes i, j, en G, se sostenga $K_iK_j\varphi$.

Más aún, la fórmula:

$$C_G\varphi \Rightarrow K_{i_1}\ldots K_{i_n}\varphi$$

se sostiene para cualquier secuencia $i_1,\ldots,i_n$ de agentes en G.

En $\mathcal{L}_{EL}$ también es posible definir un operador dual de K:

$$\hat{K}_i\varphi \stackrel{\text{def}}{=} \neg K_i\neg\varphi$$

cuya lectura puede hacerse como: "*El agente i considera epistémicamente posible que φ*".

La interpretación de las fórmulas de $\mathcal{L}_{EL}$ es similar a la de las fórmulas de la lógica modal, sólo que en este caso se extienden los marcos de Kripke asignándole una relación R_i a cada agente i en $\mathcal{A}$.

Definición 2.10 (Modelos y satisfacción para $\mathcal{L}_{EL}$)**.** Sea:

$$\mathfrak{M} = \langle \mathcal{W}, \{R_i\}_{i\in\mathcal{A}}, V\rangle$$

un marco de Kripke/Hintikka, donde a cada agente $i \in \mathcal{A}$, le corresponde una relación sobre mundos posibles:

$$R_i \subseteq \mathcal{W}\times\mathcal{W}$$

Entonces, se define *la relación* $\models$ *entre los modelos apuntados y las fórmulas de* $\mathcal{L}_{EL}$ como la mínima relación tal que:

(a) $\mathfrak{M}, w \models p$ sii $V(w,p) = \mathbf{V}$ (para $p \in \mathcal{P}$);

(b) $\mathfrak{M}, w \not\models \bot$;

(c) $\mathfrak{M}, w \models \neg\varphi$ sii no es el caso que $\mathfrak{M}, w \models \varphi$ (i.e., $\mathfrak{M}, w \not\models \varphi$);

(d) $\mathfrak{M}, w \models (\varphi \vee \psi)$ sii se tiene que $\mathfrak{M}, w \models \varphi$ ó (no exclusivo) $\mathfrak{M}, w \models \psi$;

(e) $\mathfrak{M}, w \models K_i\varphi$ sii para todo $w' \in \mathcal{W}$, si $(w, w') \in R_i$, entonces $\mathfrak{M}, w' \models \varphi$;

(f) $\mathfrak{M}, w \models C_G\varphi$ sii para todo $i \in G$, si $(w, w') \in (R_G)^+$, entonces $\mathfrak{M}, w' \models \varphi$.

Donde:

$$R_G \stackrel{\text{def}}{=} \bigcup_{i \in G} R_i$$

y $(R_G)^+$ es la cerradura transitiva de la relación R_G. ∎

Proveniente de la epistemología, la lógica epistémica pretende, desde luego, ayudar a comprender qué *es* el conocimiento. Desde este punto de vista, existe discusión en filosofía sobre qué tipo de conocimiento puede describirse con esta lógica y en qué casos. A partir de esto, es importante elegir qué axiomas deben tomarse en cuenta al modelar conocimiento. Por ejemplo, las adaptaciones de los axiomas (4), (T) y (D) a la lógica epistémica:

$$K_i\varphi \Rightarrow K_iK_i\varphi$$
$$K_i\varphi \Rightarrow \varphi$$
$$K_i\varphi \Rightarrow \hat{K}_i\varphi$$

parecen razonables al tratar sobre el conocimiento de un agente. Sin embargo, el resultado de añadir el axioma (B) —que, en conjunto con los anteriores, lleva a marcos en la clase **S5**— no es claro en todos los casos:

$$\varphi \Rightarrow K_i\hat{K}_i\varphi$$

El axioma (T) indica que el conocimiento implica verdad. Sin embargo, las lógicas que no cumplen con este axioma no quedan fuera de la interpretación epistémica. En este caso, $K_i\varphi$ puede interpretarse como "*i cree que* φ *es el caso*", pues la creencia no implica verdad.

Otro punto de debate es que los agentes que describen estos lenguajes son "razonadores ideales". Esto se debe a que un agente también conoce las

consecuencias de su conocimiento, es decir, lo que se puede inferir o demostrar a partir de su conocimiento directo.

En computación, es común que se considere a las lógicas en **S5** como lógicas del conocimiento (conocimiento "*ideal*"), y a las lógicas en **KD45** como lógicas de creencias. Además, aquí *creencia* se refiere a que un hecho se cumple en todas las posibilidades epistémicamente indistinguibles para un agente, sin que esto implique que ese hecho sea el caso en realidad. Esta noción de conocimiento es más débil que la que se asume en otros sistemas, como los de revisión de creencias o AGMs (cf. [AGM85]). Aquí, la noción de creencia es más fuerte, pues se consideran las *evidencias* que un agente tiene para creer en algo. En esta tesis se adoptan las convenciones de lógicas **S5** para describir conocimiento y **KD45** para describir creencias, con la noción débil de creencia.

La discusión sobre qué lógicas proveen de un mejor modelo del conocimiento no está libre de controversia. Para una discusión más amplia sobre estos puntos y sus implicaciones, se recomienda al lector la consulta de [Hin62, Sta06, FHMV95].

Por último, basta con mencionar de manera informal que los sistemas de demostración, resultantes de la adaptación de las reglas y axiomas de **KD45** o **S5**, en conjunto con el axioma y la regla:

$$\vdash C_G\varphi \Rightarrow K_i(\varphi \wedge C_G\varphi) \qquad (i \in G) \tag{C1}$$

$$\frac{\vdash \varphi \Rightarrow K_i(\varphi \wedge \psi) \qquad (\text{para cada } i \in G)}{\vdash \varphi \Rightarrow C_G\psi} \tag{RC}$$

son correctos y completos para sus respectivas lógicas.

2.3. Lógica de anuncios públicos

El objetivo principal de esta tesis es definir y presentar un lenguaje dinámico epistémico. Con base en esta meta, resulta útil introducir primero el lenguaje dinámico epistémico más sencillo: los anuncios públicos. La lógica epistémica describe el conocimiento de forma estática, es decir, habla del estado epistémico de un conjunto de agentes en un momento determinado del tiempo. Desde luego, la comunicación afecta el conocimiento, por lo que resulta interesante estudiar sus efectos en el estado epistémico de los comunicantes.

En lugar de estudiar por separado los estados estáticos, se opta por incluir directamente en el lenguaje fórmulas que describen acciones de comunicación y buscar, a nivel semántico, las operaciones sobre los modelos que denoten los efectos de dichas acciones. Esto se logra al combinar la lógica epistémica con los

conceptos de la lógica dinámica, lo cual resulta natural, pues la interpretación de ambas puede darse en base a semánticas de mundos posibles.

Posiblemente el ejemplo más sencillo de esta familia de lenguajes es la lógica de anuncios públicos de [Pla89]. En este lenguaje, la parte dinámica describe una situación donde a todos los agentes del sistema se les anuncia un hecho. Ésto modifica el conocimiento de cada agente sobre el hecho anunciado, pero también sobre el conocimiento que tienen sobre los demás agentes. Primero se presenta la sintaxis del lenguaje.

Definición 2.11 (Sintaxis de $\mathcal{L}_{PAL}$)**.** Sean $\mathcal{P}$ un conjunto infinito numerable de símbolos de proposición y $\mathcal{A}$ un conjunto finito de agentes. El *conjunto $\mathcal{L}_{PAL}$ de las fórmulas de la Lógica de Anuncios Públicos* se define como el menor conjunto tal que:

(a) si $p \in \mathcal{P}$, entonces $p \in \mathcal{L}_{PAL}$;

(b) $\bot \in \mathcal{L}_{PAL}$;

(c) si $\varphi \in \mathcal{L}_{PAL}$, entonces $\neg\varphi \in \mathcal{L}_{PAL}$;

(d) si $\varphi, \psi \in \mathcal{L}_{PAL}$, entonces $(\varphi \vee \psi) \in \mathcal{L}_{PAL}$;

(e) si $\varphi \in \mathcal{L}_{PAL}$ e $i \in \mathcal{A}$, entonces $K_i\varphi \in \mathcal{L}_{PAL}$;

(f) si $\varphi \in \mathcal{L}_{PAL}$ y $G \subseteq \mathcal{A}$, entonces $C_G\varphi \in \mathcal{L}_{PAL}$;

(g) si $\varphi, \psi \in \mathcal{L}_{PAL}$, entonces $[\psi!]\varphi \in \mathcal{L}_{PAL}$. ∎

La sintaxis de $\mathcal{L}_{PAL}$ es similar a la de la lógica epistémica, pero se añaden fórmulas de tipo $[\psi!]\varphi$. La intuición es la siguiente: $\psi!$ es el anuncio público de que ψ es el caso, entonces *después* del anuncio público de ψ, necesariamente se sostiene que φ.

Aquí, el concepto de anuncio público conlleva varias suposiciones:

- Los anuncios son veraces, es decir, sólo se anuncian fórmulas verdaderas;
- Todos los agentes en $\mathcal{A}$ atienden al anuncio exactamente en el mismo instante de tiempo;
- Todos los agentes en $\mathcal{A}$ entienden claramente el anuncio, es decir, para el anuncio $\psi!$, todos los agentes reciben exactamente el mismo mensaje, que ψ es el caso;
- Todos los agentes están conscientes de que el anuncio es público, es decir, todos saben que los demás también atendieron al mismo anuncio.

Dada la intuición básica, es tiempo de presentar la interpretación formal de las fórmulas:

Definición 2.12 (Satisfacción para $\mathcal{L}_{PAL}$)**.** Sean, un modelo:

$$\mathfrak{M} = \langle \mathcal{W}, \{R_i\}_{i\in\mathcal{A}}, V\rangle$$

y un elemento w en $\mathcal{W}$. Se define *la relación* $\models$ *entre modelos apuntados y las fórmulas de* $\mathcal{L}_{PAL}$ como en la definición 2.10, más la siguiente regla:

(g) $\mathfrak{M}, w \models [\psi!]\varphi$ sii $\mathfrak{M}, w \models \psi$ implica que $\mathfrak{M}|_\psi, w \models \varphi$

donde:

$$\mathfrak{M}|_\psi = \langle \mathcal{W}', \{R_i\}_{i\in\mathcal{A}}, V\rangle$$

tal que:

$$\mathcal{W}' = \{w \mid w \in \mathcal{W} \text{ y } \mathfrak{M}, w \models \psi\}$$

∎

En la definición anterior, el elemento destacable es el uso de modelos $\mathfrak{M}|_\psi$, que puede leerse como: "el modelo $\mathfrak{M}$ *relativizado* a ψ". Aquí, $|_\psi$ puede pensarse como una operación sobre modelos, donde el resultado de aplicarla a un modelo $\mathfrak{M}$ es una restricción de éste, en la que se eliminan todas las posibilidades que no satisfacen ψ. Esta noción de operaciones sobre modelos se hará explícita en las siguientes secciones.

De la definición de $|_\psi$, podría pensarse en un axioma como:

$$[\psi!]C_{\mathcal{A}}\psi$$

Sin embargo, la fórmula anterior no es válida, como puede mostrarse con la existencia de anuncios auto refutables. Por ejemplo, el siguiente caso:

$$\psi \stackrel{\text{def}}{=} \neg K_i p \wedge p$$

es claramente un contraejemplo del axioma propuesto, pues ψ se falsifica por su propio anuncio.

Existen, de hecho, sistemas de demostración correctos y completos para la lógica de anuncios públicos. Estos resultan de añadir los siguientes axiomas (llamados axiomas de reducción) a los sistemas de la lógica epistémica:

$$\begin{aligned}
&\vdash [\psi!]p \Leftrightarrow \psi \Rightarrow p \qquad (p \in \mathcal{P})\\
&\vdash [\psi!]\neg\varphi \Leftrightarrow \psi \Rightarrow \neg[\psi!]\varphi\\
&\vdash [\psi!](\varphi \wedge \gamma) \Leftrightarrow [\psi!]\varphi \wedge [\psi!]\gamma\\
&\vdash [\psi!]K_i\varphi \Leftrightarrow K_i[\psi!]\varphi
\end{aligned}$$

2.4. Bisimulación y conocimiento

Un modelo puede estudiarse desde el punto de vista de un sistema de transiciones. Desde esta perspectiva, un mundo posible representa un estado de un sistema, donde se satisfacen ciertas condiciones, y las relaciones de accesibilidad entre mundos posibles representan las posibles transiciones entre los estados del sistema.

De esta forma, dos estados o mundos posibles son equivalentes si, y sólo si, satisfacen las mismas condiciones. En este caso, dichas condiciones son las proposiciones atómicas. Siguiendo de esa manera, puede decirse que dos modelos son equivalentes, o *bisimilares*, si dado un punto de partida, para cada transición posible en el primero, existe otra en el segundo que lleva al sistema a un estado equivalente al que llegó el primero, y la condición anterior se vuelve a cumplir sucesivamente en los siguientes estados.

Definición 2.13 (Bisimilaridad de modelos $\leftrightarroweq$)**.** Sean dos modelos:

$$\mathfrak{M} = \langle \mathcal{W}, \{R_i\}_{i\in\mathcal{A}}, V\rangle$$
$$\mathfrak{M}' = \langle \mathcal{W}', \{R'_i\}_{i\in\mathcal{A}}, V'\rangle$$

y una relación simétrica:

$$S \subseteq \mathcal{W} \times \mathcal{W}'$$

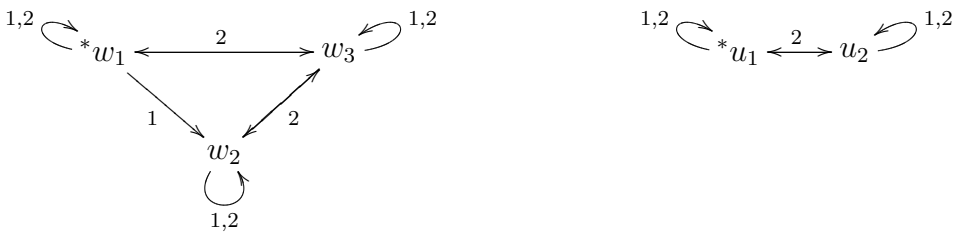
Se dice que S *es una bisimulación entre los modelos* $\mathfrak{M}$ *y* $\mathfrak{M}'$ si se cumple que:

(a) si $(w, w') \in S$, entonces para toda $p \in \mathcal{P}$ se cumple que $V(w, p) = V'(w', p)$ y,

(b) para toda v tal que $(w, v) \in R_i$, existe v' tal que $(w', v') \in R'_i$ y $(v, v') \in S$.

Si S es una bisimulación entre $\mathfrak{M}$ y $\mathfrak{M}'$ y $(w, w') \in S$, entonces se dice que *los modelos apuntados,* $(\mathfrak{M}, w)$ *y* $(\mathfrak{M}', w')$ *son bisimilares, y se escribe* $(\mathfrak{M}, w) \leftrightarroweq (\mathfrak{M}', w')$. Se dice simplemente que $\mathfrak{M}$ *y* $\mathfrak{M}'$ *son bisimilares, y se escribe* $\mathfrak{M} \leftrightarroweq \mathfrak{M}'$, si existe una bisimulación entre ellos. ∎

Por ejemplo, considere los siguientes dos modelos $\mathfrak{M}_1$ (izquierda) y $\mathfrak{M}_2$ (derecha):



Para el modelo $\mathfrak{M}_1$, sólo w_1 y w_2 son mundos p (i.e.., $V(w_1, p) = \mathbf{V}$ y $V(w_3, p) = \mathbf{F}$), mientras que para el modelo $\mathfrak{M}_2$, sólo u_1 es un mundo p. Se apuntan $\mathfrak{M}_1$ y $\mathfrak{M}_2$ por w_1 y u_1, respectivamente (marcados con *). También en el diagrama, una flecha $w \xrightarrow{i} w'$ indica que $(w, w') \in R_i$.

Desde $(\mathfrak{M}_1, w_1)$, se tienen cuatro transiciones posibles, pero sólo la transición $w_1 \xrightarrow{2} w_3$ lleva a un mundo $\neg p$. Lo mismo sucede desde $(\mathfrak{M}_2, u_1)$, pues sólo con una transición $\xrightarrow{2}$ se puede cambiar el sistema a un estado $\neg p$. En ambos modelos, estando en un mundo $\neg p$, sólo con transiciones $\xrightarrow{2}$ se puede mover a un estado p.

El comportamiento en ambos modelos apuntados es exactamente el mismo, por lo que se puede decir que ambos sistemas se *simulan* el uno al otro.

Una consecuencia de la definición de bisimulación es que dados dos modelos bisimilares se tiene que son equivalentes, es decir, modelan al mismo conjunto de fórmulas (teorema 2.14). Por ejemplo, para los modelos anteriores se cumple que:

$$\mathfrak{M}, w_1 \models K_1 p \qquad \mathfrak{M}, w_1 \models \neg K_2 p$$
$$\mathfrak{M}', u_1 \models K_1 p \qquad \mathfrak{M}', u_1 \models \neg K_2 p$$

Más aún, la misma actualización en ambos modelos (i.e., un anuncio público) les causa el mismo efecto:

$$\mathfrak{M}, w_1 \models [p!] C_{1,2} p$$
$$\mathfrak{M}', u_1 \models [p!] C_{1,2} p$$

Teorema 2.14 ($\leftrightarrows$ preserva $\models$)**.** *Sean* $(\mathfrak{M}, w)$ *y* $(\mathfrak{M}', w')$ *dos modelos apuntados tales que* $(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}', w')$*. Entonces,* $\mathfrak{M}, w \models \varphi$*, sii* $\mathfrak{M}', w' \models \varphi$*.*

Demostración. Por inducción estructural en el orden sintáctico de las fórmulas. En la base de la inducción se tiene que las proposiciones atómicas tienen las mismas valuaciones en los dos modelos apuntados, por lo tanto, se satisfacen exactamente las mismas proposiciones atómicas. Si dos fórmulas φ y ψ se satisfacen (o no) en ambos modelos, entonces es directo (por la definición de $\models$) que las fórmulas $\neg\varphi$ y $(\varphi \vee \psi)$ se satisfacen en el primer modelo sii también se satisfacen en el segundo modelo. Para fórmulas de tipo $K_i\varphi$ se tiene que φ es el caso en todo mundo accesible desde w en R_i, entonces cada uno de esos elementos de R_i tiene su símil en R_i'; en el sentido contrario, si en el segundo modelo existiera un mundo accesible desde w' por R_i', donde φ no fuese el caso,

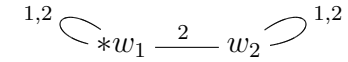
entonces existiría otro equivalente en R_i, pero no es así, pues $\mathfrak{M}, w \models K_i\varphi$. Para fórmulas de tipo $C_G\varphi$ se tiene que φ se satisface en todo mundo v tal que $(w, v) \in (R_G)^+$, esto es, v es accesible desde w en R_G por un camino de n pasos (para toda $n > 0$), donde cada paso corresponde a una pareja $(u_1, u_2) \in R_i$ ($i \in G$); por lo tanto, por la definición de bisimulación, en el segundo modelo debe existir una pareja equivalente $(u_1', u_2') \in R_i'$ para cada paso, y en donde los modelos apuntados por cada elemento del camino satisfacen las mismas fórmulas. Para fórmulas de tipo $[\psi!]\varphi$ se tiene que se eliminan los mundos $\neg\psi$ y en todos los mundos restantes accesibles desde w se satisface φ; como en ambos modelos cada mundo equivalente satisface las mismas fórmulas, al eliminar los mundos $\neg\psi$ en el segundo modelo, en los mundos accesibles desde w' también se deben satisfacer las mismas fórmulas, incluyendo φ. ∎

2.5. Actualización y estructuras de acciones

El lenguaje $\mathcal{L}_{PAL}$ es un buen comienzo para estudiar la "dinamización" de la lógica epistémica. En este contexto, lenguaje dinámico se refiere a que existen acciones (epistémicas) que modifican el conocimiento de los agentes, y dichas acciones y sus consecuencias pueden expresarse en el lenguaje. Por ejemplo, después de un anuncio público, el tamaño de la estructura de Kripke decrece, eliminando el conocimiento (o, más bien, la falta de) inconsistente con el hecho anunciado. Todo esto puede describirse en este lenguaje.

Pero ¿qué pasa con otro tipo de interacción como, por ejemplo, la comunicación privada? Este y otros escenarios de comunicación también pueden modelarse (dinámicamente) utilizando estructuras relacionales y operaciones sobre éstas. En esta sección se ejemplifica primero el modelo de un anuncio privado, después se presenta la definición formal del lenguaje $\mathcal{L}_{LAct}$, y se finaliza con la presentación de un sistema de demostración para el lenguaje.

Sea un modelo $\mathfrak{M}$ con $\mathcal{W} = \{w_1, w_2\}$, $V(w_1, p) = \mathbf{V}$, $V(w_2, p) = \mathbf{F}$ (i.e., w_1 es un "mundo p", en tanto que w_2 es un "mundo $\neg p$"), y las relaciones de accesibilidad para los agentes 1 y 2 como se muestra en el siguiente diagrama (la ausencia de dirección en los arcos asume simetría en la relación de forma implícita):



Tomando en cuenta que w_1 es el *mundo real* (marcado con $*$ en el diagra-

ma), entonces en este escenario se cumplen las siguientes afirmaciones:

$$\mathfrak{M}, w_1 \models K_1 p \tag{2.1}$$
$$\mathfrak{M}, w_1 \models \neg K_2 p \tag{2.2}$$
$$\mathfrak{M}, w_1 \models K_1 \neg K_2 p \tag{2.3}$$

pues p es el caso en el único mundo accesible a 1 (2.1); para 2, w_1 y w_2 (mundos p y $\neg p$, respectivamente) son indistinguibles (2.2); y en el único mundo accesible a 1, $\neg K_2 p$ es el caso (2.3).

Se busca modelar el escenario donde se le anuncia al agente 2 que p es, de hecho, el caso en w_1. Además, se requiere que el anuncio sea completamente privado, es decir, 1 no se "entera" de la actualización al conocimiento de 2.

Como primer acercamiento, puede intentarse eliminar el enlace que relaciona w_1 y w_2 de la relación R_2. Este nuevo modelo, sea $\mathfrak{M}'$, se muestra en el siguiente diagrama:

En $\mathfrak{M}'$, sólo se cumple parte del objetivo original, se actualiza el conocimiento de 2:

$$\mathfrak{M}', w_1 \models K_2 p \tag{2.4}$$

Sin embargo, también se cumple:

$$\mathfrak{M}', w_1 \models K_1 K_2 p \tag{2.5}$$

lo cual no concuerda con la especificación completa, pues si el anuncio es privado a 2, entonces el conocimiento de 1 no debería modificarse (2.3 vs. 2.5).

El problema que surge al modificar únicamente los elementos de R_2 es que toda la estructura también es una referencia para el resto de los agentes. Para modelar un anuncio privado, las posibilidades que consideran el resto de los agentes no deben modificarse.

Cuando una acción tiene lugar (en este caso un anuncio), los agentes pueden no estar conscientes de lo sucedido. Incluso, pueden tener inseguridad sobre si se llevó a cabo una acción u otra distinta, es decir, dos acciones pueden ser indistinguibles para algunos agentes. Entonces, se cuenta con un conjunto de posibles acciones y un conjunto de relaciones de accesibilidad sobre estas acciones para cada agente del sistema. Con estos elementos se puede construir otra estructura de Kripke: una estructura de acciones.

En [BMS99] se propone el producto de dos estructuras relacionales para modelar este tipo de cambios epistémicos. Una estructura mantiene el estado

epistémico de los agentes respecto a los hechos del mundo (proposiciones), mientras que la otra (estructura de acciones) modela el conocimiento de los agentes respecto a la acción ejecutada.

Definición 2.15 (Estructura de acciones)**.** Una *estructura de acciones* $\mathfrak{A}$ es una tupla:

$$\mathfrak{A} = \langle E, \{\rightarrow_i\}_{i\in\mathcal{A}}, \textsc{Pre}\rangle$$

donde E es un conjunto numerable de eventos, se tiene una relación

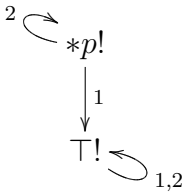
$$\rightarrow_i \subseteq E \times E$$

para cada agente i en $\mathcal{A}$ y

$$\textsc{Pre} : E \rightarrow \Phi$$

es un mapeo de cada evento en E a una precondición representada con una fórmula del lenguaje (en Φ). ∎

Se puede dar una representación gráfica para una estructura de acciones de manera similar que para una estructura de Kripke. Para el ejemplo anterior, en el siguiente diagrama se muestra la estructura $\mathfrak{A}$ que modela el conocimiento de los agentes sobre el anuncio privado "$p!$" a 2:



En el diagrama se puede observar que existen dos posibilidades: el anuncio real "$p!$" o el anuncio trivial "$\top!$" (i.e., el anuncio que no conlleva ningún cambio de conocimiento). Para 2 la única acción posible es el anuncio de p, mientras que, para 1 la única posibilidad es el anuncio trivial y además *cree* que también es la única posibilidad de 2. Esto es, el agente 1 no se percata de la ocurrencia del anuncio "$p!$".

Teniendo ambos modelos, se puede definir una operación que los combine y construya un nuevo modelo como resultado de "ejecutar" el anuncio. La definición formal se da a continuación.

Definición 2.16 (Producto de modelos, $\otimes$)**.** Sean

$$\mathfrak{M} = \langle \mathcal{W}, R(a), V\rangle$$
$$\mathfrak{A} = \langle E, \{\rightarrow_i\}, \textsc{Pre}\rangle$$

un modelo y una estructura de acciones, con w y e dos elementos distinguidos de $\mathfrak{M}$ y $\mathfrak{A}$, respectivamente. Entonces, se define *el producto* $\otimes$*, de los modelos apuntados* $(\mathfrak{M}, w)$ *y* $(\mathfrak{A}, e)$ como:

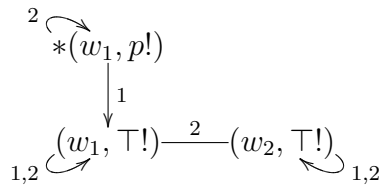
$$(\mathfrak{M}, w) \otimes (\mathfrak{A}, e) = (\mathfrak{M}', (w, e))$$

con $\mathfrak{M}' = \langle \mathcal{W}', R'(a), V' \rangle$ tal que:

- $\mathcal{W}' = \{(w, e) \mid w \in \mathcal{W},\ e \in E \text{ y } \mathfrak{M}, w \models \textsc{Pre}(e)\}$;
- $R'(i) = \{((w, e), (w', e')) \mid (w, w') \in R(i) \text{ y } (e, e') \in \rightarrow_i\}$ para cada $i \in \mathcal{A}$;
- $V'((w, e), p) = V(w, p)$ para cada $(w, e) \in \mathcal{W}'$ y cada $p \in \mathcal{P}$. ∎

La intuición es la siguiente. Se tienen dos conjuntos de posibilidades, uno para los hechos o proposiciones y otro para las acciones. Se crean parejas, mundos posibles contra acciones posibles (producto cartesiano). Cada acción posible está asociada a una precondición. Entonces, para cada pareja (w, e) se valida que en el mundo w, se cumpla la precondición de la acción e. Como las parejas representan un mundo actualizado por la acción asociada las parejas que no cumplen la validación, se consideran "imposibles", y por lo tanto se eliminan. Por último, las relaciones de accesibilidad se combinan módulo bisimulación, de manera que si un agente consideraba posibles tanto un mundo y una acción en los modelos originales, entonces también deberá considerar posible la combinación en el modelo resultante.

Para el ejemplo que se ha manejado, se tiene un nuevo modelo, donde se elimina la única imposibilidad $(w_2, p!)$ (pues w_2 es un mundo $\neg p$). Las relaciones de accesibilidad se muestran en el siguiente diagrama:



El conocimiento de 2 se actualiza (parte superior del diagrama anterior), mientras que 1 cree que no sucedió nada (parte inferior del diagrama). Es decir, el conocimiento de 1 se degrada a creencias.

A continuación se presenta la sintaxis del lenguaje $\mathcal{L}_{LAct}$ y la definición de satisfacción (semántica) del lenguaje.

Definición 2.17 (Sintaxis de $\mathcal{L}_{LAct}$)**.** Sean $\mathcal{P}$ un conjunto numerable de proposiciones atómicas y $\mathcal{A}$ un conjunto finito de agentes. Se define *el conjunto de las fórmulas de* $\mathcal{L}_{LAct}$ como el mínimo conjunto tal que:

(a) si $p \in \mathcal{P}$, entonces $p \in \mathcal{L}_{LAct}$;

(b) $\bot \in \mathcal{L}_{LAct}$ ($\bot \notin \mathcal{P}$);

(c) si $\varphi \in \mathcal{L}_{LAct}$, entonces $\neg\varphi \in \mathcal{L}_{LAct}$;

(d) si $\varphi, \psi \in \mathcal{L}_{LAct}$, entonces $(\varphi \vee \psi) \in \mathcal{L}_{LAct}$;

(e) si $\varphi \in \mathcal{L}_{LAct}$, e $i \in \mathcal{A}$, entonces $K_i\varphi \in \mathcal{L}_{LAct}$;

(f) si $\varphi \in \mathcal{L}_{LAct}$ y $G \subseteq \mathcal{A}$, entonces $C_G\varphi \in \mathcal{L}_{LAct}$;

(g) si $\varphi \in \mathcal{L}_{LAct}$, entonces $[\alpha]\varphi \in \mathcal{L}_{LAct}$.

Donde α denota una estructura de acciones $\mathfrak{A}$ apuntada por una acción $e \in E$. ∎

Definición 2.18 (Satisfacción para $\mathcal{L}_{LAct}$)**.** Sean el modelo y la estructura de acciones siguientes:

$$\mathfrak{M} = \langle \mathcal{W}, \{R_i\}_{i\in\mathcal{A}}, V\rangle$$
$$\mathfrak{A} = \langle E, \{\rightarrow_i\}_{i\in\mathcal{A}}, \textsc{Pre}\rangle$$

Se define *la relación* $\models$ *entre modelos apuntados y las fórmulas de* $\mathcal{L}_{LAct}$ como en la definición 2.10, más la siguiente regla:

(g) $\mathfrak{M}, w \models [\alpha]\varphi$ sii $\mathfrak{M}, w \models \textsc{Pre}(\alpha)$ implica que $\mathfrak{M} \otimes \mathfrak{A}, (w, e) \models \varphi$

donde α denota a la estructura de acciones apuntada $(\mathfrak{A}, e)$ y $\textsc{Pre}(\alpha)$ abrevia $\textsc{Pre}(e)$. ∎

Desde luego, al hablar de operaciones sobre modelos, dichas operaciones deberán respetar la equivalencia de modelos. En otras palabras, si dos modelos contienen la misma información y a ambos se les aplica la misma operación, los dos resultados también deberán contener la misma información. Esta propiedad se expresa en el siguiente teorema.

Teorema 2.19 ($\otimes$ preserva $\leftrightarrows$)**.** *Sean* $(\mathfrak{M}, w)$ *y* $(\mathfrak{M}', w')$ *dos modelos apuntados, y* $(\mathfrak{A}, e)$ *una estructura de acciones apuntada. Si* $(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}', w')$*, entonces* $(\mathfrak{M} \otimes \mathfrak{A}, (w, e)) \leftrightarrows (\mathfrak{M}' \otimes \mathfrak{A}, (w', e))$.

Demostración. Sólo se presenta un bosquejo de la idea principal. Si $(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}', w')$, entonces para algún agente i, a cada elemento (w, u) en R_i le corresponde otro (w', u') en R'_i, tal que toda proposición atómica tiene la misma valuación en u y en u'. Por lo tanto, los nuevos mundos combinados (w, e) y

(w', e) (cualquier e), también son equivalentes, pues se conserva la valuación original. Si f es accesible por i desde e, entonces, y sólo entonces, (u, f) y (u', f) también son accesibles por i en sus modelos resultantes (y equivalentes). Esto es el caso para cada mundo posible de los modelos originales, por lo tanto, ambos resultados contienen la misma información epistémica. ∎

Para finalizar la sección, se presenta a continuación un sistema deductivo correcto y completo para $\mathcal{L}_{LAct}$.

El sistema contiene los siguientes elementos:

(a) Las tautologías de la lógica proposicional;

(b) El axioma **K** adaptado para cada uno de los operadores modales K_i, C_G y $[\alpha]$;

(c) Los siguientes axiomas adicionales:

$$\vdash [\alpha]p \Leftrightarrow (\textsc{Pre}(\alpha) \Rightarrow p) \qquad \text{(Permanencia atómica)}$$
$$\vdash [\alpha]\neg\varphi \Leftrightarrow (\textsc{Pre}(\alpha) \Rightarrow \neg[\alpha]\varphi) \qquad \text{(Funcionalidad parcial)}$$
$$\vdash [\alpha]K_i\varphi \Leftrightarrow (\textsc{Pre}(\alpha) \Rightarrow \bigwedge\{K_i[\beta]\varphi \mid \alpha \rightarrow_i \beta\}) \qquad \text{(Acción)}$$
$$\vdash C_G\varphi \Rightarrow \varphi \wedge \bigwedge\{K_iC_G\varphi \mid i \in G\} \qquad \text{(Mezcla)}$$

(d) Las regla de Modus Ponens y las reglas Generalización adaptadas para los operadores modales K_i, C_G y $[\alpha]$;

(e) La siguiente regla de inducción:

$$\frac{\vdash \varphi \Rightarrow \psi \qquad \vdash \varphi \Rightarrow K_i\varphi \,(\text{para todo } i \in G)}{\vdash \varphi \Rightarrow C_G\psi}$$

(f) La siguiente regla de acción: Sea φ una fórmula, G un conjunto de agentes en $\mathcal{A}$. Sean ψ_β fórmulas para toda β tal que $\alpha \rightarrow_G^* \beta$ ($\rightarrow_G^*$ es la unión de la cerradura transitiva y reflexiva de las relaciones $\rightarrow_i$ para toda i en G) tales que:

- $\vdash \psi_\beta \Rightarrow [\beta]\varphi$
- Si $i \in G$ y $\beta \rightarrow_i \gamma$, entonces $\vdash (\psi_\beta \wedge \textsc{Pre}(\beta)) \Rightarrow K_i\psi_\gamma$

De las suposiciones anteriores, se infiere $\vdash \psi_\alpha \Rightarrow [\alpha]C_G\varphi$.

Capítulo 3

Cálculo π

El cálculo π forma parte de la familia de álgebras de procesos y fue propuesto por Milner et al. como un sucesor de CCS —*Calculus of Communicating Systems* (cf. [Mil80, MPW92]). La característica principal que lo distingue de su antecesor es su capacidad para expresar movilidad.

El lenguaje CCS muestra a los procesos como agentes capaces de comunicarse. Un proceso evoluciona por medio de operaciones internas, pero también comunicándose con otros procesos concurrentes. La comunicación se realiza a través de canales de comunicación, donde un proceso tiene definido qué enlaces o canales tiene a su disposición para comunicarse. En el cálculo π, un proceso es capaz no sólo de intercambiar mensajes con otros procesos concurrentes, sino que también puede intercambiar *enlaces* de comunicación con éstos. Esto es, los procesos también tienen previamente definidos los enlaces a su disposición, pero pueden compartirlos con los demás procesos.

La expresividad del cálculo π le ha otorgado gran aceptación en computación. Sus aplicaciones son diversas. Por ejemplo, ha sido utilizado tanto como herramienta de especificación de protocolos y como fundamento de lenguajes de programación (cf. [Tur96]). Sin embargo, no todas sus aplicaciones radican en las ciencias computacionales, sino que se han extendido a otros ámbitos como la biología (cf. [PRSS01, PCC06]).

La mayor parte de lo que se presenta en este capítulo ya es material es estándar. Puede consultarse [SW01] como una excelente introducción al cálculo π. Aquí se hace una presentación breve del lenguaje, su semántica formal, operacional y denotativa, y se finaliza con la presentación de un subcálculo para modelar comunicación asíncrona. Esta última variante es la que será utilizada en el siguiente capítulo.

3.1. Lenguaje

Definición 3.1 (Sintaxis del cálculo π)**.** Sea $\mathcal{N} = \{x, y, \ldots\}$ un conjunto infinito y numerable de nombres. Entonces, *el conjunto de los procesos del cálculo* π se define como el menor conjunto tal que:

(a) $\mathbf{0}$ es un proceso;

(b) si P es un proceso y π es un prefijo, entonces $\pi.P$ es un proceso;

(c) si $\pi.P$ es un proceso, entonces **if** $x = y$ **then** $\pi.P$ es un proceso;

(d) si P_1 y P_2 son procesos, entonces $P_1 \mid P_2$ es un proceso;

(e) si $\pi_1.P_1$ y $\pi_2.P_2$ son procesos, entonces $\pi_1.P_1 + \pi_2.P_2$ es un proceso;

(f) si P es un proceso, entonces $(\boldsymbol{\nu} z)\ P$ es un proceso (con $\boldsymbol{\nu} \notin \mathcal{N}$);

(g) si $\pi.P$ es un proceso, entonces $!\pi.P$ es un proceso;

(h) si $\pi.P$ es un proceso, entonces $!(\boldsymbol{\nu} z)\ \pi.P$ es un proceso (con $\boldsymbol{\nu} \notin \mathcal{N}$).

donde x, y y z son elementos de $\mathcal{N}$ y los prefijos π, π_1 o π_2 son de la forma $\overline{x}y$ (salida), $x(z)$ (entrada) o τ (acción interna) con $\tau \notin \mathcal{N}$. ∎

Se usará "nombre", "canal" o "mensaje" de manera indistinta para referirse a los elementos de $\mathcal{N}$, siempre y cuando se comprenda del contexto.

El proceso más simple, el proceso nulo o inacción, representa la finalización de un cómputo. A partir de este proceso básico se pueden construir otros más complejos por medio de las operaciones del cálculo.

A los procesos con prefijo se les llama procesos custodiados o resguardados (*guarded processes*). Se dice que un proceso del tipo $\pi.P$ tiene la capacidad de ejecutar la acción descrita por el prefijo π, para después comportarse como P. El prefijo es un guardia o custodio pues el proceso no evolucionará hasta que pueda ejercer la capacidad descrita por el prefijo.

En el proceso $\overline{x}y.P$, el prefijo $\overline{x}y$ indica la posibilidad del proceso de ejecutar una acción de salida. Esto es, el proceso es capaz de enviar el nombre y, a través del canal x, y luego se comporta como P.

El prefijo de entrada indica la acción complementaria a la salida. El proceso $x(z).P$ puede recibir un nombre y, a través de x, y después comportarse como $P\{{}^{y}/_{z}\}$ (i.e., P con la sustitución sintáctica del nombre z por y; posteriormente se definirá formalmente esta operación).

El prefijo τ o acción silenciosa indica una acción interna del proceso. Esto es, $\tau.P$ se convierte en P luego de evolucionar internamente.

Un proceso con el prefijo τ puede ejecutar una acción silenciosa siempre. Sin embargo, los procesos resguardados con prefijos de entrada y salida sólo pueden realizar la acción correspondiente cuando se ejecutan en paralelo con un proceso capaz de realizar la acción complementaria.

En el caso de los procesos paralelos $P_1 \mid P_2$, ambos se ejecutan de forma independiente del otro, excepto en el caso especial ya mencionado. Por ejemplo, en el proceso:

$$\tau.\mathbf{0} \mid \tau.\overline{x}y.\mathbf{0}$$

ambos subprocesos, $\tau.\mathbf{0}$ y $\tau.\overline{x}y.\mathbf{0}$, evolucionan independientemente, para convertirse (en dos pasos computacionales, uno para cada subproceso) en el nuevo proceso:

$$\mathbf{0} \mid \overline{x}y.\mathbf{0}$$

Normalmente, como el subproceso $\mathbf{0}$ no puede evolucionar más, independientemente de cualquier proceso con el que se ejecute en paralelo, se omite.

Otro ejemplo es el proceso:

$$x(z).\overline{w}z.\mathbf{0} \mid \overline{x}y.\mathbf{0}$$

Aquí, los dos subprocesos tienen la capacidad de comunicarse a través del canal x. Entonces, el proceso se convierte (en un sólo paso computacional) en el nuevo proceso:

$$\overline{w}y.\mathbf{0}$$

Aquí se pueden recalcar dos cuestiones. Primero, como se mencionó en el ejemplo anterior, se omite el subproceso $\mathbf{0}$ del lado derecho de $\mid$. Segundo, en el subproceso $\overline{w}y.\mathbf{0}$, se realizó una sustitución sintáctica de las ocurrencias de z por y.

En el cálculo π, los nombres poseen propiedades análogas a las de las variables del cálculo de predicados, esto es, un nombre puede estar *libre* o estar *ligado* a un prefijo. En el ejemplo anterior, en el subproceso original $x(z).\overline{w}z.\mathbf{0}$, la z del prefijo $\overline{w}z$, se refiere a la z recibida por x. Por lo tanto, se dice que el nombre z está ligado (*bound*), mientras que x, y y w son nombres libres.

El proceso $\pi_1.P_1 + \pi_2.P_2$ representa la selección no determinista de los subprocesos $\pi_1.P_1$ y $\pi_2.P_2$. Esto es, el proceso puede evolucionar a cualquiera de los dos subprocesos por medio de una elección no determinista.

El operador de concordancia (*match*) permite ejecutar procesos si al comparar dos nombres, resultan ser el mismo. Por ejemplo, el proceso:

$$(\overline{x}w.\mathbf{0} + \overline{x}u.\mathbf{0}) \mid x(z).\mathbf{if}\ z = u\ \mathbf{then}\ P$$

evolucionará tarde o temprano a P si, y sólo si, la selección no determinista se hace a favor de $\overline{x}u.\mathbf{0}$, pues al comunicarse con el subproceso derecho se realiza la sustitución de la ocurrencia de z por u[1].

El operador $(\boldsymbol{\nu}x)\ P$, restringe el nombre x al proceso P. Por ejemplo, en:

$$x(z).P_1 \mid (\boldsymbol{\nu}x)\ \overline{x}y.P_2$$

los dos subprocesos no pueden comunicarse, pues el uso de $(\boldsymbol{\nu}x)$ indica que el nombre x en $x(z).P_1$ no es el mismo que en $\overline{x}y.P_2$. De hecho, el uso de la letra griega ν se debe a la similitud de su pronunciación con la de palabra inglesa *new*. La restricción $(\boldsymbol{\nu}x)$ puede tratarse como un indicador de que el nombre x es distinto a cualquier nombre que ocurra fuera del alcance del operador.

La replicación de procesos $!\pi.P$ indica que se existe un número no acotado de copias de $\pi.P$ ejecutándose en paralelo [2]. Esto es, la replicación puede tratarse como una solución a la siguiente definición recursiva:

$$!\pi.P \stackrel{\text{def}}{=} \pi.P \mid !\pi.P$$

Es importante destacar que los operadores + y | son asociativos. Como consecuencia, el no determinismo ocurre también de forma implícita en el uso de la composición paralela. Por ejemplo en el proceso:

$$x(z).P_1 \mid \overline{x}w.P_2 \mid \overline{x}u.P_3$$

los dos subprocesos de la derecha pueden comunicarse con $x(z).P_1$, la elección es no determinista, de manera que el proceso anterior puede evolucionar en cualquiera de los siguientes dos procesos en un paso computacional:

$$P_1\{^w/_z\} \mid \overline{x}u.P_3$$

o bien:

$$P_1\{^u/_z\} \mid \overline{x}w.P_2$$

Otra característica interesante del cálculo π, y que lo hace distinto a otros cálculos y álgebras de procesos, es la capacidad de expresar *movilidad*. En el cálculo π, la movilidad es la posibilidad de utilizar un nombre recibido en una acción de entrada como canal de comunicación. Por ejemplo, en el proceso:

$$\overline{x}y.y(z).P_1 \mid x(w).\overline{w}u.P_2$$

[1] Se debe recalcar que en este caso, igual que para los subprocesos de una selección no determinista, P no puede ser un proceso arbitrario, sino uno con prefijo.

[2] Nótese que la replicación está restringida a procesos con prefijo. Puede eliminarse esta restricción, pero no se aumenta la expresividad del cálculo y la semántica requiere más reglas.

el subproceso izquierdo, realmente está enviando un enlace por el cual puede comunicarse posteriormente con el receptor. El proceso evoluciona de la siguiente forma:

$$y(z).P_1 \mid \overline{y}u.P_2$$

y después en:

$$P_1\{^{u}/_{z}\} \mid P_2$$

Desde luego, en el ejemplo anterior, el receptor no tiene garantía de que el mensaje recibido por y realmente provenga del receptor con el que originalmente se comunicó (i.e., nada impide que otro proceso paralelo también utilice y para enviar un mensaje distinto). Esta "falla" puede corregirse fácilmente utilizando la restricción de nombres.

Por ejemplo, si se modifica ligeramente el proceso de ejemplo:

$$((\boldsymbol{\nu} y)\ \overline{x}y.y(z).P_1) \mid x(w).\overline{w}u.P_2$$

el nombre y ahora se encuentra bajo el alcance de una restricción[3], de manera que al enviar el nombre, el alcance de la restricción se extiende hasta abarcar al proceso receptor:

$$((\boldsymbol{\nu} y)\ y(z).P_1 \mid \overline{y}u.P_2)$$

Después de la interacción de los procesos, se garantiza que el uso del nombre y es privativo del emisor y receptor.

3.2. Semántica operacional estructural

La semántica operacional del cálculo π se da en términos de relaciones de transiciones entre procesos. Existen dos variantes principales de estas relaciones, las etiquetadas y las no etiquetadas. Las relaciones de transición no etiquetadas requieren a su vez de otra relación de equivalencia entre procesos para dar una semántica completa del cálculo. La relación de equivalencia adicional se denomina *congruencia estructural* y define algunas propiedades algebraicas de los procesos.

En el presente se muestran las dos alternativas. Primero se muestra, por su simplicidad y por ser más intuitiva, la relación de transición no etiquetada (también llamada relación basada en *reducciones*) y después una variante etiquetada. La diferencia más importante, para fines de este trabajo, es que

[3]El alcance de la restricción está delimitado por un par de paréntesis. Si el alcance no se hace explícito de esta forma, la precedencia del operador de restricción es la más baja de todos los operadores.

el uso de una relación etiquetada permite omitir la congruencia estructural. Adicionalmente, usando relaciones etiquetadas se pueden definir relaciones de equivalencia entre procesos (bisimulaciones) más discriminatorias que con las variantes sin etiquetas.

3.2.1. Por reducciones

Para comenzar se presenta la definición formal de varias nociones descritas intuitivamente en la sección anterior.

Definición 3.2 (Nombres libres y ligados)**.** El conjunto $\texttt{fn}(P)$ *de los nombres libres de un proceso* P, se construye conforme a las siguientes reglas:

(a) $\texttt{fn}(\mathbf{0}) = \varnothing$;

(b) $\texttt{fn}(\overline{x}y.P) = \texttt{fn}(P) \cup \{x, y\}$;

(c) $\texttt{fn}(x(z).P) = (\texttt{fn}(P) \cup \{x\}) - \{z\}$;

(d) $\texttt{fn}(\tau.P) = \texttt{fn}(P)$;

(e) $\texttt{fn}((\boldsymbol{\nu} z)\ P) = \texttt{fn}(P) - \{z\}$;

(f) $\texttt{fn}(\textbf{if}\ x = y\ \textbf{then}\ P) = \texttt{fn}(P)$;

(g) $\texttt{fn}(!P) = \texttt{fn}(P)$;

(h) $\texttt{fn}(P_1 + P_2) = \texttt{fn}(P_1) \cup \texttt{fn}(P_2)$;

(i) $\texttt{fn}(P_1 \mid P_2) = \texttt{fn}(P_1) \cup \texttt{fn}(P_2)$.

El conjunto $\texttt{bn}(P)$ *de los nombres ligados de un proceso* P, se construye conforme a las siguientes reglas:

(a) $\texttt{bn}(\mathbf{0}) = \varnothing$;

(b) $\texttt{bn}(\overline{x}y.P) = \texttt{bn}(P)$;

(c) $\texttt{bn}(x(z).P) = \texttt{bn}(P) \cup \{z\}$;

(d) $\texttt{bn}(\tau.P) = \texttt{bn}(P)$;

(e) $\texttt{bn}((\boldsymbol{\nu} z)\ P) = \texttt{bn}(P) \cup \{z\}$;

(f) $\texttt{bn}(\textbf{if}\ x = y\ \textbf{then}\ P) = \texttt{bn}(P)$;

(g) $\texttt{bn}(!P) = \texttt{bn}(P)$;

(h) $\texttt{bn}(P_1 + P_2) = \texttt{bn}(P_1) \cup \texttt{bn}(P_2)$;

(i) $\texttt{bn}(P_1 \mid P_2) = \texttt{bn}(P_1) \cup \texttt{bn}(P_2)$. ∎

La definición anterior describe cómo determinar si un nombre está ligado para algún proceso dado. Básicamente, un nombre está ligado si es el objeto de un prefijo de entrada o de una restricción, de otro modo está libre.

Otras definiciones relevantes son la de convertibilidad α y sustitución de nombres. La convertibilidad α simplemente formaliza la noción de que, por ejemplo, los siguientes dos procesos representan el mismo cómputo:

$$(\boldsymbol{\nu} y)\ (x(z).\overline{u}z.\mathbf{0} \mid u(s).\overline{s}y.\mathbf{0})$$
$$(\boldsymbol{\nu} z)\ (x(a).\overline{u}a.\mathbf{0} \mid u(b).\overline{b}z.\mathbf{0})$$

Definición 3.3 (Convertibilidad $\alpha, =_\alpha$)**.** *Dos procesos P y Q son α-convertibles o α-equivalentes* ($P =_\alpha Q$), si Q puede obtenerse de P a través de un número finito de cambios en nombres ligados en sus subtérminos, donde:

(a) Si un nombre w no ocurre en un proceso P, entonces el proceso $P\{^w/_x\}$ se obtiene al sustituir por w cada ocurrencia libre de x en P;

(b) Un cambio de nombres ligados en un término $x(z).P$ o $(\boldsymbol{\nu} z)\ P$, se obtiene al reemplazarlos por los términos $x(w).(P\{^w/_z\})$ o $(\boldsymbol{\nu} w)\ (P\{^w/_z\})$, respectivamente. ∎

Definición 3.4 (Congruencia estructural, $\equiv$)**.** *La relación $\equiv$ de congruencia estructural entre procesos* se define como la menor congruencia tal que:

(a) Si $P =_\alpha Q$, entonces $P \equiv Q$;

(b) La composición paralela satisface las siguientes reglas de un monoide abeliano:

$$P \mid Q \equiv Q \mid P$$
$$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$$
$$P \mid \mathbf{0} \equiv P$$

(c) La selección no determinista satisface las siguientes reglas de un monoide abeliano:

$$P + Q \equiv Q + P$$
$$(P + Q) + R \equiv P + (Q + R)$$
$$P + \mathbf{0} \equiv P$$

(d) Se satisfacen las siguientes reglas para la concordancia y replicación:

$$\textbf{if } x = x \textbf{ then } P \equiv P$$
$$!P \equiv P \mid !P$$

(e) Se satisfacen las siguientes reglas de extensión y contracción del alcance de una restricción:

$$(\boldsymbol{\nu} z)\ \mathbf{0} \equiv \mathbf{0}$$
$$(\boldsymbol{\nu} z)\ (P \mid Q) \equiv P \mid ((\boldsymbol{\nu} z)\ Q) \qquad \text{si } z \notin \mathtt{fn}(P)$$
$$(\boldsymbol{\nu} z)\ (P + Q) \equiv P + ((\boldsymbol{\nu} z)\ Q) \qquad \text{si } z \notin \mathtt{fn}(P)$$
$$(\boldsymbol{\nu} z)\ \textbf{if } x = y \textbf{ then } P \equiv \textbf{if } x = y \textbf{ then } (\boldsymbol{\nu} z)\ P \qquad \text{si } z \neq x \text{ y } z \neq y$$
$$(\boldsymbol{\nu} z)\ (\boldsymbol{\nu} s)\ P \equiv (\boldsymbol{\nu} s)\ (\boldsymbol{\nu} z)\ P$$

∎

Con base en las definiciones anteriores, se puede especificar la semántica por reducciones del cálculo π. Esta semántica, simplemente formaliza las nociones descritas en la primera sección de este capítulo.

Definición 3.5 (Relación de reducciones entre procesos, $\longrightarrow$)**.** Se define *la relación de reducción* $\longrightarrow$ *entre procesos del cálculo* π de acuerdo con las reglas en la tabla 3.1. ∎

En este momento es pertinente un ejemplo. Sea el siguiente proceso:

$$P \stackrel{\text{def}}{=} (\boldsymbol{\nu} w)\ ((\overline{x}a.w(z).\mathbf{0} + \overline{x}b.\mathbf{0}) \mid x(z).\overline{w}z.\mathbf{0})$$

En el lado derecho de la composición paralela se tiene que $x(z).\overline{w}z.\mathbf{0} \equiv x(z).\overline{w}z.\mathbf{0} + \mathbf{0}$.

Entonces, por Reduct-Struct y Reduct-Inter:

$$(\overline{x}a.w(z).\mathbf{0} + \overline{x}b.\mathbf{0}) \mid x(z).\overline{w}z.\mathbf{0} \longrightarrow w(z).\mathbf{0} \mid \overline{w}a.\mathbf{0}$$

$$\textsc{Reduct-Struct}\ \frac{P \equiv Q \qquad P' \equiv Q' \qquad P \longrightarrow P'}{Q \longrightarrow Q'}$$

$$\textsc{Reduct-Inter}\ \frac{}{(\overline{x}y.P_1 + Q_1) \mid (x(z).P_2 + Q_2) \longrightarrow P_1 \mid P_2\{^y/_z\}}$$

$$\textsc{Reduct-Tau}\ \frac{}{\tau.P \longrightarrow P}$$

$$\textsc{Reduct-Par}\ \frac{P \longrightarrow P'}{P \mid Q \longrightarrow P' \mid Q}$$

$$\textsc{Reduct-Res}\ \frac{P \longrightarrow P'}{(\boldsymbol{\nu} z)\ P \longrightarrow (\boldsymbol{\nu} z)\ P'}$$

Cuadro 3.1: Reglas de reducción para el cálculo π

Y, por Reduct-Res:

$$P \longrightarrow (\boldsymbol{\nu} w)\ (w(z).\mathbf{0} \mid \overline{w}a.\mathbf{0})$$

Finalmente, aplicando de nuevo las mismas tres reglas:

$$(\boldsymbol{\nu} w)\ (w(z).\mathbf{0} \mid \overline{w}a.\mathbf{0}) \longrightarrow (\boldsymbol{\nu} w)\ (\mathbf{0} \mid \mathbf{0}) \equiv (\boldsymbol{\nu} w)\ \mathbf{0}$$

Desde luego, si se toma en cuenta que $\overline{x}a.w(z).\mathbf{0} + \overline{x}b.\mathbf{0} \equiv \overline{x}b.\mathbf{0} + \overline{x}a.w(z).\mathbf{0}$, entonces, utilizando las mismas reglas, también se tiene la siguiente reducción:

$$P \longrightarrow (\boldsymbol{\nu} w)\ (\overline{x}a.w(z).\mathbf{0} \mid \overline{w}b.\mathbf{0})$$

Y desde ese punto no existen más transiciones posibles.

Como se puede ver por el ejemplo, la semántica por reducciones es muy simple. Sin embargo, se tiene un costo por esa simplicidad. Apoyándose en esta semántica la única manera de comparar dos procesos es por medio del número de reducciones que tiene cada uno. Por supuesto, ésto no es una buena medida de comparación, pues dos procesos con el mismo número de reducciones pueden modelar comportamientos completamente diferentes. Los detalles se dejan para la sección relativa a equivalencia de procesos.

3.2.2. Por transiciones etiquetadas

Como ya se mencionó, existe otra variante de la semántica operacional del cálculo π usando transiciones con etiquetas. Aunque puede seguirse usando

la relación de congruencia en conjunto con las reglas operacionales de esta variante, en ciertas aplicaciones es más conveniente prescindir de ésta. La definición es muy simple. Por ejemplo, en lugar de utilizar una relación externa para definir la operación de concordancia (i.e., **if** $x = x$ **then** $P \equiv P$), ésta se expresa como una regla más del sistema de transiciones. Esto se hace de forma similar para la restricción y la replicación.

A continuación se da la definición formal del sistema de transiciones etiquetado y después se compara con el sistema anterior.

Definición 3.6 (Acciones y nombres libres y ligados de una acción)**.** Se define *el conjunto de acciones Act* como sigue:

$$\begin{aligned} Act = &\{\overline{x}y \mid x, y \in \mathcal{N}\} \\ &\cup \{xy \mid x, y \in \mathcal{N}\} \\ &\cup \{\overline{x}(y) \mid x, y \in \mathcal{N}\} \\ &\cup \{\tau\} \end{aligned}$$

Adicionalmente, se define *el conjunto de nombres ligados de una acción* α como sigue:

$$\mathtt{bn}(\alpha) = \begin{cases} y & \text{si } \alpha = \overline{x}(y) \\ \varnothing & \text{de otra manera} \end{cases}$$

Igualmente, se define *el conjunto de nombres libres de una acción* α tal que, para todo nombre x en α, $x \in \mathtt{fn}(\alpha)$ sii $x \notin \mathtt{bn}(\alpha)$. Se denota con $\mathtt{n}(\alpha)$ al conjunto de todos los nombres que ocurren en α. ∎

Definición 3.7 (Relación etiquetada de transición $\xrightarrow{\alpha}$)**.** *La relación etiquetada de transición* $\xrightarrow{\alpha}$ entre procesos, con α una acción en *Act*, se construye de acuerdo con las reglas de la tabla 3.2. Por claridad y para facilitar la lectura, se omiten de la tabla las reglas Par-R, Comm-R y Close-R, simétricas de Par-L, Comm-L y Close-L, respectivamente. ∎

Con una mirada superficial esta semántica parece igual a la anterior, pero sin la utilización de la relación de congruencia estructural. Sin embargo, no es así. Los prefijos en este sistema representan las acciones que puede ejecutar un proceso, es decir, representan las capacidades del proceso. El poder expresar qué capacidades tiene un proceso, además de cómo evoluciona, permitirá

El proceso $\overline{x}y.P$ es capaz de enviar un mensaje y por el canal x, y se tiene una transición acorde, esto es, $\overline{x}y.P \xrightarrow{\overline{x}y} P$. El proceso $x(z).Q$ puede recibir un mensaje y por el canal x para después comportarse como $Q\{{}^{y}/_{z}\}$, es decir,

$$\textsc{Inp}\ \frac{}{x(z).P \xrightarrow{xy} P\{{}^{y}/_{z}\}} \qquad \textsc{Par-L}\ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}\ \mathtt{bn}(\alpha) \cap \mathtt{fn}(Q) = \varnothing$$

$$\textsc{Out}\ \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \textsc{Comm-L}\ \frac{P \xrightarrow{xy} P' \qquad Q \xrightarrow{\overline{x}y} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\textsc{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P} \qquad \textsc{Sum-L}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

$$\textsc{Match}\ \frac{\pi.P \xrightarrow{\alpha} P'}{\textbf{if}\ x = x\ \textbf{then}\ \pi.P \xrightarrow{\alpha} P'} \qquad \textsc{Open}\ \frac{P \xrightarrow{\bar{x}y} P'}{(\boldsymbol{\nu} y)\ P \xrightarrow{\bar{x}(y)} P'}\ x \neq y$$

$$\textsc{Res}\ \frac{P \xrightarrow{\alpha} P'}{(\boldsymbol{\nu} y)\ P \xrightarrow{\alpha} (\boldsymbol{\nu} y)\ P'}\ y \notin \mathtt{n}(\alpha) \qquad \textsc{Close-L}\ \frac{P \xrightarrow{\bar{x}(y)} P' \qquad Q \xrightarrow{xy} Q'}{P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu} y)\ (P' \mid Q')}$$

$$\textsc{Rep-Act}\ \frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P' \mid !P} \qquad \textsc{Alpha}\ \frac{P =_{\alpha} Q \qquad P' =_{\alpha} Q' \qquad P \xrightarrow{\alpha} P'}{Q \xrightarrow{\alpha} Q'}$$

Cuadro 3.2: Reglas de transición para el Cálculo π

$x(z).Q \xrightarrow{xy} Q\{{}^{y}/_{z}\}$. Igual que en la semántica por reducciones, si los procesos anteriores se ejecutan en paralelo, entonces pueden comunicarse. La diferencia con la semántica por reducciones, es que cada subproceso tiene transiciones independientes. Más aún, $x(z).Q$ puede recibir un mensaje y, pero también puede recibir, por ejemplo, un mensaje a. Estas transiciones hablan de todo lo que es capaz de hacer el proceso, independientemente de los procesos con los que se ejecute en paralelo.

Ya se mencionó la característica llamada movilidad del cálculo π, que permite a dos procesos intercambiar enlaces de comunicación. Estos enlaces también pueden ser privados, es decir, pueden estar dentro del alcance de una restricción. Si un nombre dentro del alcance de una restricción es enviado y recibido por un proceso fuera del alcance de dicha restricción, el alcance se expande hasta abarcar al nuevo proceso. En este caso, $\overline{x}(y)$ es la acción de envío de un nombre privado. Las reglas Open, Close-L y Close-R otorgan las mismas propiedades para la restricción que la relación de congruencia estructural.

3.3. Equivalencia entre procesos

La relación de congruencia estructural puede no ser suficiente como equivalencia para procesos. El objetivo es definir equivalencias de procesos más discriminatorias basadas en el comportamiento de los procesos. La idea principal es que un proceso es capaz de ejecutar un conjunto de secuencias de acciones, y estas secuencias pueden registrarse por un observador externo. Si el conjunto de secuencias de acciones ejecutadas por dos procesos son iguales, entonces dichos procesos son equivalentes. Esta equivalencia se denomina *bisimulación.*

Existen diversas variantes de bisimulación entre procesos, aquí se describirán sólo dos de ellas. La primera es en relación a la semántica de transiciones sin etiquetas. La segunda, más útil, está relacionada con la segunda versión etiquetada.

Definición 3.8 (Bisimulación por reducción)**.** Una relación S entre procesos es una *bisimulación por reducción* si $(P, Q) \in S$ implica que:

(a) si $P \longrightarrow P'$, entonces existe Q' tal que $Q \longrightarrow Q'$ y $(P', Q') \in S$;

(b) si $Q \longrightarrow Q'$, entonces existe P' tal que $P \longrightarrow P'$ y $(P', Q') \in S$.

Si $(P, Q) \in S$ para alguna bisimulación por reducción S, entonces se dice que P y Q son *bisimilares por reducción* y se escribe $P \sim_\tau Q$. ∎

Aunque la definición 3.8 está dada en términos de reducciones, es notable que éstas coinciden con la transición $\xrightarrow{\tau}$, de la semántica de transiciones con etiquetas. Entonces, $\sim_\tau$ considera equivalentes a dos procesos con el mismo número de interacciones internas sin tomar en cuenta el detalle de éstas. Esto aporta poca claridad, pues se relaciona a todos los procesos que no tienen transiciones posibles. Por ejemplo, $\mathbf{0}$ y $(\boldsymbol{\nu} x)\ \overline{x}y$ son semánticamente equivalentes, y por lo tanto también son bisimilares de acuerdo con $\sim_\tau$.

Otro caso es $\mathbf{0} \sim_\tau w(u).\overline{x}w.\mathbf{0}$, pues el segundo proceso tampoco tiene ninguna reducción posible. Sin embargo, si se coloca cada uno de esos procesos en paralelo con $\overline{w}y.x(z).P$, el comportamiento es distinto. Esto es porque a diferencia de $\mathbf{0}$, $w(u).\overline{x}w.\mathbf{0}$ tiene capacidades diferentes (a saber, es capaz de recibir un nombre u por el canal w y después emitir un mensaje w por el canal x, mientras que el proceso $\mathbf{0}$ no tiene ninguna capacidad).

Una mejor opción para comparar procesos es considerar también la estructura de las reducciones, examinando también las posibilidades de cada subproceso por separado. La idea es suponer que existe un observador externo, capaz de interactuar con dos procesos y diferenciarlos por sus capacidades.

La siguiente definición hace uso de la segunda variante de la semántica operacional presentada.

Definición 3.9 (Bisimulación fuerte)**.** Una relación S entre procesos es una *bisimulación fuerte* si $(P, Q) \in S$ implica que:

(a) si $P \xrightarrow{\alpha} P'$, entonces existe Q' tal que $Q \xrightarrow{\alpha} Q'$ y $(P', Q') \in S$;

(b) si $Q \xrightarrow{\alpha} Q'$, entonces existe P' tal que $P \xrightarrow{\alpha} P'$ y $(P', Q') \in S$.

Si $(P, Q) \in S$ para alguna bisimulación fuerte S, entonces se dice que P y Q son *fuertemente bisimilares* y se escribe $P \sim Q$. ∎

Por ejemplo, se tiene que los procesos (asumiendo que $z \notin \texttt{fn}(P)$):

$$\tau.P \sim_\tau \overline{x}y.\mathbf{0} \mid x(z).P$$
$$\tau.P \not\sim \overline{x}y.\mathbf{0} \mid x(z).P$$

Lo anterior es porque aunque los procesos comparten una transición $\xrightarrow{\tau}$, el proceso de la derecha tiene otras capacidades que pudieran ser detectadas por un observador externo (por ejemplo otro proceso $x(z).Q$).

Una consecuencia de la capacidad discriminatoria de la bisimulación fuerte es que es contextual, es decir, $\sim$ se preserva bajo prefijos, composición paralela, replicación y suma.

Si $P \sim Q$, entonces también se cumplen las siguientes afirmaciones:

$$\begin{aligned} \pi.P &\sim \pi.Q \\ P \mid R &\sim Q \mid R \\ !P &\sim !Q \qquad \text{(si son procesos con prefijo)} \\ P + R &\sim Q + R \end{aligned}$$

3.4. Procesos asíncronos

Las dos acciones de comunicación del cálculo π son resguardos. Esto es, un proceso que puede enviar o recibir un mensaje no lo hará hasta que otro proceso pueda realizar la acción complementaria y esperará indefinidamente hasta que esto sea el caso. Esta característica del cálculo π implica que la comunicación entre dos procesos está sincronizada.

El problema es que en muchas aplicaciones la comunicación síncrona no se da de manera natural. Incluso, la implementación resulta más compleja. La

sincronía en la comunicación se da por el hecho de que tanto el envío como la recepción ocurren en el mismo instante. Cuando no existe sincronía, este hecho no es el caso, es decir, un mensaje emitido puede esperar un tiempo no definido en el ambiente hasta ser recibido.

Tomando en cuenta lo anterior, para modelar comunicación asíncrona en el cálculo π, es necesario eliminar la característica de resguardo para los procesos con prefijo de salida. De esta forma, se tiene un subcálculo con una sintaxis ligeramente modificada, capaz de modelar asincronía.

Definición 3.10 (Sintaxis del cálculo π asíncrono)**.** *El conjunto de los procesos del cálculo π asíncrono* se define como el menor conjunto tal que:

(a) $\mathbf{0}$ es un proceso asíncrono;

(b) $\overline{x}y$ es un proceso asíncrono ($x, y \in \mathcal{N}$);

(c) si P es un proceso asíncrono, entonces $x(z).P$ es un proceso asíncrono ($x, z \in \mathcal{N}$);

(d) si P es un proceso asíncrono, entonces $\tau.P$ es un proceso asíncrono ($\tau \notin \mathcal{N}$);

(e) si P_1 y P_2 son procesos asíncronos, entonces $P_1 \mid P_2$ es un proceso asíncrono;

(f) si P es un proceso asíncrono, entonces $(\boldsymbol{\nu} z)\ P$ es un proceso asíncrono ($z \in \mathcal{N}$);

(g) si P es un proceso asíncrono, entonces $!P$ es un proceso asíncrono.

En este subcálculo, se tienen tres modificaciones importantes. Primero, se eliminan los prefijos de salida, dejando a las acciones de salida como procesos independientes. Esta es la diferencia fundamental entre los dos cálculos: un proceso $\overline{x}y$ representa un mensaje que *ya fue enviado* y que se encuentra esperando en el ambiente para ser consumido. A diferencia del cálculo π completo, $\overline{x}y$ ya no representa una capacidad que un proceso *pueda* ejercer, sino una acción ya realizada por sí misma.

La otras diferencias no son tan relevantes. Se elimina la concordancia y la selección no determinista. Lo relevante es la selección no determinista que en realidad no representa una pérdida de expresividad, pues como se discutió en las secciones anteriores, el uso de la composición paralela introduce no determinismo de manera implícita. De hecho, es posible codificar $+$ en este subcálculo (esto es cierto sólo cuando los subprocesos de $+$ son resguardados). Se recomienda la consulta de [Nes00, NP96, Pal97] para mayor detalle en la

comparación del poder expresivo entre los dos cálculos y una discusión más detallada de diferentes codificaciones entre ellos.

Como un subconjunto del cálculo π síncrono, el cálculo π asíncrono hereda algunas de las definiciones de su antecesor, como las de nombres libres y nombres ligados. A continuación se presenta un sistema de transiciones adecuado para el subcálculo.

Definición 3.11 (Relación etiquetada de transición $\xrightarrow{\alpha}$)**.** Sea α una acción en *Act*. Entonces, *la relación etiquetada de transición* $\xrightarrow{\alpha}$ *entre procesos asíncronos*, se construye de acuerdo con las reglas de la tabla 3.3. ∎

$$\textsc{Inp}\ \frac{}{x(z).P \xrightarrow{xy} P\{{}^{y}/_{z}\}} \qquad \textsc{Par-L}\ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}\ \mathtt{bn}(\alpha) \cap \mathtt{fn}(Q) = \varnothing$$

$$\textsc{Out}\ \frac{}{\overline{x}y \xrightarrow{\overline{x}y} \mathbf{0}} \qquad \textsc{Comm-L}\ \frac{P \xrightarrow{xy} P' \qquad Q \xrightarrow{\overline{x}y} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\textsc{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P} \qquad \textsc{Open}\ \frac{P \xrightarrow{\bar{x}y} P'}{(\boldsymbol{\nu} y)\ P \xrightarrow{\bar{x}(y)} P'}\ x \neq y$$

$$\textsc{Res}\ \frac{P \xrightarrow{\alpha} P'}{(\boldsymbol{\nu} y)\ P \xrightarrow{\alpha} (\boldsymbol{\nu} y)\ P'}\ y \notin \mathtt{n}(\alpha) \qquad \textsc{Close-L}\ \frac{P \xrightarrow{\bar{x}(y)} P' \qquad Q \xrightarrow{xy} Q'}{P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu} y)\ (P' \mid Q')}$$

$$\textsc{Rep-Act}\ \frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P' \mid !P} \qquad \textsc{Alpha}\ \frac{P =_{\alpha} Q \qquad P' =_{\alpha} Q' \qquad P \xrightarrow{\alpha} P'}{Q \xrightarrow{\alpha} Q'}$$

Cuadro 3.3: Reglas de transición para el cálculo π asíncrono

Desafortunadamente, no todas las definiciones del cálculo completo pueden reutilizarse. Entre éstas está la definición de bisimulación. Esto se debe a que la noción de observador externo no puede aplicarse de igual forma en el cálculo asíncrono. Si se tiene un proceso asíncrono cualquiera, un observador externo que inserte procesos de salida en el sistema no puede decir si su mensaje ya fue recibido, pues no hay prefijos de salida. Esto implica una redefinición de bisimulación para los procesos asíncronos.

Definición 3.12 (Bisimulación asíncrona fuerte)**.** Una relación simétrica S entre procesos asíncronos es una *bisimulación asíncrona fuerte* si $(P, Q) \in S$ implica que:

(a) si $P \xrightarrow{\alpha} P'$ y $\alpha \neq xy$, entonces existe Q' tal que $Q \xrightarrow{\alpha} Q'$ y $(P', Q') \in S$;

(b) si $P \xrightarrow{xy} P'$, entonces:

(I) existe Q' tal que $Q \xrightarrow{xy} Q'$ y $(P', Q') \in S$ o

(II) existe Q' tal que $Q \xrightarrow{\tau} Q'$ y $(P', (Q' \mid \overline{x}y)) \in S$

Si dos procesos P y Q están relacionados por una bisimulación asíncrona fuerte, entonces se dice que son *fuertemente bisimilares* y se escribe $P \sim_a Q$. ∎

Otra particularidad del subcálculo asíncrono es que muchas de las variantes de bisimulación coinciden. En particular, en este trabajo, el interés es por la versión fuerte. Se da otra definición equivalente (también llamada 1-bisimulación) y ambas se usarán indistintamente.

Definición 3.13 (Bisimulación asíncrona fuerte (alternativa))**.** *La relación* $\sim_{a1}$ *de bisimilaridad asíncrona fuerte* se define como la menor relación simétrica, tal que $P \sim_{a1} Q$ implica que:

(a) Si $P \xrightarrow{\alpha} P'$ y $\alpha \neq xy$, entonces existe Q' tal que $Q \xrightarrow{\alpha} Q'$ y $P' \sim_{a1} Q'$;

(b) Se cumple que $(P \mid \overline{x}y) \sim_{a1} (Q \mid \overline{x}y)$ para todo $\overline{x}y$. ∎

Afortunadamente, la bisimulación asíncrona fuerte conserva algunas de las propiedades de su semejante síncrona. Si $P \sim_a Q$, entonces también se cumplen las siguientes afirmaciones:

$$\pi.P \sim_a \pi.Q$$
$$P \mid R \sim_a Q \mid R$$
$$!P \sim_a !Q \qquad \text{(si son procesos con prefijo)}$$

3.5. Sincronía en el cálculo π asíncrono

En el trabajo de Boudol ([Bou92]) se presentan dos aspectos interesantes del cálculo π asíncrono. El primero es una codificación del cálculo lambda, de la que no se hablará más. El segundo es que es posible simular la comunicación síncrona del cálculo π original, pero en su versión asíncrona. Para este propósito, el autor define un protocolo previo de intercambio de canales privados, después del que se transmite el dato deseado.

El objetivo es que el protocolo simule la comunicación donde las acciones de entrada y de salida sean ambas prefijos de resguardo. En la versión asíncrona

del cálculo, la acción de salida no es una acción de resguardo, pues no hay garantía de que se reciba el mensaje emitido. Sin embargo, para simular un resguardo de salida $\overline{x}y.P$, se puede exigir un acuse por parte del receptor:

$$\overline{x}y \mid u_{(v)}.P$$

mientras que el receptor se codifica de tal forma que cuando recibe el mensaje responda al emisor:

$$x_{(z)}.(\overline{u}v \mid Q)$$

donde P y Q son el resto de la ejecución del emisor y receptor, respectivamente. Además, se asume que los nombres u y v no ocurren libres en ninguno de estos dos procesos.

En este punto el protocolo aún no es correcto, pues nada impide que el mensaje emitido sea capturado por otro proceso y que, además, otro proceso distinto use el canal de sincronización u para emitir un mensaje que no sea el acuse. Esto puede corregirse encerrando a los dos procesos dentro de una restricción $_{(}\boldsymbol{\nu} u_{)}$:

$$_{(}\boldsymbol{\nu} u_{)}\ ((\overline{x}y \mid u_{(v)}.P) \mid x_{(z)}.(\overline{u}v \mid Q))$$

Sin embargo, interesa que aunque los procesos sean independientes de una restricción común, aun puedan sincronizarse y establecer comunicación. Esto se hace por medio de un intercambio previo de canales privados, donde el emisor primero envía su canal privado. El receptor utiliza el canal del emisor para enviarle ahora su propio canal privado. Por último, el emisor envía el mensaje por el canal privado del receptor asegurándose que sólo éste lo recibirá:

$$E \stackrel{\text{def}}{=} {}_{(}\boldsymbol{\nu} u_{)}\ (\overline{x}u \mid u_{(v)}.(\overline{v}y \mid P))$$
$$R \stackrel{\text{def}}{=} {}_{(}\boldsymbol{\nu} v_{)}\ (x_{(u)}.(\overline{u}v \mid v_{(z)}.Q))$$

La ejecución paralela de estos procesos ocurre como sigue:

$$\begin{aligned}(E \mid R) &\xrightarrow{\tau} {}_{(}\boldsymbol{\nu} u_{)}\ (u_{(v)}.(\overline{v}y \mid P) \mid {}_{(}\boldsymbol{\nu} v_{)}\ (\overline{u}v \mid v_{(z)}.Q)) \\ &\xrightarrow{\tau} {}_{(}\boldsymbol{\nu} u_{)}\ {}_{(}\boldsymbol{\nu} v_{)}\ (\overline{v}y \mid P \mid v_{(z)}.Q) \\ &\xrightarrow{\tau} {}_{(}\boldsymbol{\nu} u_{)}\ {}_{(}\boldsymbol{\nu} v_{)}\ (P \mid Q\{{}^{y}/_{z}\})\end{aligned}$$

En un inicio, el emisor no puede continuar su ejecución hasta no recibir un mensaje por el canal privado u; cuando el receptor recibe este canal, se extiende el alcance de $_{(}\boldsymbol{\nu} u_{)}$ a los dos procesos. Entonces, el receptor envía su canal privado, extendiendo el alcance de $_{(}\boldsymbol{\nu} v_{)}$, y permitiendo que, finalmente, el mensaje y se envíe y se reciba por el canal del receptor y los procesos continúan su ejecución normal.

Capítulo 4

Acciones epistémicas del Cálculo π asíncrono

Este y el siguiente capítulo incluyen la aportación original de la tesis. Se define un lenguaje dinámico epistémico cuyas acciones son procesos del cálculo π asíncrono. Existe una gama amplia de lenguajes dinámicos epistémicos que cubren variados escenarios de comunicación, con aplicación directa en computación como, por ejemplo, el análisis de sistemas distribuidos (cf. [HM84]). También el cálculo π es usado con esos propósitos, pues el modelado de sistemas de ese tipo resulta natural en un álgebra de procesos. El objetivo, es que una vez que se obtiene una descripción formal de un sistema en el cálculo π asíncrono, se puede proceder directamente al análisis de diversas propiedades de corrección, seguridad, etc., ligadas a la comunicación.

El trabajo se realiza en dos partes. En este capítulo se define la sintaxis del lenguaje, se especifica su semántica y se termina con una discusión acerca de algunas propiedades deseables para el lenguaje, pero que no se cumplen. En el siguiente capítulo se revisarán algunas modificaciones para solucionar el problema.

4.1. El lenguaje $\mathcal{L}_{EA\pi}$

La sintaxis del lenguaje incluye dos subconjuntos, uno para la parte lógica y otro para la parte dinámica. La parte lógica no es diferente a la de los lenguajes descritos en capítulos anteriores. La parte dinámica es una pequeña modificación de la sintaxis del cálculo π asíncrono.

Las modificaciones a la sintaxis original del cálculo π tienen como fin añadirle capacidad para expresar detalladamente las acciones atómicas rea-

lizadas por cada individuo de un conjunto de agentes. Se utiliza el cálculo π, pues es un lenguaje rico que permite expresar ese detalle en la estructura que tiene internamente la interacción de dos procesos.

Definición 4.1 (Sintaxis de las fórmulas de $\mathcal{L}_{EA\pi}$)**.** Sean

$$\mathcal{P} = \{p, q, \ldots\}$$

un conjunto infinito numerable de proposiciones atómicas,

$$\mathcal{A} = \{1, \ldots, n\}$$

un conjunto finito de agentes y Π el conjunto de los procesos asíncronos de la definición 4.2. Entonces, se define *el conjunto* Φ *de las fórmulas de* $\mathcal{L}_{EA\pi}$ como el menor conjunto tal que:

(a) $\bot \in \Phi$;

(b) si $p \in \mathcal{P}$, entonces $p \in \Phi$;

(c) si $\varphi \in \Phi$, entonces $\neg\varphi \in \Phi$;

(d) si $\varphi, \psi \in \Phi$, entonces $\varphi \vee \psi \in \Phi$;

(e) si $\varphi \in \Phi$ e $i \in \mathcal{A}$, entonces $K_i\varphi \in \Phi$;

(f) si $\varphi \in \Phi$ y $G \subseteq \mathcal{A}$, entonces $C_G\varphi \in \Phi$;

(g) si $\varphi \in \Phi$ y $P \in \Pi$, entonces $[P]\varphi \in \Phi$. ∎

Definición 4.2 (Sintaxis de los procesos de $\mathcal{L}_{EA\pi}$)**.** Sea

$$\mathcal{N} = \{x, y, \ldots\}$$

un conjunto infinito numerable de nombres. Entonces, *el conjunto* Π *de los procesos asíncronos de* $\mathcal{L}_{EA\pi}$ se define como el menor conjunto tal que:

(a) $\mathbf{0} \in \Pi$;

(b) si $x, y \in \mathcal{N}$ e $i \in \mathcal{A}$, entonces $\overline{x}_i y \in \Pi$;

(c) si $x \in \mathcal{N}$, $i \in \mathcal{A}$ y $\varphi \in \Phi$, entonces $\overline{x}_i\varphi \in \Pi$;

(d) si $x, z \in \mathcal{N}$, $i \in \mathcal{A}$ y $P \in \Pi$, entonces $x_{i(z)}.P \in \Pi$;

(e) si $P \in \Pi$, entonces $\tau.P \in \Pi$;

(f) si $P, Q \in \Pi$, entonces $P \mid Q \in \Pi$;

(g) si $z \in \mathcal{N}$ y $P \in \Pi$, entonces $(\boldsymbol{\nu} z)\, P \in \Pi$;

(h) si $\overline{x}_i\eta \in \Pi$, entonces $!\overline{x}_i\eta \in \Pi$;

(i) si $\pi.P \in \Pi$, entonces $!\pi.P \in \Pi$. ∎

Hay dos diferencias con el cálculo π asíncrono original. Primero, en las acciones de entrada y salida se especifica el agente que las realiza y, segundo, se contemplan acciones del tipo $\overline{x}_i\varphi$, donde φ representa una fórmula y no un nombre. Como convención, para referirse a proposiciones atómicas, se usarán las letras p, q, r, etc., así como, las letras x, y, w, etc., para referirse a nombres en $\mathcal{N}$. También, se usará la variable η, en acciones del tipo $\overline{x}_i\eta$, para indicar que se trata de un nombre o una fórmula arbitraria.

Un proceso $\overline{x}_i\eta$ representan un mensaje asíncrono η, enviado a través del canal x, sólo que aquí se indica el agente responsable de esa acción, esto es, el agente i. El proceso $x_{i(z)}.P$ también representa la disposición para recibir un mensaje por medio del canal x, aquí también se indica el agente receptor. El resto de las construcciones de los procesos es la misma que la presentada en el capítulo anterior, con la excepción de que la replicación se restringe a procesos custodiados y a mensajes asíncronos individuales.

Este lenguaje hereda el no determinismo de su antecesor y además introduce la comunicación asíncrona en esta familia de lógicas. Por ejemplo, para el proceso

$$\overline{x}_1 p \mid \overline{x}_2 q \mid x_{3(z)}.\mathbf{0} \tag{4.1}$$

se tiene que el agente 3 tiene la posibilidad de recibir uno de los dos mensajes disponibles en el medio. Entonces, después de la ejecución de este proceso, o bien $K_3 p$, o bien $K_3 q$ (asumiendo que, en un inicio, $\neg K_3 p \wedge \neg K_3 q$). Más aún, los procesos de salida a la izquierda de (4.1) representan mensajes que ya se enviaron, de manera que cuando un 1 y 2 colocan sus mensajes en el medio, no tienen la certeza ni de que se reciba, ni de quién lo reciba, pues cualquier agente que tenga un proceso paralelo escuchando por el canal x puede consumir el mensaje.

El resto del capítulo se dedicará a especificar la semántica formal para el lenguaje. La semántica propuesta contempla dos aspectos: primero, los efectos epistémicos de la comunicación basada en envío de mensajes asíncronos y, segundo, el no determinismo que surge implícitamente de las acción del cálculo π asíncrono. Para el primer aspecto se utilizan las herramientas de la lógica de Baltag [BMS99], y para el segundo se utiliza la máquina abstracta basada en

listas en [PC04]. En este punto, es pertinente aclarar que el objetivo del trabajo no es el de especificar una semántica denotativa del cálculo π asíncrono. Para eso se remite al lector al trabajo de Stark [Sta96] y al de Aziz y Hamilton [AH01].

4.2. Semántica formal de $\mathcal{L}_{EA\pi}$

La interpretación de las fórmulas es estándar, esto es, se utilizan marcos de Kripke como modelos de éstas. Un proceso del lenguaje define un conjunto de posibles operaciones a realizar sobre los modelos. Estas operaciones son la aplicación de una secuencia de productos de estructuras de acciones, que determinan los cambios epistémicos derivados de la ejecución de los procesos.

4.2.1. Parte estática

A continuación se presentan las definiciones básicas para la interpretación de la parte estática del lenguaje.

Definición 4.3 (Modelos para $\mathcal{L}_{EA\pi}$)**.** *Un modelo $\mathfrak{M}$ para las fórmulas en Φ,* se define como la tupla:

$$\mathfrak{M} = \langle \mathcal{W}, \{R_i\}_{i\in\mathcal{A}}, V\rangle$$

donde:

- $\mathcal{W} = \{w_1, w_2, \ldots\}$ es un conjunto numerable de mundos posibles;
- $R_i \subseteq \mathcal{W}\times\mathcal{W}$ es una relación de accesibilidad entre mundos posibles para cada agente $i \in \mathcal{A}$;
- $V : \mathcal{W}\times\mathcal{P} \to \{\mathrm{V}, \mathrm{F}\}$ es una función de valuación de las proposiciones atómicas en cada mundo posible;

Dado un modelo y un elemento distinguido w de $\mathcal{W}$, se dice que *la pareja* $(\mathfrak{M}, w)$ *es un modelo apuntado por* w, algunas veces se omiten los paréntesis. ∎

Definición 4.4 (Satisfacción en $\mathcal{L}_{EA\pi}$ —parte estática—)**.** Dados un modelo $\mathfrak{M} = \langle \mathcal{W}, \{R_i\}_{i\in\mathcal{A}}, V\rangle$ y un elemento distinguido $w \in \mathcal{W}$, se define *la relación* $\models$ *entre modelos apuntados y las fórmulas (estáticas) en* Φ de acuerdo con las siguientes reglas:

(a) $\mathfrak{M}, w \not\models \bot$;

(b) $\mathfrak{M}, w \models p$ sii $V(w, p) = \mathrm{V}$;

(c) $\mathfrak{M}, w \models \neg\varphi$ sii $\mathfrak{M}, w \not\models \varphi$;

(d) $\mathfrak{M}, w \models (\varphi \vee \psi)$ sii $\mathfrak{M}, w \models \varphi$ ó $\mathfrak{M}, w \models \psi$;

(e) $\mathfrak{M}, w \models K_i\varphi$ sii para todo $w' \in \mathcal{W}$, si $(w, w') \in R_i$, entonces $\mathfrak{M}, w' \models \varphi$;

(f) $\mathfrak{M}, w \models C_G\varphi$ sii para todo $w' \in \mathcal{W}$, si $(w, w') \in (R_G)^+$, entonces $\mathfrak{M}, w' \models \varphi$. ∎

La relación de satisfacción es estándar para todas las fórmulas y ya se discutieron los detalles en el capítulo 2. La aportación de este trabajo está en la interpretación de los procesos como acciones epistémicas de la siguiente sección.

4.2.2. Parte dinámica

En el caso de fórmulas como $[P]\varphi$, se tiene que después de la ejecución del proceso P pueden ocurrir diferentes intercambios de mensajes y que, a su vez, pueden generar diferentes actualizaciones en el modelo.

Por ejemplo, el proceso:

$$P \stackrel{\text{def}}{=} \overline{x}_1\varphi \mid \overline{x}_1\psi \mid x_{2(z)}.\overline{y}_2 z \tag{4.2}$$

tiene, entre otras, las siguientes dos transiciones posibles:

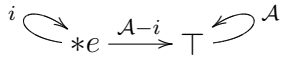
$$P \xrightarrow{\tau} \overline{x}_1\varphi \mid \overline{y}_2\psi \tag{4.3}$$

$$P \xrightarrow{\tau} \overline{x}_1\psi \mid \overline{y}_2\varphi \tag{4.4}$$

En este caso se tiene que el agente 2 pudo enterarse que φ o ψ. Sin embargo, el agente 1 no puede decir cuál de los dos mensajes emitidos se recibió, pues la comunicación carece de sincronía. Incluso, el emisor de un mensaje asíncrono no puede decir si su mensaje será recibido del todo. De la misma manera, el subproceso $x_{2(z)}.\overline{y}_2 z$ indica que el agente 2 está esperando recibir un mensaje disponible por el canal x.

De lo anterior, siguiendo el sistema de transiciones para el cálculo π asíncrono, se tiene que solamente las transiciones $\xrightarrow{\tau}$ son las únicas que pueden (en algunos casos) afectar el estado epistémico de los agentes. Estos cambios se expresan utilizando estructuras de acciones (ver capítulo 2).

Las estructura de acción que representa la recepción de un mensaje asíncrono φ es la siguiente:



donde i es el receptor del mensaje y e es la acción ejecutada con la precondición:

$$\textsc{Pre}(e) = \varphi \wedge K_j\varphi$$

para un emisor j.

La estructura anterior se interpreta de la siguiente forma: el agente i aprende que $K_j\varphi$, mientras que los demás agentes no se enteran del cambio ($\top$ es el anuncio trivial). También la precondición implica que los agentes son veraces, y que sólo comunican lo que realmente conocen, o mejor dicho, creen.

Para la comunicación asíncrona de este lenguaje sólo hacen falta las instancias de estructuras de acciones de este tipo. Esto se captura en la siguiente definición.

Definición 4.5 (Estructura de acciones $\mathfrak{A}^{\varphi}_{i,j}$ para $\mathcal{L}_{EA\pi}$)**.** Dados dos agentes en $\mathcal{A}$, un emisor i y un receptor j, de un mensaje asíncrono $\overline{x}_i\varphi$ (con φ una fórmula). Entonces, se define *una estructura de acción* $\mathfrak{A}^{\varphi}_{i,j}$ como la tupla:

$$\mathfrak{A}^{\varphi}_{i,j} = \langle E, \{\rightarrow_n\}_{n\in\mathcal{A}}, \textsc{Pre}\rangle$$

donde:

- $E = \{e, \top\}$;
- $\rightarrow_j = \{(e,e), (\top,\top)\}$;
- $\rightarrow_n = \{(e,\top), (\top,\top)\}$ (para todo $n \in \mathcal{A} - \{j\}$);
- $\textsc{Pre}(e) = \varphi \wedge K_i\varphi$;
- $\textsc{Pre}(\top) = \top$. ∎

Por ejemplo, para el proceso (4.2) se tiene que la fórmula:

$$[\overline{x}_1\varphi \mid \overline{x}_1\psi \mid x_{2(z)}.\overline{y}_2 z]\chi$$

se satisface en $(\mathfrak{M}, w)$ si, y sólo sí, χ se satisface en todos los posibles modelos actualizados, uno donde 2 aprende φ y otro donde aprende ψ. Esto es, se cumplen:

$$(\mathfrak{M}, w) \otimes (\mathfrak{A}^{\varphi}_{1,2}, e) \models \chi \tag{4.5}$$

$$(\mathfrak{M}, w) \otimes (\mathfrak{A}^{\psi}_{1,2}, e) \models \chi \tag{4.6}$$

Nótese que la aplicación del producto con estas estructuras resulta en modelos **KD45**, por lo que $K_i\varphi$ no necesariamente implica φ, es decir, se trata de creencias no de conocimiento como tal. Que la precondición de la acción realizada en la estructura $\mathfrak{A}^{\varphi}_{i,j}$ sea $\varphi \wedge K_i\varphi$, quiere decir que el receptor j confía en la veracidad del emisor, y acutaliza sus creencias acorde.

Entonces, la interpretación de un proceso es un conjunto de funciones que al aplicarse al modelo actual, resultan en un nuevo modelo actualizado con los cambios necesarios.

Con la finalidad de construir mecánicamente la interpretación de los procesos, se define una transformación que realiza dos cosas: primero, elimina las restricciones de nombres $(\boldsymbol{\nu}x)$, sustituyendo el objeto de dichas restricciones por nombres nuevos y únicos y, segundo, ya con todos los nombres libres, se insertan todos los procesos paralelos en un multiconjunto (o lista, desde el punto de vista de una implementación en computadora) que se utiliza para calcular todas las reducciones posibles.

La eliminación de restricciones se basa en el siguiente hecho (de la congruencia estructural):

$$P \mid (\boldsymbol{\nu}z)\, Q) \equiv (\boldsymbol{\nu}z)\, P \mid Q) \qquad \text{si } z \notin \mathtt{fn}(P)$$

Esto se puede hacer de forma completamente mecánica, generando nombres únicos, de manera que ya no sea necesaria la restricción.

La idea es generar un ambiente en donde residen todos los procesos paralelos en donde todos los nombres que pueden interactuar sean libres. Por ejemplo, el siguiente proceso:

$$\overline{x}_1\varphi \mid (\boldsymbol{\nu}x)\, \overline{x}_2\psi \mid x_3(z).\mathbf{0})$$

se convierte en un ambiente ρ:

$$\rho = \{\!|\overline{x}_1\varphi, \overline{x}^1_2\psi, x^1_3(z).\mathbf{0}|\!\}$$

donde $x^1 \neq x$ es un nombre nuevo y distinto a cualquier otro nombre libre que ocurra en el proceso original.

Convención 4.6. Es importante aclarar un punto aquí. En los procesos $\overline{x}_i\varphi$ y $x_j(z).P$, los subíndices i y j indican el agente que realiza la acción, sin embargo, x es el mismo nombre en ambos y si los procesos se componen en paralelo pueden interactuar. En lo que resta de la tesis, se toma la convención de que *nombres con superíndices distintos representan nombres diferentes*. Por ejemplo, los siguientes procesos concurrentes $\overline{x}^1_i\varphi \mid x^2_j(z).P$ no pueden interactuar, pues los nombres x^1 y x^2 son distintos. ∎

Para llevar a cabo lo anterior, se define una operación:

$$P \oplus (\rho, s)$$

que inserta un proceso cualquiera P, en un ambiente ρ, donde s es el conjunto de los nombres libres de todos los procesos que ocurren en ρ y los de P.

Definición 4.7 (Operación $\oplus$)**.** Dados un proceso P, un multiconjunto ρ, cuyos elementos son procesos en Π, y un conjunto $s \subseteq \mathcal{N}$ tal que contiene los nombres libres de todos los procesos en ρ y los de P, se define *la operación* $\otimes$ *que inserta el proceso* P *en el ambiente* ρ de acuerdo a las siguientes reglas:

$$\begin{aligned}
\mathbf{0} \oplus (\rho, s) &= (\rho, s) \\
\overline{x}_i\eta \oplus (\rho, s) &= (\rho \uplus \{\!|\overline{x}_i\eta|\!\}, s) \\
x_i(z).P \oplus (\rho, s) &= (\rho \uplus \{\!|x_i(z).P|\!\}, s) \\
(P \mid Q) \oplus (\rho, s) &= P \oplus Q \oplus (\rho, s) \\
!\overline{x}_i\eta \oplus (\rho, s) &= (\rho \uplus \{\!|!\overline{x}_i\eta|\!\}, s) \\
!\pi.P \oplus (\rho, s) &= (\rho \uplus \{\!|\pi.(P \mid !\pi.P)|\!\}, s) \\
(\boldsymbol{\nu} z)\ P \oplus (\rho, s) &= P\{^{z^n}/_z\} \oplus (\rho, s \cup \{z^n\})
\end{aligned}$$

Donde:

$$n = min\{n \mid n \in \mathbb{N} \wedge z^{n-1} \in s\}$$

y $\uplus$ es la unión estándar de multiconjuntos. ∎

Otro paso previo a la interpretación es la definición de las funciones de actualización de modelos. Como ya se mencionó, de la ejecución de un proceso resulta un conjunto de funciones posibles a aplicar a un modelo. Dichas funciones son la composición de varias actualizaciones hechas por la interacción de los procesos paralelos de un ambiente. Estas funciones se representan como abstracciones lambda, definidas a continuación.

Definición 4.8 (Funciones de actualización)**.** Se definen las siguientes *funciones de actualización de modelos (apuntados)*:

$$id \stackrel{\text{def}}{=} \lambda M.M$$

$$tau \stackrel{\text{def}}{=} \lambda M.M$$

$$com(i,j,\eta) \stackrel{\text{def}}{=} \begin{cases} tau & \text{si } \eta \in \mathcal{N} \\ \lambda M.(\langle\{\},\{\},\{\}\rangle, w) & \text{si } \mathfrak{M}, w \not\models K_i\eta \\ \lambda M.(\mathfrak{A}^{\eta}_{i,j}, e) \otimes M & \text{en otro caso} \end{cases}$$

Se utiliza $\overrightarrow{\mathbf{K}}$ para denotar a la clase de todos los modelos apuntados de $\mathcal{L}_{EA\pi}$. El tipo de todas estas funciones es $\overrightarrow{\mathbf{K}} \to \overrightarrow{\mathbf{K}}$. ∎

Cabe aclarar que en el segundo caso de la función *com*, $\langle\{\}, \{\}, \{\}\rangle$ se refiere al modelo vacío que satisface, por vacuidad, cualquier fórmula. El razonamiento es el siguiente. Este lenguaje asume que los agentes son veraces, y que sólo comunican lo que realmente conocen o creen. Contemplar escenarios de comunicación donde los agentes mienten o tratan de engañar es más complejo y está fuera del alcance de este trabajo. Por lo anterior, si un agente emite un mensaje falso, se aplica la regla típica de falsedad en lógica, esto es, de falso lo que sea.

Entonces, dado un proceso cualquiera, una posible ejecución se traduce en la aplicación de una función construida como la composición secuencial de una o varias de las funciones anteriores. Para esto, se define una relación de interpretación:

$$[\![(\rho, s)]\!]_{\mathfrak{M},w}\phi$$

donde ρ es un ambiente de procesos en paralelo con nombres libres en s y el parámetro ϕ es una función de actualización o composión de éstas que representa los cambios que se han hecho al modelo hasta el momento.

Primero se presenta la definición formal de la interpretación y luego se muestra un ejemplo.

Definición 4.9 (Interpretación de procesos $[\![\cdot]\!]$)**.** Sea ρ un multiconjunto cuyos elementos están en Π y $s \subseteq \mathcal{N}$ tal que, para todo $P \in \rho$, se tiene que $\mathtt{fn}(P) \subseteq s$. Sea ϕ una función cuyo rango y dominio son los modelos apuntados de $\mathcal{L}_{EA\pi}$. Sea $\mathfrak{M}, w$ un modelo apuntado de $\mathcal{L}_{EA\pi}$. Entonces, se define la *relación de interpretación*

$$[\![(\rho, s)]\!]_{\mathfrak{M},w}\phi \subseteq \{f \mid f : \overrightarrow{\mathbf{K}} \to \overrightarrow{\mathbf{K}}\}$$

conforme a las siguientes reglas:

$$[\![(\rho, s)]\!]_{\mathfrak{M},w}\phi = \{\phi\} \tag{I1}$$

Si no existe una pareja de elementos $\{\!|\overline{x}_i\eta, x_j(z).P|\!\} \subseteq \rho$ o una pareja $\{\!|!\overline{x}_i\eta, x_j(z).P|\!\} \subseteq \rho$.

En caso contrario, la relación se define como sigue:

$$[\![(\rho, s)]\!]_{\mathfrak{M},w}\phi = A_1 \cup A_2 \tag{I2}$$

Donde:

$$A_1 \stackrel{\text{def}}{=} \bigcup_{\overline{x}_i\eta \in \rho} \bigcup_{x_j(z).P \in \rho} [\![P\{^{\eta}/_{z}\} \oplus (\rho', s)]\!]_{\mathfrak{M},w}(com(i, j, \eta) \circ \phi) \tag{I2.1}$$

con $\rho' = \rho - \{\!|\overline{x}_i\eta, x_j(z).P|\!\}$ y

$$A_2 \stackrel{\text{def}}{=} \bigcup_{!\overline{x}_i\eta\in\rho} \bigcup_{x_j(z).P\in\rho} [\![P\{^{\eta}/_{z}\} \oplus (\rho', s)]\!]_{\mathfrak{M},w}(com(i,j,\eta)\circ\phi) \tag{I2.2}$$

con $\rho' = \rho - \{\!|x_j(z).P|\!\}$. ∎

Hasta este punto ya se ha alcanzado un buen grado de complejidad, por lo que es conveniente presentar el ejemplo prometido.

Sea $(\mathfrak{M}, w)$ un modelo apuntado y el proceso siguiente con sus nombres libres:

$$\begin{aligned} P &\stackrel{\text{def}}{=} \overline{x}_1\varphi \mid x_2(z).\overline{y}_2 z \mid y_3(z).\mathbf{0} \mid y_4(z).\mathbf{0} \\ s &= \mathtt{fn}(P) = \{x, y\} \end{aligned}$$

Primero, es necesario transformar el proceso en un ambiente ρ, esto se hace insertándolo en un ambiente inicial vacío, esto es:

$$P \oplus (\{\!|\}\!|, s) = (\rho, s)$$

donde:

$$\rho = \{\!|\overline{x}_1\varphi, x_2(z).\overline{y}_2 z, y_3(z).\mathbf{0}, y_4(z).\mathbf{0}|\!\}$$

Una vez transformado el proceso, se procede a construir su interpretación, esto es, se calcula:

$$[\![(\rho, s)]\!]_{\mathfrak{M},w}(id)$$

Aquí, se utiliza la función id como parámetro inicial. La idea es que un proceso que no tiene reducciones posibles, por la definición (I1) la función que actualiza el modelo original es en realidad la función identidad, es decir, no se produce ningún cambio. Si el proceso puede reducirse, entonces las funciones resultantes se van a ir componiendo secuencialmente con la identidad como base.

Para este caso, se tiene que inicialmente, sólo existe una reducción posible (i.e., una pareja de procesos que pueden comunicarse), por lo que por (I2.1) se tiene lo siguiente:

$$[\![(\rho, s)]\!]_{\mathfrak{M},w}(id) = A$$

donde:

$$\begin{aligned} A &= [\![(\rho', s)]\!]_{\mathfrak{M},w}(com(1,2,\varphi)\circ id) \\ \rho' &= \{\!|\overline{y}_2\varphi, y_3(z).\mathbf{0}, y_4(z).\mathbf{0}|\!\} \end{aligned}$$

En este momento, ρ' tiene dos posibles reducciones, sean A' y A'', por lo que nuevamente por (I2.2):

$$A = A' \cup A''$$

Donde:

$$\begin{aligned} A' &= [\![(\{\!|y_3(z).\mathbf{0}|\!\}, s)]\!]_{\mathfrak{M},w}(com(2,4,\varphi) \circ com(1,2,\varphi) \circ id) \\ &= \{com(2,4,\varphi) \circ com(1,2,\varphi) \circ id\} \\ A' &= [\![(\{\!|y_4(z).\mathbf{0}|\!\}, s)]\!]_{\mathfrak{M},w}(com(2,3,\varphi) \circ com(1,2,\varphi) \circ id) \\ &= \{com(2,3,\varphi) \circ com(1,2,\varphi) \circ id\} \end{aligned}$$

En ambos casos, el ambiente resultante es un proceso de salida que no tiene un proceso receptor complementario. Por lo tanto, el resultado de interpretar el ambiente resultante en ambos casos es simplemente la función calculada hasta ese punto.

Finalmente, simplemente sustituyendo los valores en las ecuaciones anteriores, se tiene que:

$$\begin{aligned} [\![(\{\!|\overline{x}_1\varphi, x_2(z).\overline{y}_2 z, y_3(z).\mathbf{0}, y_4(z).\mathbf{0}|\!\}, s)]\!]_{\mathfrak{M},w}(id) = \\ \{com(2,4,\varphi) \circ com(1,2,\varphi) \circ id, \\ com(2,3,\varphi) \circ com(1,2,\varphi) \circ id\} \end{aligned}$$

Más aún, para este ejemplo, la siguiente afirmación:

$$(\mathfrak{M}, w) \models [P]\chi$$

se cumple si, y sólo sí, se cumplen las siguientes dos afirmaciones:

$$\begin{aligned} (com(2,4,\varphi) \circ com(1,2,\varphi) \circ id)\,(\mathfrak{M}, w) &\models \chi \\ (com(2,3,\varphi) \circ com(1,2,\varphi) \circ id)\,(\mathfrak{M}, w) &\models \chi \end{aligned}$$

Para finalizar esta sección, sólo resta la definición formal de la satisfacción para la parte dinámica del lenguaje.

Definición 4.10 (Satisfacción en $\mathcal{L}_{EA\pi}$ —parte dinámica—)**.** Dados un modelo $\mathfrak{M} = \langle \mathcal{W}, R(a), V \rangle$ y un elemento distinguido $w \in \mathcal{W}$, se define *la relación* $\models$ *entre modelos apuntados y las fórmulas (dinámicas) en* Φ de la siguiente manera:

$$\mathfrak{M}, w \models [P]\varphi$$

si, y sólo si, para toda función ϕ contenida en la clase:

$$[\![(P \oplus \{\!|\}, \mathtt{fn}(P))]\!]_{\mathfrak{M},w}(id)$$

se cumple que:

$$\phi(\mathfrak{M}, w) \models \varphi$$

∎

4.3. Bisimulaciones

Existen ciertas propiedades deseables que la semántica de $\mathcal{L}_{EA\pi}$ debe cumplir. Estas son, básicamente, la preservación de bisimulaciones en los modelos y en los procesos. La primera es válida, pero no la segunda.

Para definir la bisimulación entre procesos es necesario adaptar la definición de la relación de transición $\xrightarrow{\alpha}$ entre procesos asíncronos para tomar en cuenta el envío de mensajes con proposiciones atómicas. Para este fin, modificamos el conjunto de etiquetas o acciones y las reglas de transición para diferenciar el uso de los mensajes que contienen proposiciones atómicas.

Definición 4.11 (Relación de transición $\xrightarrow{\alpha}$ entre procesos de $\mathcal{L}_{EA\pi}$)**.** Sea el siguiente conjunto de etiquetas:

$$\begin{aligned} Act = &\{\overline{x}_i\eta \mid i \in \mathcal{A} \text{ y } \eta \in \mathcal{N} \cup \Phi\} \\ &\cup \{x_i\eta \mid i \in \mathcal{A} \text{ y } \eta \in \mathcal{N} \cup \Phi\} \\ &\cup \{\overline{x}_{i(y)} \mid i \in \mathcal{A} \text{ y } y \in \mathcal{N}\} \\ &\cup \{\tau\} \end{aligned}$$

Se define la *relación de transición etiquetada* $\xrightarrow{\alpha}$ *entre procesos de* $\mathcal{L}_{EA\pi}$ de acuerdo con las reglas de la tabla 3.3, pero con la siguientes modificaciones:

$$\textsc{Inp}\ \frac{}{x_i(z).P \xrightarrow{x_j\eta} P\{^{\eta}/_{z}\}} x \in \mathcal{N} \qquad \textsc{Comm-L}\ \frac{P \xrightarrow{x_i\eta} P' \qquad Q \xrightarrow{\overline{x}_j\eta} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} x \in \mathcal{N}$$

$$\textsc{Out}\ \frac{}{\overline{x}_i\eta \xrightarrow{\overline{x}_i\eta} \mathbf{0}} x \in \mathcal{N}, \eta \in \mathcal{N} \cup \Phi \qquad \textsc{Open}\ \frac{P \xrightarrow{\overline{x}_i y} P'}{(\boldsymbol{\nu} y)\ P \xrightarrow{\overline{x}_i(y)} P'} x, y \in \mathcal{N} \wedge x \neq y$$

∎

Para este sistema de transiciones, sólo se hacen las modificaciones necesarias para diferenciar las acciones hechas por distintos agentes.

Por ejemplo, sean los siguientes dos procesos:

$$\overline{x}_1\varphi \qquad \overline{x}_2\varphi$$

En ambos procesos tienen la capacidad de enviar un mensaje φ por el canal x. Sin embargo, la acción es realizada por agentes distintos, y por lo tanto, se consideran transiciones distintas ($\xrightarrow{\overline{x}_1\varphi}$ y $\xrightarrow{\overline{x}_2\varphi}$).

Un caso diferente se da con los prefijos de entrada:

$$x_1(z).P \qquad x_2(z).P$$

Ambos procesos son capaces de recibir algún mensaje η por el canal x, pero aquí, el mensaje pudo ser emitido por cualquier agente, por lo que tienen las mismas transicionse posibles: $\xrightarrow{x_1\eta}$, $\xrightarrow{x_2\eta}$, $\xrightarrow{x_3\eta}$, etc.

Dado el nuevo sistema de transiciones, se puede reutilizar la definición de bisimulación basada en capacidades para este lenguaje.

Definición 4.12 (Bisimulación asíncrona fuerte entre procesos de $\mathcal{L}_{EA\pi}$)**.** Una relación simétrica S entre procesos asíncronos de $\mathcal{L}_{EA\pi}$ es una *bisimulación asíncrona fuerte* si $(P, Q) \in S$ implica que:

(a) si $P \xrightarrow{\alpha} P'$ y $\alpha \neq x_i\eta$, entonces existe Q' tal que $Q \xrightarrow{\alpha} Q'$ y $(P', Q') \in S$;

(b) si $P \xrightarrow{x_i\eta} P'$, entonces:

(I) existe Q' tal que $Q \xrightarrow{x_i\eta} Q'$ y $(P', Q') \in S$ o

(II) existe Q' tal que $Q \xrightarrow{\tau} Q'$ y $(P', (Q' \mid \overline{x}_i\eta)) \in S$

Si dos procesos P y Q están relacionados por una bisimulación asíncrona fuerte, entonces se dice que son *fuertemente bisimilares* y se escribe $P \sim_a Q$. ∎

Igual que con el cálculo original, se puede dar una definición alternativa.

Definición 4.13 (Bisimulación asíncrona fuerte (alternativa) entre procesos de $\mathcal{L}_{EA\pi}$)**.** *La relación $\sim_{a1}$ de bisimilaridad asíncrona fuerte* se define como la menor relación simétrica, tal que $P \sim_{a1} Q$ implica que:

(a) si $P \xrightarrow{\alpha} P'$ y $\alpha \neq x_i\eta$, entonces existe Q' tal que $Q \xrightarrow{\alpha} Q'$ y $P' \sim_{a1} Q'$ y;

(b) se cumple que $(P \mid \overline{x}_i\eta) \sim_{a1} (Q \mid \overline{x}_i\eta)$ para todo $\overline{x}_i\eta$. ∎

Con la definición alternativa, se puede observar más claramente una diferencia fundamental con el cálculo original, esto es,

$$x_i(z).\overline{x}_i z \not\sim_a \tau.\mathbf{0}$$

Esto es por que los procesos anteriores no se comportan igual al ejecutarlos en paralelo con cualquier proceso de salida (segunda cláusula de la definición de $\sim_{a1}$).

Por ejemplo, para el primero:

$$x_i(z).\overline{x}_i z \mid \overline{x}_j\eta \xrightarrow{\tau} \overline{x}_i\eta$$

y para el segundo:

$$\tau.\mathbf{0} \mid \overline{x}_j\eta \xrightarrow{\tau} \overline{x}_j\eta$$

De lo anterior, se tiene que $\overline{x}_i\eta \sim_{a1} \overline{x}_j\eta$ si, y sólo si, $i = j$. Esto es porque en este lenguaje, un mensaje asíncrono, además de la información propia del mensaje, también contiene la información de quién fue el emisor, por lo que, mensajes iguales de emisores diferentes, en realidad son mensajes diferentes.

Para finalizar la sección, se introduce la discusión que lleva al lenguaje del siguiente capítulo.

De acuerdo con las definiciones anteriores, se tiene que los siguientes procesos son bisimilares:

$$P \stackrel{\text{def}}{=} (\boldsymbol{\nu} x)\ (\overline{x}_1 p \mid x_2(z).\mathbf{0})$$
$$Q \stackrel{\text{def}}{=} \tau.\mathbf{0}$$

- Caso $P \sim_a Q$:
  Se tiene que:

  $$P \xrightarrow{\tau} (\boldsymbol{\nu} x)\ \mathbf{0} \quad \text{(por Res)}$$
  $$Q \xrightarrow{\tau} \mathbf{0} \quad \text{(por Tau)}$$

  Y no hay más transiciones posibles. Desde luego, los subprocesos de P tienen las transiciones independientes $\xrightarrow{\overline{x}_1 p}$ y $\xrightarrow{x_1 p}$, pero x es el sujeto de la restricción, por lo que la condición de la regla Res no se cumple y dichas transiciones no se pueden aplicar.

- Caso $P \sim_{a1} Q$:
  Los dos procesos se transforman en un proceso sin transiciones posibles, de manera que al componerlos en paralelo con cualquier proceso $\overline{x}_i\eta$ este no puede interactuar con ellos quedando sólo una única transición posible $\xrightarrow{\overline{x}_i\eta}$ en ambos casos.

Al construir la interpretación de los procesos anteriores, se tiene lo siguiente:

$$[\![(P \oplus \{\!|\}, \mathtt{fn}(P))]\!]_{\mathfrak{M},w}(id) = \{com(1,2,p) \circ id\}$$
$$[\![(Q \oplus \{\!|\}, \mathtt{fn}(Q))]\!]_{\mathfrak{M},w}(id) = \{tau \circ id\}$$

Si se toma en cuenta un modelo apuntado tal que:

$$\mathfrak{M}, w \models p \wedge K_1 p \wedge \neg K_2 p$$

Se tiene que:

$$\mathfrak{M}, w \models [P] K_2 p$$
$$\mathfrak{M}, w \not\models [Q] K_2 p$$

Lo anterior no es el comportamiento esperado, pues si los dos procesos son "indistinguibles", entonces deberían generar los mismos cambios epistémicos.

En el siguiente capítulo se define un nuevo lenguaje que, por medio de restricciones simples en la sintaxis de $\mathcal{L}_{EA\pi}$, otorga el comportamiento deseado sin restarle utilidad. La prueba de todas las propiedades del lenguaje se posponen hasta el siguiente capítulo.

Capítulo 5

Un lenguaje epistémico con procesos concurrentes y comunicación asíncrona

Para comenzar este capítulo, se introduce como discusión la utilidad de este lenguaje a manera de justificación. Como consecuencia del objetivo de la tesis, se pretende obtener un lenguaje que dado un proceso del cálculo π que describa el *comportamiento* de un sistema, también sea capaz de describir el flujo de conocimiento derivado de la comunicación dentro de dicho proceso.

Con ese objetivo, resulta sumamente útil que dado otro proceso, por ejemplo, más eficiente, se pruebe si el flujo de conocimiento no cambia. Desde luego, la primer medida es verificar que ambos procesos sean bisimilares. Otra posible aplicación sería, dado un cierto modelo de "auditoría" de flujo de información (no sólo de comportamiento, eso ya lo hace el cálculo π por sí mismo) codificado en cálculo π, verificar si un proceso dado cumple con la especificación.

De esta discusión, se desprende el siguiente escenario: Dados algunos procesos, por ejemplo P, Q y R, ejecutados concurrentemente por tres agentes 1, 2 y 3. Se sabe que en una determinada situación $(\mathfrak{M}, w)$, después de su ejecución se cumplen ciertas propiedades φ. Descrito simbólicamente, por ejemplo, con la siguiente notación:

$$\mathfrak{M}, w \models [(P)_1 \mid (Q)_2 \mid (R)_3]\varphi$$

Si el agente 1 intercambia su proceso P por otro equivalente P' (i.e., $P \sim_a P'$), ¿se debe conservar la afirmación? Es decir, ¿se cumple lo siguiente?

$$\mathfrak{M}, w \models [(P')_1 \mid (Q)_2 \mid (R)_3]\varphi$$

En este capítulo se define un lenguaje capaz de modelar este tipo de escenarios, basándose únicamente en las herramientas construidas en el capítulo anterior.

5.1. Lenguaje

Al lenguaje presentado se le denomina $\underline{\mathcal{L}_{EA\pi}}$, pues como se verá más adelante, realmente es un subconjunto de $\mathcal{L}_{EA\pi}$. Se define la sintaxis a continuación.

Definición 5.1 (Sintaxis de las fórmulas de $\underline{\mathcal{L}_{EA\pi}}$)**.** Sean

$$\mathcal{P} = \{p, q, \ldots\}$$

un conjunto numerable de proposiciones atómicas,

$$\mathcal{A} = \{1, \ldots, n\}$$

un conjunto finito de agentes y $\underline{\Pi}$ el conjunto de los procesos asíncronos de la definición 5.2. Entonces, se define *el conjunto* $\underline{\Phi}$ *de las fórmulas de* $\underline{\mathcal{L}_{EA\pi}}$ como el menor conjunto tal que:

(a) $\bot \in \underline{\Phi}$;

(b) si $p \in \mathcal{P}$, entonces $p \in \underline{\Phi}$;

(c) si $\varphi \in \underline{\Phi}$, entonces $\neg\varphi \in \underline{\Phi}$;

(d) si $\varphi, \psi \in \underline{\Phi}$, entonces $\varphi \vee \psi \in \underline{\Phi}$;

(e) si $\varphi \in \underline{\Phi}$ e $i \in \mathcal{A}$, entonces $K_i\varphi \in \underline{\Phi}$;

(f) si $\varphi \in \underline{\Phi}$ y $G \subseteq \mathcal{A}$, entonces $C_G\varphi \in \underline{\Phi}$;

(g) si $\varphi \in \underline{\Phi}$ y $P \in \underline{\Pi}$, entonces $[P]\varphi \in \underline{\Phi}$. ∎

Definición 5.2 (Sintaxis de los procesos de $\underline{\mathcal{L}_{EA\pi}}$)**.** *El conjunto* $\underline{\Pi}$ *de los procesos de* $\underline{\mathcal{L}_{EA\pi}}$ es la composición paralela:

$$(P_1)_{i_1} \mid \cdots \mid (P_n)_{i_n}$$

Donde $n > 0$, cada i_j es un agente en $\mathcal{A}$, no necesariamente distintos entre sí, y cada P_k es un proceso en el conjunto $\underline{\Pi'}$, construido como el menor conjunto que cumple con las siguientes reglas inductivas:

(a) $\mathbf{0} \in \underline{\Pi'}$;

(b) si $x, y \in \mathcal{N}$, entonces $\overline{x}y \in \underline{\Pi'}$;

(c) si $x \in \mathcal{N}$ y $\varphi \in \underline{\Phi'}$, entonces $\overline{x}\varphi \in \underline{\Pi'}$;

(d) si $x, z \in \mathcal{N}$ y $P \in \underline{\Pi'}$, entonces $x(z).P \in \underline{\Pi'}$;

(e) si $P, Q \in \underline{\Pi'}$, entonces $P \mid Q \in \underline{\Pi'}$;

(f) si $z \in \mathcal{N}$ y $P \in \underline{\Pi'}$, entonces $(\boldsymbol{\nu} z)\ P \in \underline{\Pi'}$;

(g) si $\overline{x}\eta \in \underline{\Pi'}$, entonces $!\overline{x}\eta \in \underline{\Pi'}$;

(h) si $x(z).P \in \underline{\Pi'}$, entonces $!x(z).P \in \underline{\Pi'}$. ∎

Las fórmulas del lenguaje continúan siendo las mismas que las de $\mathcal{L}_{EA\pi}$, por lo que no se hablará más de ellas. Los procesos también son similares, sólo que aquí en lugar de indicar el agente que realiza cada acción atómica, se asocia un proceso completo a un agente. Esto es, la notación $(P)_1 \mid (Q)_2$ indica que el agente 1 ejecuta todo el proceso P en paralelo con el proceso Q que ejecuta el agente 2.

La intuición para la estructura interna de los procesos también es la misma. Por ejemplo, en el proceso:

$$(\overline{x}\varphi \mid y(z).\mathbf{0})_1 \mid (\overline{y}\psi \mid x(z).\mathbf{0})_2$$

se indica que el subproceso del agente 1 es capaz de enviar un dato por el canal x y recibir otro por el canal y. De la misma manera, el subproceso del agente 2 tiene la capacidad de realizar las acciones complementarias a las del subproceso del agente 1.

La codificación anterior, también ayuda a clarificar bastante la sintaxis del lenguaje, ya que los procesos son prácticamente iguales a los del cálculo π asíncrono original, con la única diferencia de que aquí también se permite el envío de fórmulas arbitrarias.

Otra diferencia que debe mencionarse con respecto al lenguaje del capítulo anterior es la ausencia de prefijos τ. Esta restricción realmente no quita expresividad a los procesos, pues un prefijo τ puede emularse fácilmente de acuerdo con la siguiente ecuación:

$$\tau.P \stackrel{\text{def}}{=} ((\boldsymbol{\nu} x)\ (\overline{x}y \mid x(y).P))$$

donde x e y son nombres que no ocurren libres en P.

5.2. Semántica

La semántica de este lenguaje pide darse en términos de $\mathcal{L}_{EA\pi}$. Para este propósito, es necesario definir una mapeo que traduzca las fórmulas de un lenguaje a otro. Ese el objetivo de la siguiente función:

$$T : \underline{\Phi} \to \Phi$$

Esencialmente, el funcionamiento de T es descender inductivamente por una fórmula para transformarla en otra. Cada fórmula construida de acuerdo con la definición 5.1 se deja intacta, con la excepción de las fórmulas de tipo $[P]\varphi$. En este caso, se extiende la funcionalidad de T con un mapeo para los procesos que, se espera no cause confusión, se denomina también T. Esto es,

$$T : \underline{\Pi'} \times \mathcal{A} \to \Pi$$

La definición completa del mapeo de fórmulas se omite, pues es trivial. El único caso relevante es el ya mencionado, que es el siguiente:

$$T\left([(P_1)_{i_1} \mid \cdots \mid (P_n)_{i_n}]\varphi\right) = [T(P_1, i_1) \mid \cdots \mid T(P_n, i_n)]T(\varphi) \tag{5.1}$$

La definición completa del mapeo entre procesos se da en la siguiente definición.

Definición 5.3 (Traducción de procesos de $\underline{\mathcal{L}_{EA\pi}}$ a procesos de $\mathcal{L}_{EA\pi}$)**.** Dados un proceso en $\underline{\Pi'}$ y un agente en $\mathcal{A}$, se construye la *función de traducción* T *de procesos de* $\underline{\mathcal{L}_{EA\pi}}$ *a procesos de* $\mathcal{L}_{EA\pi}$ de acuerdo con las siguientes reglas:

$$\begin{aligned}
T(\mathbf{0}, i) &= \mathbf{0} \\
T(\overline{x}y, i) &= \overline{x}_i y \\
T(\overline{x}\varphi, i) &= \overline{x}_i T(\varphi) \qquad (\varphi \in \underline{\Phi}) \\
T(x_{(z)}.P, i) &= x_{i(z)}.(T(P, i)) \\
T(P \mid Q, i) &= (T(P, i)) \mid (T(Q, i)) \\
T((\boldsymbol{\nu} z)\ P, i) &= (\boldsymbol{\nu} z)\ (T(P, i)) \\
T(!P, i) &= !(T(P, i))
\end{aligned}$$

∎

Por ejemplo, por medio de este mapeo la siguiente fórmula de $\underline{\mathcal{L}_{EA\pi}}$:

$$p \Rightarrow [(\overline{x}r \mid y_{(z)}.\mathbf{0})_1 \mid (\overline{y}s \mid x_{(z)}.\mathbf{0})_2]q$$

se traduce en la fórmula de $\mathcal{L}_{EA\pi}$:

$$p \Rightarrow [\overline{x}_1 r \mid y_1(z).\mathbf{0} \mid \overline{y}_2 s \mid x_2(z).\mathbf{0}]q$$

En este punto, es posible definir fácilmente la relación de satisfacción para las fórmulas de este lenguaje en términos del lenguaje anterior.

Definición 5.4 (Satisfacción en $\underline{\mathcal{L}_{EA\pi}}$)**.** Dados un modelo $\mathfrak{M} = \langle \mathcal{W}, \{R_i\}_{i\in\mathcal{A}}, V\rangle$ y un elemento distinguido $w \in \mathcal{W}$, se define *la relación* $\models$ *entre modelos apuntados y las fórmulas en* $\underline{\Phi}$ de la siguiente manera:

$$\mathfrak{M}, w \models \varphi \text{ sii } \mathfrak{M}, w \models T(\varphi)$$

Donde, en el lado derecho $\models$ se refiere a la relación de satisfacción de $\mathcal{L}_{EA\pi}$. ∎

5.3. Bisimulaciones

Un proceso de $\underline{\mathcal{L}_{EA\pi}}$ es la composición paralela de uno o más subprocesos ejecutados por algún agente en particular. Estos subprocesos son en realidad procesos de $\mathcal{L}_{EA\pi}$, por lo que la definición de bisimilaridad también puede extenderse para los procesos de este lenguaje. Sin embargo, aquí sólo va a interesar comparar los subprocesos independientes de esa composición paralela.

Definición 5.5 (Bisimulación para procesos de $\underline{\mathcal{L}_{EA\pi}}$)**.** *Una relación S entre procesos de $\underline{\Pi'}$ es una bisimulación* sii la relación $S' = \{(T(P,i), T(Q,i)) \mid (P,Q) \in S \wedge i \in \mathcal{A}\}$ para cualquier i, es una bisimulación de procesos de $\mathcal{L}_{EA\pi}$ (definición 4.12). También, se dice que *dos procesos P y Q son bisimilares,* $P \sim_a Q$, si $T(P,i) \sim_a T(Q,i)$, cualquier $i \in \mathcal{A}$. De la misma forma se dice que *dos procesos P y Q son 1-bisimilares,* $P \sim_{a1} Q$, si $T(P,i) \sim_{a1} T(Q,i)$ (definición alternativa 4.13). ∎

Ahora ya se pueden demostrar las propiedades del lenguaje. Pero primero una aclaración.

Convención 5.6. Con el fin de evitar repeticiones, en los siguientes lemas y teoremas se asume que siempre que un agente emite un mensaje, este es veraz. Es importante hacer esta aclaración, ya que, por ejemplo, los siguientes procesos:

$$(\boldsymbol{\nu} x)\ (\overline{x}_i\varphi \mid x_i(z).\mathbf{0}) \sim_a \tau.\mathbf{0}$$

aunque son bisimilares, si en el momento de evaluar el significado del mensaje $\overline{x}_i\varphi$, resulta que $K_i\varphi$ no se satisface, entonces no se va a preservar la bisimilaridad en el modelo semántico. ∎

Los siguientes lemas sobre las funciones de actualización servirán de apoyo para la prueba de las propiedades del lenguaje.

Lema 5.7. Las funciones de actualización *id*, *tau* y *com* y la composición de estas preservan la bisimilaridad de un modelos, Es decir, $(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}', w')$ sii $\phi(\mathfrak{M}, w) \leftrightarrows \phi(\mathfrak{M}', w')$ con ϕ una función de actualización o la composición de funciones de actualización.

Demostración. Para las funciones *id* y *tau* es trivial pues sólo son la identidad de modelos. Para la función *com* cuando su argumento es una fórmula, se tiene que es la aplicación del producto $\otimes$ al modelo y una estructura de acción $\mathfrak{A}^{\varphi}_{i,j}$, esto vale por el teorema 2.19. De la misma manera, para la composición de estas funciones también vale, pues el resultado de aplicar una función es un modelo, que como ya se vio preserva bisimulación, entonces al aplicarle de nuevo otra función se sigue preservando la bisimulación. ∎

Este lema indica que cualquier función que se le aplique a dos modelos bisimilares, va a resultar en otros dos modelos bisimilares, pues si los modelos originales representan el mismo estado epistémico del conjunto de agentes, entonces aplicarles una función tendrá que resultar en modelos con las mismas modificaciones (tomando en cuenta la convención de que los agentes sean veraces).

Lema 5.8. Sea un agente $i \in \mathcal{A}$, una fórmula arbitraria $\varphi \in \Phi$ y un modelo apuntado $(\mathfrak{M}, w)$ tal que $(\mathfrak{M}, w) \models K_i\varphi$. Entonces, $com(i, i, \varphi)(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}, w)$.

Demostración. Por definición se tiene que $com(i, i, \varphi)(\mathfrak{M}, w) = (\lambda M.(\mathfrak{A}^{\varphi}_{i,i}, e) \otimes M)(\mathfrak{M}, w) = (\mathfrak{A}^{\varphi}_{i,i}, e) \otimes (\mathfrak{M}, w)$. Primero se muestra cómo este producto no altera el estado epistémico de los agentes en $\mathcal{A} - \{i\}$, y después se muestra lo mismo para el agente i. En el nuevo modelo generado los mundos posibles son parejas (w', e) y $(w', \top)$, donde w' es un mundo posible del modelo original y se conservan sus valuaciones. De las parejas (w', e), se eliminan las que no satisfacen $K_i\varphi$, de las parejas $(w', \top)$ no se elimina ninguna. Los agentes en $\mathcal{A} - \{i\}$ sólo va a a tener acceso a las parejas $(w', \top)$ sii ya tenían acceso al w' original. Para todos los agentes en $\mathcal{A}$ las relaciones de accesibilidad entre parejas $(w', \top)$ quedan igual, y como conservan las valuaciones originales, entonces también van a satisfacer exactamente las mismas fórmulas. Para el agente i sólo van a ser accesibles, en un paso, las parejas (w', e), y como estas parejas contienen los mismos w' que ya eran accesibles por i en el modelo original, entonces, no cambia el conocimiento de i de primer orden (no considera el conocimiento del conocimiento de los demás). Para el conocimiento de i de

orden superior, se toman en cuenta las relaciones de accesibilidad del resto de los agentes, como éstas sólo llevan a los mundos $(w', \top)$ que ya eran accesibles en el modelo original, y entre mundos $(w', \top)$ las relaciones quedan iguales, entonces también van a satisfacer las mismas fórmulas. ∎

El lema 5.8 indica que si un agente consume un mensaje que él mismo emitió, no se afecta la información epistémica del modelo resultante. Esto es, aunque el resultado de aplicar una función $com(i, i, \varphi)$ a algún modelo $\mathfrak{M}, w$ no es estrictamente en el mismo modelo, sí se conserva la información de éste, pues el resultado es bisimilar al modelo original.

Teorema 5.9 ($\leftrightarrows$ preserva $\models$)**.** *Sean una fórmula φ de $\underline{\mathcal{L}_{EA\pi}}$ y dos modelos apuntados $(\mathfrak{M}, w)$ y $(\mathfrak{M}', w')$ tales que $(\mathfrak{M}, w) \leftrightarrows (\mathfrak{M}', w')$. Entonces,*

$$(\mathfrak{M}, w) \models \varphi \text{ sii } (\mathfrak{M}', w') \models \varphi$$

Demostración. De nuevo, toda la argumentación está dada en la prueba del teorema 2.14, el único caso relevante es para las fórmulas de tipo $[P]\varphi$. En este caso, se tiene que a P primero se le aplica la función de traducción T para convertirlo en un proceso de $\mathcal{L}_{EA\pi}$, por lo que su interpretación $[\![]\!]$ es una clase que contine funciones de actualización que por el lema 5.7 preservan la bisimulación. Por lo tanto, (por hipótesis inductiva) también se preserva la satisfacción. ∎

El teorema 5.9 simplemente indica que la propiedad de la lógica epistémica donde dos modelos bisimilares satisfacen las mismas fórmulas, también se tiene para el lenguaje $\underline{\mathcal{L}_{EA\pi}}$ (y, de hecho, también para $\mathcal{L}_{EA\pi}$).

Teorema 5.10 ($\sim_a$ preserva $\models$)**.** *Sea un proceso $(P_1)_{i_1} \mid \cdots \mid (P_k)_{i_k} \mid \cdots \mid (P_n)_{i_n}$ de $\underline{\mathcal{L}_{EA\pi}}$ y un modelo apuntado $(\mathfrak{M}, w)$. Sea un proceso Q_k tal que para algún P_k, $P_k \sim_a Q_k$. Entonces, se cumple que:*

$$\mathfrak{M}, w \models [(P_1)_{i_1} \mid \cdots \mid (P_k)_{i_k} \mid \cdots \mid (P_n)_{i_n}]\chi$$

si, y sólo si,

$$\mathfrak{M}, w \models [(P_1)_{i_1} \mid \cdots \mid (Q_k)_{i_k} \mid \cdots \mid (P_n)_{i_n}]\chi$$

Demostración. La prueba es por inducción sobre el número n de subprocesos. En cada caso también se procede inductivamente sobre la estructura del proceso, y se muestra su comportamiento en todas sus posibles transiciones.

En la base de la inducción está el caso $n = 1$. Aquí se tiene que si P_1 y Q_1 tienen una transición $\xrightarrow{\overline{x}_{i_i}\eta}$, entonces hay un elemento $\overline{x}_{i_1}\eta$ o $!\overline{x}_{i_1}\eta$ en el ambiente ρ resultado de la operación $\oplus$ en ambos casos.

Si los procesos tienen una transición $\xrightarrow{\overline{x}_{i_i}(y)}$, entonces, en el ambiente generado también habrá un elemento $\overline{x}_{i_1}y^n$ o $!\overline{x}_{i_1}y^n$ como resultado de la operación $\oplus$ en ambos casos, con y^n un nombre nuevo y único generado por $\oplus$.

Si los procesos tienen una transición $\xrightarrow{\tau}$, entonces como en este lenguaje no se tiene el prefijo τ como primitiva, tuvo que ser simulado por la comunicación de dos subprocesos. El primer caso es cuando la comunicación es por canales restringidos de la forma $(\boldsymbol{\nu}x)\ (\overline{x}_{i_1}\eta \mid x_{i_1}(z).P')$, aquí la restricción se elimina generando un nombre único x^n, por lo que los procesos van a interactuar genrando una función $com(i_1, i_1, \eta)$ y por el lema 5.8 dicha función preserva $\models$ y como el nombre es único, esa es la única interacción que pueden tener esos elementos sin afectar más. El otro caso es cuando la transición se genera por la comunicación de dos procesos fuera de una restricción. Aquí, se tiene que forzosamente deben haber una transición con una acción de salida y otra con una acción de entrada en los dos procesos y, por lo tanto, sus correspondientes elementos en el ambiente generado.

Si el proceso P_1 tiene una transición de entrada $\xrightarrow{x_{i_1}(\eta)}$, entonces pueden ocurrir alguno de los siguientes dos casos. El primer caso es que Q_1 tenga una transición igual, entonces en ambas traducciones el ambiente generado va a tener un elemento de tipo $x_{i_1}(z).P'$. En el segundo caso se tiene que $P_1 \xrightarrow{x_{i_1}(\eta)} P_1'$, $Q_1 \xrightarrow{\tau} Q_1'$ y $P_1' \sim_a (Q_1' \mid \overline{x}_{i_1}\eta)$ para cualquier $\overline{x}_{i_1}\eta$. Para la transición τ de Q_1 aplica el argumento del párrafo anterior, lo importante es que $P_1' \sim_a (Q_1' \mid \overline{x}_{i_1}\eta)$, en este caso se tiene que ambos procesos van a tener una acción de salida $\xrightarrow{\overline{x}_{i_1}}$ y su correspondiente elemento en el ambiente generado.

Ahora se va a probar el caso $n > 1$, es decir, cuando el proceso se ejecuta concurrentemente con los procesos de otros agentes.

Si los procesos bisimilares tienen una transición $\xrightarrow{\overline{x}_{i_1}\eta}$ o $\xrightarrow{\overline{x}_{i_1}(\eta)}$ (con η un nombre en este caso) en el ambiente generado van a existir elementos $\overline{x}_{i_1}\eta$, $\overline{x}_{i_1}\eta^n$ (η^n un nombre único generado por $\oplus$) o una replicación $!\overline{x}_{i_1}\eta$. Esto implica que si el proceso de otro agente tiene un proceso que pueda escuchar por el canal x, entonces se va a generar una función $com(i_i, i_j, \eta)$ en todos los casos. Si no existe ningún proceso de otro agente que escuche por el mismo canal, entonces no se produce ningún cambio epistémico.

Si los procesos tienen una transición $\xrightarrow{\tau}$, entonces se tienen dos casos. Primero que la transición τ fue generada bajo una restricción del canal transmisor y receptor, por lo que la recepción la realiza el mismo agente sin causar cambios epistémicos (lema 5.8). Si la transición es generada por la comunicación de un canal libre (nuevamente entre el mismo agente), entonces también deben existir una transición de salida (párrafo anterior) y una de entrada, para lo

que aplica el argumento del párrafo siguiente.

Si $P_1 \xrightarrow{x_{i_1}\eta} P_1'$, entonces ocurre uno de los dos casos siguientes. El primero es que $Q_1 \xrightarrow{x_{i_1}\eta} Q_1'$, con $P_1' \sim_a Q_1'$, por lo que en el ambiente generado para cualquiera de los dos procesos, va a existir un proceso capaz de escuchar un mensaje por el canal x y, si cualquier proceso de otro agente emite un mensaje por ese canal, se va a poder consumir. El segundo caso es que $Q_1 \xrightarrow{\tau} Q_1'$ y $P_1' \sim_a (Q_1' \mid \overline{x}_{i_1}\eta)$. En este caso, se tiene que *después* de consumir algún mensaje $\overline{x}_{i_j}\eta$, P_1' deberá tener la capacidad de emitir el mismo mensaje, esto sólo sucede si el mensaje es del mismo agente que lo consumió, es decir $i_1 = i_k$, y por lo tanto no habrá cambios en su estado epistémico, después de esto los procesos seguirán comportándose igual (por hipótesis inductiva). ∎

El teorema 5.10 representa el principal resultado del trabajo. Como se estableció en la introducción de este capítulo, con este teorema se asegura la modularidad de los sistemas modelados con este lenguaje. Por ejemplo, si cualquier agente cambia la implementación de sus procesos y no se conservan las propiedades, entonces la implementación puede ser incorrecta de acuerdo a las especificaciones.

5.4. Sincronización

En la última sección del capítulo 3 se habló de un protocolo de sincronización para el cálculo π asíncrono. El protocolo se basa en un intercambio de canales privados entre emisor y receptor, de forma que se garantice que las acciones de envío y recepción sean las únicas posibles en el sistema de transiciones.

Desde luego, se espera la pregunta de si el uso de este protocolo por los agentes de este lenguaje afecta el conocimiento común entre ellos. Desafortunadamente la respuesta no es positiva. Para lograr esto, hace falta un lenguaje más expresivo donde los agentes posean conocimiento sobre la estructura de los programas que usan para comunicarse. Para aclarar este punto, se presenta un ejemplo del intercambio de un dato simple.

Los procesos de envío y recepción para una proposición atómica p, son los siguientes:

$$E \stackrel{\text{def}}{=} (\boldsymbol{\nu} u)\ (\overline{x}u \mid u(v).\overline{v}p)$$
$$R \stackrel{\text{def}}{=} (\boldsymbol{\nu} v)\ (x(u).(\overline{u}v \mid v(z).\mathbf{0}))$$

Entonces, interesa saber la interpretación del siguiente proceso:

$$(E)_i \mid (R)_j$$

donde i es el agente emisor y j el receptor.

Para hacer explícitas las acciones atómicas que realiza cada agente, primero se debe traducir el proceso a uno del lenguaje $\mathcal{L}_{EA\pi}$ usando la función de traducción T:

$$((\boldsymbol{\nu} u)\ (\overline{x}_i u \mid u_{i(v)}.\overline{v}_i p)) \mid ((\boldsymbol{\nu} v)\ (x_{j(u)}.(\overline{u}_j v \mid v_{j(z)}.\mathbf{0})))$$

El siguiente paso, es eliminar las restricciones de nombres generando nombres únicos, e insertar los procesos paralelos en un ambiente ρ para interpretarlos. Para esto se utiliza la operación $\oplus$ con un ambiente inicial vacío. El resultado es el siguiente:

$$\rho = \{\!|\overline{x}_i u^1, u_{i(v)}^1.\overline{v}_i p, x_{j(u)}.(\overline{u}_j v^1 \mid v_{j(z)}^1.\mathbf{0})|\!\}$$

En este ambiente, sólo los procesos que se comunican por el canal x pueden interactuar, generando una función de actualización $com(i, j, u^1)$, el ambiente resultante es el siguiente:

$$\rho' = \{\!|u_{i(v)}^1.\overline{v}_i p, \overline{u}_j^1 v^1, v_{j(z)}^1.\mathbf{0}|\!\}$$

Ahora sólo es posible la comunicación por el canal u^1, generando una función $com(j, i, v^1)$, el ambiente restante queda como sigue:

$$\rho'' = \{\!|\overline{v}_i^1 p, v_{j(z)}^1.\mathbf{0}|\!\}$$

Esta última interacción por el canal v^1 genera la función de actualización $com(i, j, p)$. Como el proceso restante es el proceso nulo, el ambiente queda vacío, por lo que se detiene la interpretación.

En cada interacción, se construye la composición de las funciones generadas, a partir de la función inicial id. La función resultante es la siguiente:

$$com(i, j, u^1) \circ com(j, i, v^1) \circ com(i, j, p) \circ id$$

Como en las funciones $com(i, j, u^1)$ y $com(j, i, v^1)$ el objeto transmitido es un nombre, estas son equivalentes a la función identidad. Por lo tanto, la función anterior es equivalente a $com(i, j, p)$. La aplicación de esta función a cualquier modelo $(\mathfrak{M}, w)$, es la siguiente:

$$com(i, j, p)(\mathfrak{M}, w) = (\mathfrak{A}_{i,j}^p, e) \otimes (\mathfrak{M}, w)$$

En el modelo resultante, el agente j sólo tiene acceso a los mundos donde $p \wedge K_i p$ es el caso, mientras que para el resto de los agentes no se alteran sus relaciones de accesibilidad. Esto es, el emisor sigue sin enterarse que su mensaje fue recibido.

El problema es que aunque la interpretación del proceso respeta las transiciones del sistema, la única transición posible sigue representando comunicación asíncrona. Este punto no queda del todo claro en el cálculo original, pero al darle una interpretación epistémica queda claro.

Para que se pueda lograr comunicación síncrona, o bien, se puede agregar esta característica en el lenguaje de acciones, o bien, se necesita un lenguaje más expresivo. El intercambio de canales privado indica que los agentes conocen la estructura del programa con el que se están comunicando, y están seguros que el mensaje que envían por el canal privado llegará a su destino. Lo anterior no puede expresarse en el lenguaje $\mathcal{L}_{EA\pi}$ y, probablemente necesitaría un lenguaje de orden superior.

Sin embargo, si fuera posible describir escenarios de este tipo, aún quedarían cuestiones a resolver. Por ejemplo, ¿cómo puede un agente estar seguro de que el agente con el que se comunica no cambia la estructura de su programa? Esto podría darse por fallas u otras causas.

De cualquier manera, con este ejemplo también se ilustra como este lenguaje puede describir escenarios como el del problema del consenso. De hecho, el que no se alcance el conocimiento común es precisamente por la imposibilidad de resolver dicho problema.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Resumen

Antes de hablar de las posibles extensiones a la tesis, es conveniente hacer un resumen de ésta, resaltando sus ideas generales.

En el contexto de la lógica epistémica, las estructuras de Kripke representan el estado epistémico de uno o más agentes de un sistema. En un estado epistémico se describe el conocimiento de los agentes sobre los hechos del sistema, esto es, sobre las fórmulas del lenguaje. La descripción también incluye conocimiento de orden superior, es decir, conocimiento sobre el conocimiento, tanto propio como de otros agentes.

Una estructura de acciones, es una estructura de Kripke pero que modela un evento de comunicación, por ejemplo, un anuncio público o un anuncio privado. Al aplicar el producto de modelos $\otimes$ a dos estructuras de Kripke, una que describe un evento de comunicación y otra que describe un estado epistémico, el resultado es un nuevo estado epistémico que describe el conocimiento de los agentes después de la comunicación descrita por la estructura de acciones. Desde el punto de vista anterior, un evento de comunicación puede interpretarse como una función de actualización, cuya aplicación lleva de un estado epistémico a otro.

Por otra parte, con el cálculo π se pude describir el comportamiento de procesos concurrentes, cuyas acciones atómicas o básicas son las de envío y recepción de mensajes. Entonces, la ejecución de un proceso no es más que una secuencia de intercambio de mensajes, por lo que sus efectos en el estado epistémico de un grupo de agentes, puede interpretarse como la aplicación secuencial de funciones de actualización.

El lenguaje definido toma como base la lógica epistémica proposicional

estándar, con los operadores modales K_i y C_G de conocimiento y conocimiento común. El operador $[P]$ indica la ejecución de un proceso P, el cual está construido como la composición paralela de uno o más procesos del cálculo π asíncrono. Para cada proceso de esta composición paralela, se indica el agente que lo ejecuta, de forma que un proceso como $(P)_i \mid (Q)_j$ puede leerse como: el agente i ejecuta el proceso P, concurrentemente con el proceso Q del agente j.

Para dar significado a los procesos de este lenguaje, se necesitan varios pasos. Primero, se aplica un tratamiento sintáctico donde a cada acción atómica del proceso se le anexa la información de qué agente es el que la ejecuta. El segundo paso, también sintáctico, se eliminan las restricciones de nombres de los procesos (i.e., se eliminan los operadores $(\boldsymbol{\nu} z)$), para después generar, conforme a las reglas de transición de los procesos, una función de actualización que represente las modificaciones en el estado epistémico por la ejecución de un proceso. Después de este tratamiento, queda como resultado una clase que contiene varias funciones de actualización, una para cada posible ejecución de un proceso. Al aplicar una de estas funciones a un modelo, resulta un modelo actualizado por todos los eventos de comunicación de la ejecución del proceso.

Por último, se mostró que la interpretación de los procesos preserva la bisimulación de procesos y modelos. Esto es, si se intercambia el proceso de un agente por otro bisimilar en una fórmula, ésta seguirá siendo satisfecha por los mismos modelos, y también, dos modelos bisimilares van a satisfacer exactamente las mismas fórmulas.

6.2. Trabajo futuro

Hay varias formas de extender el trabajo de esta tesis, desde los puntos de vista teórico y práctico. En la parte teórica, la extensión más evidente es la búsqueda de una axiomatización apropiada, de manera que se tenga un método de demostración sintáctico, además del semántico. Desde el punto de vista práctico, una extensión muy sencilla es la implementación de un verificador de modelos.

La tarea de implementar un programa que verifique la satisfacción de las fórmulas del lenguaje es muy directa. Como se mencionó, parte de la semántica del lenguaje está basada en la máquina abstracta de [PC04], por lo que es muy cercana a la implementación en un lenguaje funcional con manejo de listas. Para esto, simplemente se sustituye el uso de multiconjuntos, por el uso de listas, y la unión de multiconjuntos en la operación $\oplus$, por la concatenación de listas. Un problema que podría surgir, es la interpretación de procesos con

transiciones infinitas. Para esto puede sustituirse el operador de replicación $!P$, con otro finito $!_n P$ donde $n \geq 0$ indica el número máximo de replicaciones paralelas de P. Para dicho operador, puede darse la siguiente regla estructural:

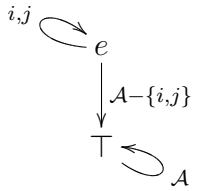
$$!_0 P \equiv \mathbf{0}$$
$$!_{n+1} P \equiv P \mid !_n P$$

Desde luego limitar el lenguaje a procesos finitos disminuye notablemente su capacidad expresiva, sin embargo, el lenguaje sigue siendo útil para mostrar propiedades de un sistema por partes.

Desde el punto de vista teórico, también puede buscarse extender el cálculo a la versión síncrona. Esto no representa un gran problema, pues las herramientas semánticas son las mismas. Por ejemplo, para representar la comunicación síncrona de dos procesos, puede pensarse en sustituir la función de actualización $com(i, j, \eta)$ por una versión síncrona:

$$scom(i, j, \eta) \stackrel{\text{def}}{=} \lambda M.(\mathfrak{S}^{\eta}_{i,j}, e) \otimes M$$

Donde el modelo de acción $\mathfrak{S}^{\eta}_{i,j}$ (con η una fórmula) puede representarse con la siguiente gráfica:



En este modelo, la acción e representa la comunicación síncrona entre los agentes i y j. De este modo, sólo ellos se percatan de la acción e, mientras que los demás agentes creen que no sucedió nada (acción $\top$). También, el agente receptor conoce quién es el emisor, por lo que la precondición de la acción se expresa como $\textsc{Pre}(e) = \eta \wedge K_i \eta$. La diferencia con la comunicación asíncrona es que, en este caso ambos agentes están consientes del evento sucedido, por lo que es posible que el conocimiento común entre ellos se afecte positivamente.

Otra adición posible es considerar como primitiva del lenguaje los prefijos de concordancia **if** $x = y$ **then** P. Más aún, podría pensarse en un prefijo basado en fórmulas del lenguaje, por ejemplo, **if** φ **then** P. De esta manera, podrían construirse programas que eviten el envío de mensajes inválidos o no veraces.

Existen muchas variantes del cálculo π, con las que se podría buscar extender el trabajo. Como se mencionó en la introducción de la tesis, una ventaja que tiene el lenguaje presentado en este trabajo es que es capaz modelar cómo

suceden los cambios epistémicos, a diferencia de otros lenguajes epistémicos que sólo expresan que la acción sucedió, sin decir cómo fue la estructura interna de la comunicación. En este sentido, puede buscarse un lenguaje de acciones que pueda ser interpretado como un rango mucho más amplio de estructuras de acciones.

Sin embargo, lo anterior no es una tarea del todo fácil ya que la capacidad expresiva de las estructuras de acciones es muy amplia. Por ejemplo, puede expresarse comunicación multipunto, comunicación donde otros agentes "espían" sin que los demás se percaten de ello, etc. Es probable, que un lenguaje de acciones que represente todas las posibilidades sea poco práctico, pero pueden identificarse cuáles son las características necesarias y suficientes para expresar determinados escenarios. De esta manera, podría elegirse el lenguaje adecuado dependiendo del contexto de la aplicación.

Como se mencionó al final del capítulo 5, una limitante del lenguaje es que no puede expresar un nivel de conocimiento extra de los agentes sobre la propia estructura de la comunicación. Puede buscarse una manera de añadir esta característica al lenguaje. Sin embargo, además de la dificultad, pueden presentarse otros problemas como, por ejemplo, qué pasaría si los agentes asumieran cierta estructura de los programas con los que se comunican y sus suposiciones son erróneas. Esto puede darse con programas defectuosos, etc.

Además de las aplicaciones directas para protocolos de red u otros sistemas de agentes cognoscitivos concurrentes, también es posible investigar la utilidad de este lenguaje para sistemas secuenciales basados en el paradigma orientado a objetos. Este paradigma de programación también puede modelarse con álgebras de procesos con primitivas de comunicación como el cálculo π. En esta perspectiva, los procesos representan objetos, y la comunicación entre ellos (llamadas a métodos) se puede representar con el envío mensajes. Como medida de un observador externo, la bisimilaridad de procesos puede usarse para comprobar que las implementaciones de una interfaz sean correctas.

Todas las extensiones anteriores se mantienen en la "misma línea" del trabajo. Sin embargo, pueden pensarse otras aplicaciones. Por ejemplo, cómo encaja un lenguaje de este tipo en los sistemas de revisión de creencias y en otras lógicas doxásticas. En estos sistemas, "creencia" denota un concepto más fuerte, en donde un agente cree en algo de acuerdo con un conjunto evidencias. En este contexto, la comunicación implica que los agentes deben mantener la consistencia de sus creencias y evidencias. Existen propuestas (diferentes de los conocidos AGMs) donde se extiende el uso de los modelos y las estructuras de acciones, equipándolos con una relación de preorden. La relación de preorden entre los hechos del sistema, representa lo que los agentes creen que sería el caso, si sus creencias fueran falsas. En este sentido, la revisión se realiza

siguiendo el preorden en caso de inconsistencias (cf. [Auc03]).

En resumen, se presentó un lenguaje con capacidad de describir sistemas con múltiples agentes con comunicación asíncrona, donde además de hablar sobre los cambios en el estado epistémico de los agentes, producto de la comunicación entre estos, con este lenguaje también es posible describir detalladamente cómo se da la comunicación, es decir, su estructura. Lo anterior es la mayor aportación del trabajo, y en los párrafos anteriores se proponen mayor investigación para brindar más utilidad.

Bibliografía

[AGM85] C. A. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50:510–530, 1985.

[AH01] Benjamin Aziz and Geoff W. Hamilton. A denotational semantics for the π-calculus. In *IWFM*, 2001.

[Auc03] G. Aucher. A combined system for update logic and belief revision. Master's thesis, 2003.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, New York, NY, USA, 2001.

[BMS99] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. Technical Report SEN-R9922, CWI, 1999.

[Bou92] Gérard Boudol. Asynchrony and the pi-calculus. Rapport, INRIA Sofia-Antipolis, May 1992.

[FHMV95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Reasoning about knowledge. MIT Press, Cambridge, MA, USA, 1995.

[Hin62] Jaakko Hintikka. *Knowledge and Belief*. Cornell University Press, Ithaca, New York, 1962.

[HKT00] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. Foundations of Computing. MIT Press, October 2000.

[HM84] J. Y. Halpern and Y. O. Moses. Knowledge and common knowledge in distributed environments. In *Proceedings of the 3rd ACM Conference on Principles of Distributed Computing*. ACM Press, 1984.

[HT91] Kohei Honda and Mario Tokoro. An object calculus for asynchronous communication. *Lecture Notes in Computer Science*, 512:133–147, 1991.

[Kri63] Saul A. Kripke. Semantical analysis of modal logic I. *Zeitschr. Math. Logik Grund. Math*, 9:67–96, 1963.

[Mil80] Robin Milner. A calculus of communicating systems. *LNCS*, 92, 1980.

[MPW92] R. Milner, J. Parrow, and J. Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–40,41–77, September 1992.

[Nes00] Uwe Nestmann. What is a 'good' encoding of guarded choice? *Journal of Information and Computation*, 156:287–319, 2000.

[NP96] Uwe Nestmann and Benjamin C. Pierce. Decoding choice encodings. In Ugo Montanari and Vladimiro Sassone, editors, *CONCUR '96: Concurrency Theory, 7th International Conference*, volume 1119, pages 179–194, Pisa, Italy, 1996. Springer-Verlag.

[Pal97] Catuscia Palamidessi. Comparing the expressive power of the synchronous and the asynchronous pi-calculus. In *Symposium on Principles of Programming Languages*, pages 256–265, 1997.

[PC04] Andrew Phillips and Luca Cardelli. A correct abstract machine for the stochastic pi-calculus. In *Concurrent Models in Molecular Biology (Bioconcur'04)*, London, ago 2004.

[PCC06] Andrew Phillips, Luca Cardelli, and Giuseppe Castagna. A graphical representation for biological processes in the stochastic pi-calculus. 4230:123–152, 2006.

[Pla89] Jan A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, and M. Hadzikadic, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[PRSS01] Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80(1):25–31, October 2001.

[Sta96] Ian Stark. A fully abstract domain model for the π-calculus. In *Eleventh Annual Symposium on Logic in Computer Science (LICS) (New Brunswick, New Jersey)*, pages 36–42. IEEE, Computer Society Press, July 1996.

[Sta06] Robert Stalnaker. On logics of knowledge and belief. *Philosophical Studies*, 128(1):169–199, 2006.

[SW01] Davide Sangiorgi and David Walker. *The Pi-Calculus — A Theory of Mobile Processes*. Cambridge University Press, 2001.

[Tur96] David Turner. *The Polymorphic Pi-Calculus: Theory and Implementation*. PhD thesis, 1996.

[vB05] Johan van Benthem. Open problems in logical dynamics. In Dov M. Gabbay, Sergei S. Goncharov, and Michael Zakharyaschev, editors, *Mathematical Problems from Applied Logic I: Logics for the XXIst Century*, International Mathematical Series, page 137. Springer, 2005.

[vBvEK05] Johan van Benthem, Jan van Eijck, and Bartled Kooi. Logics of communication and change. 2005.

[vDvdHK03] H. van Ditmarsch, W. van der Hoek, and B. Kooi. Concurrent dynamic epistemic logic. In K. Jorgensen, V.F. Hendricks, and S. Pederson, editors, *Knowledge Contributors*, pages 105–143. Kluwer Academic Pusblishers, Dordrecht, The Netherlands, 2003.

[VQHQ06] Fernando R. Velázquez-Quesada and Francisco Hernández-Quiroz. Some semantics for a logical language for the game of dominoes. In *AIA'06: Proceedings of the 24th IASTED international conference on Artificial intelligence and applications*, pages 293–298, Anaheim, CA, USA, 2006. ACTA Press.