Una comparación formal entre CCS y ACP *

Pedro Ruben D'Argenio[1]
pedro@unlp.edu.ar

Juan Vicente Echagüe[1,2]
echague@fing.edu.uy

[1] LIFIA
Depto. de Informática
Fac. de Ciencias Exactas
U. Nacional de La Plata
La Plata, Argentina

[2] InCo
Fac. de Ingeniería
Universidad de la República
Montevideo, Uruguay

Resumen

Este artículo presenta una relación formal entre el cálculo CCS de Robin Milner [Mil80, Mil89] y las álgebras de proceso (ACP) mayormente debidas a J. Bergstra y J. Klop [BK85, BW90]. Tal relación formal está dada por una función que mapea términos de CCS a términos de ACP composicionalmente. A partir de esta función se establece un conjunto de relaciones entre los sistemas de transiciones de ambas teorías, y entre sus términos dentro de ciertos marcos semánticos, para concluir que, en cierta manera, la teoría ecuacional de ACP es una extensión conservativa de la teoría ecuacional de CCS dentro del marco de los procesos finitos.

Abstract

This article presents a formal relation between the Robin Milner's CCS calculus [Mil80, Mil89] and the process algebras (ACP) mainly due to J. Bergstra y J. Klop [BK85, BW90]. Such a relation is given by a compositional function which maps CCS terms onto ACP terms. From this function we state several relations between the transition systems of both theories. We also find certain semantics relation between the set of terms. Finally we conclude that the equational theory of ACP is, in a certain manner, a conservative extension of the equational theory of CCS in the framework of finite processes.

1 Introducción

En este trabajo nos interesamos en la *Semántica de Sistemas Reactivos*. Los Sistemas Reactivos son aquellos que tienen como vocación principal no el cálculo de una cierta función de los datos de entrada, sino el mantener una cierta interacción con su ambiente. Ejemplos de este tipo de sistemas son las máquina vendedoras, los protocolos de comunicaciónes, los procesadores de textos, los equipos de fútbol y cada uno de sus jugadores. Uno de los tema centrales de la Semántica de Sistemas Reactivos es el representar y trabajar sobre la noción de *comportamiento de un sistema*, que llamamos *proceso* en este contexto.

Una aproximación común a este tema es la de construir los procesos a partir de una serie de procesos elementales y algunas formas de combinarlos para construir procesos mas complejos. Estas construcciones son las llamadas *álgebras de procesos* (tambien llamados *lenguajes de descripción de procesos*). Esta aproximación composicional tiene dos grandes ventajas. La primera de ellas es que la elección de los combinadores representa una intuición sobre las operaciones fundamentales en los sistemas reactivos y en algún sentido, de los lenguajes de programación que podríamos usar para

* *In Proc. of PANEL'94, XX Conf. Latinoamericana de Informática. Setember, 1994. Mexico.*

construirlos. De esta manera las álgebras de procesos son lenguajes de programación elementales donde aparecen solamente los elementos referentes al control y al comportamiento observable desde el exterior. La segunda es que las álgebras de procesos son verdaderas álgebras, provistas entonces de técnicas de prueba conocidas en matemáticas.

La multiplicidad de álgebras propuestas desde inicio de los '80 (CCS [Mil80], SCCS [Mil83], CSP [BHR84], ACP [BK85, BW90], CHOCS [Tho89], ATP [NRSV90], π calculus [MPW92]) motiva el estudio comparado de ellas. Esto nos permite comprender a un álgebra en términos de otra, y nos da también una manera de comprender mejor la intuición representada detras de cada una. En este terreno hay varias preguntas pertinentes. La primera de ellas sobre la expresividad: ¿cuáles son los modelos representables en un lenguaje dado? Otras se relacionan con los combinadores: ¿qué operación sobre los modelos representa un combinador dado? ¿cuál es la relacion entre diferentes combinadores de diferentes lenguajes?

Existen diversas formas de enfrentar estas preguntas. De manera arbitraria podemos caracterizarlas como más semánticos o más sintácticos. Entre las aproximaciónes mas semánticas podemos estudiar los modelos que aceptan las diferentes álgebras, como se hace en [Bro83]. Como todas ellas aceptan como modelos los Sistemas de Transiciones (ST), ésto nos da un territorio común donde relacionarlas. Nuestras preguntas del párrafo anterior se reducen a: ¿cuáles son los ST representables en un lenguaje? ¿qué operaciones en los ST representa cada combinador? ¿qué relación hay entonces entre las operaciones que representan combinadores de álgebras diferentes?

Un enfoque mas sintáctico (que seguimos aquí) es el de comparar diferentes lenguajes a partir de su signatura (sus combinadores), estudiando con detalle las Especificaciones de los Sistemas de Transiciones (TSS) [BIM88, GV89], que son utilizadas para dar semántica sobre los ST siguiendo la técnica introducida por Plotkin en [Plo81]. A nuestro entender, esta aproximación permite atacar con simplicidad y detalle una de las cuestiones planteadas: ¿cuál es la relación entre diferentes combinadores de diferentes lenguajes?

En este trabajo mostramos un caso de aplicación de las técnicas arriba descritas para el análisis de diferentes teorías en el caso particular de dos álgebras de procesos bien conocidas: CCS [Mil80, Mil89] y ACP [BK85, BW90].

Para establecer esta comparación presentamos y estudiamos con detalle una traducción composicional Φ de los términos finitos (sin recursión) de CCS sobre los términos (finitos) de ACP. Formalmente, $\Phi : \mathcal{T}_{\mathrm{CCS}} \rightarrow \mathcal{T}_{\mathrm{ACP}_\tau}$. Esta traducción se realiza sobre una instanciación particular de ACP, con un alfabeto de acciones y una función de comunicación adecuada.

Esta traducción es directa sobre los combinadores de suma no deterministica y renombre. Un poco mas delicados son los casos de la restricción y el prefijo, debido, en el primer caso, a que las acciones en CCS son vistas conjuntamente con sus *acciones complementarias*, mientras ACP las individualiza, y en el segundo, al hecho de disponer en ACP de una primitiva mas fuerte que el prefijo: la composicion secuencial. Finalmente, la operación de concurrencia merece particular atención, ya que en ambas teorías poseen comportamientos diferentes en el caso de establecer comunicación.

Desde el punto de vista de las TSS, la funcion Φ puede extenderse canónicamente de los terminos a los literales, las reglas de inferencia y a las pruebas de las transiciones. En este sentido, al definir Φ composicionalmente, hemos dado tambien una traducción de la TSS de CCS.

Mostramos aquí que las traducciones de las reglas de CCS son reglas derivables de las de ACP. Concluimos directamente que la imagen de todas las transiciones de un proceso P de CCS son transiciones de $\Phi(P)$. Vemos también que ACP es estrictamente mas grande que CCS, en el sentido que exiten términos de ACP que no son imagen de ningun término CCS. Inclusive restringiendonos a los términos que se encuentran en Φ(CCS) la TSS de ACP no es una extension conservativa de la de Φ(CCS): en algunos casos el término $\Phi(P)$ interviene en mas transiciones que las P. Esto último nos dice que, considerados como sistemas de transicion, P y $\Phi(P)$ no son isomorfos, y mostramos también ejemplos donde no son fuertemente bisimilares [Par81]. Hemos caracterizado una familia grande de $\mathcal{T}_{\text{CCS}}$ donde la Φ es una bisimulación, que incluye al conjunto de términos en los que no aparece el operador de concurrencia.

Para el caso general, para todo el conjunto términos CCS, probamos que Φ es una τ-bisimulacion enraizada [Mil80]. Esto nos dice que en la parte alcanzable por Φ, CCS y ACP difieren en algunos pasos invisibles, pero que estos pasos pueden encontrarse en posiciones no importantes dentro de la estructura arborescente.

Como corolario de lo anterior se prueba que todo par de término de CCS que son equivalentes en la teoría ecuacional de CCS, sus traducciones también lo son en la teoría ecuacional de ACP_τ y viceversa. Esto es, la teoría ecuacional de ACP_τ es en, algún sentido, una extensión conservativa de la de CCS.

La estructura de este trabajo es la siguiente: En la sección 2 recordamos las nociones básicas de Sistema de Transiciones, las equivalencias semánticas que van a utilizarse y las especificaciones de sistemas de transiciones. En la sección siguiente (3) se resumen CCS y ACP. La sección 4 contiene los aportes de este trabajo: la presentación y el estudio de la traducción composicional de CCS a ACP_τ. Esta parte se concluye con los resultados sobre las teorias ecuacionales. Las conclusiones de este trabajo son presentadas en la sección 5.

2 Procesos

2.1 Sistemas de transiciones

Es usual describir a los procesos por medio de Sistemas de Transiciones Etiquetadas (LTS) [Kel76].

Definición 2.1 (LTS) *Un* sistema de transiciones etiquetadas (LTS) *es una terna* $\mathbf{P} = (\mathcal{P}, \mathcal{A}, \rightarrow)$ *donde* $\mathcal{P} = \{P, Q, \ldots\}$ *es un conjunto de* estados*,* $\mathcal{A} = \{a, b, \ldots\}$ *es un conjunto de* acciones*, y* $\rightarrow \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{P}$ *es la* relación de transición*. Generalmente escribiremos* $P \xrightarrow{a} Q$ *en lugar de* $(P, a, Q) \in \rightarrow$*.*

De esta manera, $P \xrightarrow{a} Q$ significa que un sistema que está en un estado P evoluciona a otro estado Q realizando una acción a. Podemos ver que la relación de transición modela el comportamiento operacional de un sistema, por esto, este estilo de modelar el comportamiento es usualmente llamado *semántica operacional*. Un LTS es generalmente representado como un grafo dirigido con raíz y etiquetado en las aristas.

Consideraremos una acción especial $\tau \notin \mathcal{A}$ denominada *acción invisible* que representa el comportamiento interno de un proceso, por ejemplo, una computadora que acaba de aceptar un comando y está decidiendo qué hacer, o un hombre pensando. La idea es que el observador de un proceso no pueda distinguir fácilmente la ocurrencia de la acción invisible, y el "grado de detección" de ésta determina la variedad de equivalencias. Definimos $\mathcal{A}_\tau = \mathcal{A} \cup \{\tau\}$.

Definición 2.2 *Consideremos el LTS* $\mathbf{P} = (\mathcal{P}, \mathcal{A}_\tau, \rightarrow)$*.* $\Rightarrow \subseteq \mathcal{P} \times (\mathcal{A} \cup \{\epsilon\}) \times \mathcal{P}$ *está definida por:* $P \overset{\epsilon}{\Longrightarrow} Q$ *if* $P = Q$ *o si existen ciertos* $P_0, \ldots, P_n \in \mathcal{P}$ $(n > 0)$ *tal que* $P = P_0 \xrightarrow{\tau} P_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} P_n = Q$*, y* $P \overset{a}{\Longrightarrow} Q$ *si existen* R *y* R' *tal que* $P \overset{\epsilon}{\Longrightarrow} R \xrightarrow{a} R' \overset{\epsilon}{\Longrightarrow} Q$*.*

El operador $\hat{}$ *queda definido como* $\hat{a} = a$ *para todo* $a \neq \tau$ *y* $\hat{\tau} = \epsilon$*.*

Nótese que $P \xrightarrow{a} Q$ implica $P \overset{\hat{a}}{\Longrightarrow} Q$ pero que la recíproca no se cumple en general.

Generalmente necesitamos verificar si dos procesos son iguales de acuerdo a algún criterio, por ejemplo, cuando testeamos si un sistema se comporta como su especificación. Sin embargo, el criterio para "igualar" procesos parece no ser único. Así, nos encontramos con diferentes *equivalencias semánticas*. En este artículo trataremos con *semánticas de simulación* y *bisimulación*. La idea principal de ambas semánticas puede ser propuesta como un juego. Diremos que un sistema $\mathcal{S}$ puede simular a otro sistema

$\mathcal{S}'$ si cada acción que $\mathcal{S}'$ realiza puede ser tambien realizada por $\mathcal{S}$. $\mathcal{S}$ y $\mathcal{S}'$ son similares si pueden simularse mutuamente. La bisimulación es explicada por el mismo juego con una regla adicional: en cualquier momento, el rol de los sistemas $\mathcal{S}$ y $\mathcal{S}'$ pueden cambiar.

Definición 2.3 (Semántica de simulación) *Consideremos el LTS* $\mathbf{P} = (\mathcal{P}, \mathcal{A}, \rightarrow)$. *Una* simulación *es una relación* $S \subseteq \mathcal{P} \times \mathcal{P}$ *que satisface, para todo* $a \in \mathcal{A}$*:*

$$\text{si } PSQ \text{ y } P \xrightarrow{a} P', \text{ entonces } \exists Q' \in \mathcal{P} : Q \xrightarrow{a} Q' \text{ y } P'SQ'.$$

Un proceso P puede ser simulado por *otro proceso* Q *(notación* $P \underline{\subset} Q$*), si existe una simulación* S *tal que* PSQ. *Dos procesos* P *y* Q *son* similares *(notación* $P \rightleftharpoons Q$*), si* $P \underline{\subset} Q$ *y* $Q \underline{\subset} P$.

Definición 2.4 (Semántica de bisimulación (fuerte)) [Par81] *Consideremos el LTS* $\mathbf{P} = (\mathcal{P}, \mathcal{A}, \rightarrow)$. *Una relación* $S \subseteq \mathcal{P} \times \mathcal{P}$ *es una* bisimulación (fuerte) *si y solo si* S *y* S^{-1} *son simulaciones. Dos procesos* P *y* Q *son* (fuertemente) bisimilares *(notación* $P \leftrightarroweq Q$*), si existe una bisimulación* S *tal que* PSQ.

Definición 2.5 (Semántica de τ-bisimulación) [Mil80, Mil89] *Consideremos el LTS* $\mathbf{P} = (\mathcal{P}, \mathcal{A}_\tau, \rightarrow)$. *Una* τ-bisimulación *es una relación* $S \subseteq \mathcal{P} \times \mathcal{P}$ *que satisface, para todo* $a \in \mathcal{A}_\tau$*:*

1. *si* PSQ *y* $P \xrightarrow{a} P'$, *entonces* $\exists Q' \in \mathcal{P} : Q \xRightarrow{\hat{a}} Q'$ *y* $P'SQ'$*;*

2. *si* PSQ *y* $Q \xrightarrow{a} Q'$, *entonces* $\exists P' \in \mathcal{P} : P \xRightarrow{\hat{a}} P'$ *y* $P'SQ'$.

Dos procesos P *y* Q *son* τ-bisimilares *(notación* $P \leftrightarroweq_\tau Q$*), si existe una* τ*-bisimulación* S *tal que* PSQ.

Dos procesos P *y* Q *son* τ-bisimilares enraizados (rooted τ-bisimilars), *(notación* $P \leftrightarroweq_{r\tau} Q$*) si* $P \leftrightarroweq_\tau Q$ *y además:*

1. $P \xrightarrow{\tau} P'$ *implica* $\exists Q'', Q' : Q \xrightarrow{\tau} Q'' \xRightarrow{\epsilon} Q'$ *y* $P' \leftrightarroweq_\tau Q'$*;*

2. $Q \xrightarrow{\tau} Q'$ *implica* $\exists P'', P' : P \xrightarrow{\tau} P'' \xRightarrow{\epsilon} P'$ *y* $P' \leftrightarroweq_\tau Q'$.

Teorema 2.1 $\leftrightarroweq$ *es una equivalencia más fina que* $\rightleftharpoons$ *y que* $\leftrightarroweq_{r\tau}$. *Además* $\leftrightarroweq_{r\tau}$ *es más fina que* $\leftrightarroweq_\tau$.

2.2 Especificaciones de sistemas de transiciones

En la tarea de verificación, uno no acostumbra a demostrar propiedades sobre el sistema completo, sino sobre subsistemas de éste, esto es, sobre partes que serán insertadas en un cierto contexto. Entonces, necesitamos equivalencias que sean preservadas por contextos, necesitamos *congruencias*. Un contexto es una pieza de un sistema construido de cierta forma con un "hueco". Tal "cierta forma" es establecida por el *lenguaje* con el que estemos tratando, el cual puede ser dado por una *signatura*.

Definición 2.6 (Signaturas y términos) *Una* signatura *es una estructura* $\Sigma = (F, r)$ *donde* F *es un conjunto de* nombres de funciones *(u* operadores*), y* $r : F \rightarrow \mathbf{N}$ *es la* función rango *que da la aridad de los operadores. Un operador de aridad 0 se denomina* constante.

Sea $\mathcal{V}$ *un conjunto de* variables. *El conjunto de* (Σ-)términos *sobre* $\mathcal{V}$ *(notación* $\mathcal{T}(\Sigma, \mathcal{V})$*) es el menor conjunto que satisface:*

- $\mathcal{V} \subseteq \mathcal{T}(\Sigma, \mathcal{V})$;
- $\phi \in F$ *y* $t_1, \ldots, t_{r(\phi)} \in \mathcal{T}(\Sigma, \mathcal{V})$ *implica* $\phi(t_1, \ldots, t_{r(\phi)}) \in \mathcal{T}(\Sigma, \mathcal{V})$.

Los términos que no contienen variables se denominan cerrados. *Denotaremos con* $\mathcal{T}(\Sigma)$ *al conjunto de los términso cerrados.*

Definición 2.7 (Congruencia) *Una relación de equivalencia* $\cong \in \mathcal{T}(\Sigma) \times \mathcal{T}(\Sigma)$ *es una* relación de congruencia para Σ *si para todo* $\phi \in \Sigma$ *y todo término cerrado* $u_1, \ldots, u_{r(\phi)}, v_1, \ldots, v_{r(\phi)}$ *se tiene*

$$\forall i \in \{1, \ldots, r(\phi)\} : u_i \cong v_i \text{ implica } \phi(u_1, \ldots, u_{r(\phi)}) \cong \phi(v_1, \ldots, v_{r(\phi)})$$

En la sección previa, hemos visto como representar el comportamiento de un sistema por medio de LTSs. Tal sitema estará representado por un programa o una especificación (un término) en un dado lenguaje (una signatura). Entonces, un LTS puede dar la semántica de un término. Un método común para definir esta clase de sistemas de transiciones es por medio de *especificaciones de sistemas de transiciones*.

Definición 2.8 (TSS) [Plo81, BIM88, GV89] *Una* especificación de sistemas de transiciones (TSS) *es una terna* $\mathbb{T} = (\Sigma, \mathcal{A}, \mathcal{R})$*, donde* Σ *es una signatura,* $\mathcal{A}$ *es un conjunto de etiquetas o acciones y* $\mathcal{R}$ *es un conjunto de reglas de la forma:*

$$\frac{\{t_k \xrightarrow{a_k} t'_k | k \in K\}}{t \xrightarrow{a} t'}$$

siendo K *un conjunto índice,* $t_k, t'_k, t, t' \in \mathcal{T}(\Sigma, \mathcal{V})$ *y* $a_k, a \in \mathcal{A}$*. Las expresiones de la forma* $t \xrightarrow{a} t'$ *se denominan* literales. *En una regla* $r \in \mathcal{R}$*, los literales sobre la línea se llaman* premisas *de* r*, y el literal debajo de la línea,*conclusión *de* r*. Una regla se denomina* axioma *si su conjunto de premisas es vacío, y en tal caso escribiremos* $t \xrightarrow{a} t'$ *en lugar de* $\frac{\emptyset}{t \xrightarrow{a} t'}$.

En este marco, diremos que una transición $t \xrightarrow{a} t'$ es posible si ésta puede ser *probada* en $\mathbb{T}$.

Definición 2.9 (Prueba) *Sea* $\mathbb{T} = (\Sigma, \mathcal{A}, \mathcal{R})$ *un TSS. Una* prueba *para un literal* ξ *en* $\mathbb{T}$*(notación* $\mathbb{T} \vdash \xi$*) es un árbol invertido (que bifurca hacia arriba) bien fundado, donde sus nodos están etiquetados con literales* $t \xrightarrow{a} t'$ *siendo* $t, t' \in \mathcal{T}(\Sigma, \mathcal{V})$ *y* $a \in \mathcal{A}$*, tal que:*

- la raíz está etiquetada con ξ*,*

- si χ *es la etiqueta de un nodo* s *y* $\{\chi_k | k \in K\}$ *es el conjunto de etiquetas de los nodos inmediatamente sobre* s*, luego existe una regla* $\frac{\{\zeta_k | k \in K\}}{\zeta} \in \mathcal{R}$ *y una sustitución* $\theta : \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$ *tal que* $\chi = \theta(\zeta)$ *y* $\chi_k = \theta(\zeta_k)$*.*

Definición 2.10 (El LTS especificado por un TSS) *Sea* $\mathbb{T} = (\Sigma, \mathcal{A}, \mathcal{R})$ *un TSS. El LTS especificado por* $\mathbb{T}$*está dado por* $\mathbf{P} = (\mathcal{T}(\Sigma), \mathcal{A}, \to')$ *donde* $\to'$ *está definida por:* $t \xrightarrow{a}{}' t'$ *sii* $\mathbb{T} \vdash t \xrightarrow{a} t'$*. (Generalmente evitaremos diferenciaciones escribiendo* $\to$ *en lugar de* $\to'$*.)*

3 Las teorías de concurrencia

En esta sección introducimos las teorías de concurrencia que vamos a tratar.

En lo siguiente utilizaremos la siguiente nomenclatura: si T es una teoría, luego Σ_{T} es la signatura de T, $\mathcal{T}_{\mathrm{T}}$ su conjunto de términos, $\mathbb{T}_{\mathrm{T}}$ el TSS que define T, y así siguiendo. Además, para cualquier conjunto de símbolos $\{\gamma_1, \ldots, \gamma_m\}$ tal que $\{\gamma_1, \ldots, \gamma_m\} \cap \mathcal{A} = \emptyset$, definimos $\mathcal{A}_{\gamma_1, \ldots, \gamma_m} = \mathcal{A} \cup \{\gamma_1, \ldots, \gamma_m\}$.

3.1 CCS (Calculus of Communicating Systems)

Este cálculo fue introducido por Robin Milner en [Mil80]. El siguiente resumen fue adaptado de [Mil85, Mil89, Gla86].

3.1.1 Sintaxis

Sea $\mathbf{A}$ un conjunto de *nombres* ($\mathsf{a} \in \mathbf{A}$). Sea $\overline{\mathbf{A}}$ un conjunto de *co-nombres* ($\overline{\mathsf{a}} \in \overline{\mathbf{A}}$) tal que $\mathbf{A} \cap \overline{\mathbf{A}} = \emptyset$. Además $\mathbf{A}$ y $\overline{\mathbf{A}}$ son coordinables de acuerdo a la biyección $\mathsf{a} \mapsto \overline{\mathsf{a}}$. Diremos que $\overline{\mathsf{a}}$ es el *complemento* de a. Sea $\mathcal{A} = \mathbf{A} \cup \overline{\mathbf{A}}$. La noción de acción complementaria es extendida a $\mathcal{A}_\tau$ haciendo $\overline{\tau} = \tau$ y $\overline{\overline{a}} = a$ para todo $a \in \mathcal{A}_\tau$.

Sea $\mathfrak{R}_{\mathrm{CCS}}$ el conjunto de *funciones de reetiquetado*, esto es, de las funciones $f : \mathcal{A}_\tau \to \mathcal{A}_\tau$ tal que $f(\tau) = \tau$ y $f(\overline{a}) = \overline{f(a)}$.

La signatura de CCS, Σ_{CCS}, contiene la constante 0 (denominada inacción), los operadores unarios $a.$ (prefijo, definido para todo $a \in \mathcal{A}_\tau$), $\backslash X$ (restricción, definido para todo $X \subseteq \mathcal{A}$) y $[f]$ (renombrado, definido para todo $f \in \Re_{\mathrm{CCS}}$), más los operadores binarios $+$ (suma) y $|$ (composición).

Asumiremos que $a.$ asocia más fuerte que $|$ y a su vez $|$ asocia más fuerte que $+$.

3.1.2 Semántica operacional

La semántica operacional de los términos CCS está definida por el TSS $\mathbb{T}_{\mathrm{CCS}} = (\Sigma_{\mathrm{CCS}}, \mathcal{A}_\tau, \mathcal{R}_{\mathrm{CCS}})$ cuyas reglas son dadas en la Tabla 1, donde $a \in \mathcal{A}_\tau$.

Es bien conocido que tanto la bisimulacioń fuerte como τ-bisimulación enraizada son congruencias para todos los combinadores de CCS.

3.1.3 Teorías ecuacionales

En las tablas 2 y 3 damos dos teorías ecuacionales para CCS, donde $a \in \mathcal{A}_\tau$. Dado que $+$ es asociativo y conmutativo, escribiremos $\sum_{i \in \{1,\ldots,n\}} P_i$ por $P_1 + \cdots + P_n$. El conjunto de axiomas $\mathcal{E}_\tau$ se forma con los de $\mathcal{E}$ más los adicionales.

Los siguientes teoremas estabelecen un modelo completo y correcto para cada uno de los sistemas de axiomas para procesos finitos, esto es, para $\mathcal{T}_{\mathrm{CCS}}$.

Teorema 3.1 *Para todo $P, Q \in \mathcal{T}_{\mathrm{CCS}}$, $P \leftrightarroweq Q$ sii $\mathcal{E} \vdash P = Q$, esto es, $\mathcal{T}_{\mathrm{CCS}}/\leftrightarroweq\ \models \mathcal{E}$.*

Teorema 3.2 *Para todo $P, Q \in \mathcal{T}_{\mathrm{CCS}}$, $P \leftrightarroweq_{r\tau} Q$ sii $\mathcal{E}_\tau \vdash P = Q$, esto es, $\mathcal{T}_{\mathrm{CCS}}/\leftrightarroweq_{r\tau}\ \models \mathcal{E}_\tau$.*

3.2 ACP (Algebra of Communicating Processes)

Esta sección resume resultados aparecidos mayormente en [BK85, BW90]. Utilizaremos x, y, z como variables (en lugar de P, Q, R) para seguir la notación de la literatura. ACP_c es generalmente llamada ACP. Nosotros adoptamos este nombre para evitar confusión y poder usar ACP más generalmente.

3.2.1 Signatura

Sea $\mathcal{A}$ un conjunto de constantes que no contiene a δ y a τ. Sea $\Re_{\mathrm{ACP}}$ el conjunto de las *funciones de reetiquetado*, esto es, las funciones $f : \mathcal{A} \to \mathcal{A} \cup \{\tau, \delta\}$. La signatura contiene las constantes a (acciones atómicas, $a \in \mathcal{A}$), δ (deadlock o inacción) y τ (acción invisible), los operadores unarios ρ_f (renombrado, definido para toda $f \in \Re_{\mathrm{ACP}}$), ∂_H (encapsulamiento, definido para todo $H \subseteq \mathcal{A}$) y τ_I (abstracción, definido para todo $I \subseteq \mathcal{A}$), y los operadores binarios $\cdot$ (composición secuencial), $+$ (composición alternativa), $||$ (composición paralela o merge), $\lfloor\!\lfloor$ (merge de izquierda) y $|$ (merge de comunicación).

En general, consideraremos que $\cdot$ asocia más fuerte que $||$, $\lfloor\!\lfloor$ y $|$, y estos operadores lo hacen más fuerte que $+$. Así, $xy||z + uv$ significa $((x \cdot y)||z) + (u \cdot v)$.

3.2.2 Teorías ecuacionales

Una *función de comunicación* sobre $\mathcal{A}$ es una función parcial $\gamma : \mathcal{A} \times \mathcal{A} \rightharpoonup \mathcal{A}$ satisfaciendo conmutatividad y asociatividad. Si $\gamma(a, b)$ no está definida, para ciertos a, b, diremos que a y b *no comunican*. Una acción a tal que $a = \gamma(b, c)$ para ciertos b, c, se denomina *acción de comunicación*.

Las Tablas 4 y 5 muestran dos conjuntos de axiomas, donde $a, b \in \mathcal{A}_\delta$, $H, I \subseteq \mathcal{A}$ y $f \in \Re_{\mathrm{ACP}}$. Existe una consideración especial: τ y τ_I no pertenecen a Σ_{ACP_c}. Además, el conjunto de axiomas para ACP_τ está formado por los de ACP_c más los adicionales.

Notar que tanto ACP_c como ACP_τ son teorías parcialmente especificadas dado que dependen no sólo del conjunto de constantes $\mathcal{A}$, sino también de como se define γ.

Act	$a.P \xrightarrow{a} P$		
Sum$_1$	$\dfrac{P \xrightarrow{a} P'}{P+Q \xrightarrow{a} P'}$	**Sum$_2$**	$\dfrac{Q \xrightarrow{a} Q'}{P+Q \xrightarrow{a} Q'}$
Com$_1$	$\dfrac{P \xrightarrow{a} P'}{P\|Q \xrightarrow{a} P'\|Q}$	**Com$_2$**	$\dfrac{Q \xrightarrow{a} Q'}{P\|Q \xrightarrow{a} P\|Q'}$
Com$_3$	$\dfrac{P \xrightarrow{a} P' \;\; Q \xrightarrow{\overline{a}} Q'}{P\|Q \xrightarrow{\tau} P'\|Q'}\ (a \neq \tau)$		
Res	$\dfrac{P \xrightarrow{a} P'}{P\backslash X \xrightarrow{a} P'\backslash X}\ (a, \overline{a} \notin X)$	**Rel**	$\dfrac{P \xrightarrow{a} P'}{P[f] \xrightarrow{f(a)} P'[f]}$

Tabla 1: Semántica operacional de CCS

A1	$P+Q=Q+P$	Rs1	$0\backslash X = 0$
A2	$P+(Q+R)=(P+Q)+R$	Rs2	$(P+Q)\backslash X = P\backslash X + Q\backslash X$
A3	$P+P=P$	Rs3	$(a.P)\backslash X = (\overline{a}.P)\backslash X = 0$ if $a \in X$
A4	$P+0=P$	Rs4	$(a.P)\backslash X = a.(P\backslash X)$ if $a, \overline{a} \notin X$
Rl1	$0[f]=0$	Rl3	$(a.P)[f] = f(a).(P)[f]$
Rl2	$(P+Q)[f] = P[f]+Q[f]$		

A8 $$P|Q = \sum_{j\in J} a_j.(P_j|Q) \;+\; \sum_{k\in K} b_k.(P|Q_k) \;+\; \sum_{j\in J, k\in K, a_j=\overline{b_k}\neq\tau} \tau.(P_j|Q_k)$$

donde $P = \sum_{j\in J} a_j.P_j$ y $Q = \sum_{k\in K} b_k.Q_k$

Tabla 2: Axiomas de CCS para bisimulación fuerte: $\mathcal{E}$

A5	$a.\tau.P = a.P$
A6	$P+\tau.P = \tau.P$
A7	$a.(P+\tau.Q)+a.Q = a.(P+\tau.Q)$

Tabla 3: Axiomas adicionales para τ-bisimulación enraizada: $\mathcal{E}_\tau$

3.2.3 Los modelos de términos

ACP agrega la noción de *terminación satisfactoria*. Como vimos, el proceso de inacción (0 en CCS, δ en ACP) no puede realizar ninguna acción. Así, nada más puede ocurrir una vez que éste es alcanzado. Por otro lado, la terminación satisfactoria propone continuar ejecutando en la manera en que esto sea posible. De esta manera, un proceso debe informar que ha terminado satisfactoriamente y por ello necesitamos una nueva acción. Sea $\surd$ tal acción (léase *tick*). Esta es incluida no sólo en el conjunto de acciones sino también en el conjunto de estados. Entonces, al escribir $a \xrightarrow{a} \surd$ estamos diciendo que el proceso a puede realizar una acción a y terminar satisfactoriamente. Esto nos lleva a considerar $\surd$ como una nueva constante dentro de la signatura.

Definición 3.1 *El TSS para $\mathcal{T}_{\mathrm{ACP}_c}$ está definido por $\mathbb{T}_{\mathrm{ACP}_c} = (\Sigma_{\mathrm{ACP}_c} \cup \{\surd\}, \mathcal{A}_{\surd}, \mathcal{R}_{\mathrm{ACP}_c})$, donde $\mathcal{R}_{\mathrm{ACP}_c}$ está dado en las Tablas 6 y 7 con $a, b, c \in \mathcal{A}$.*

Definición 3.2 [D'A94] *El TSS para $\mathcal{T}_{\mathrm{ACP}_\tau}$ está definido por $\mathbb{T}_{\mathrm{ACP}_\tau} = (\Sigma_{\mathrm{ACP}_\tau} \cup \{\surd\}, \mathcal{A}_{\tau\surd}, \mathcal{R}_{\mathrm{ACP}_\tau})$, donde $\mathcal{R}_{\mathrm{ACP}_\tau}$ está dado por $\mathcal{R}_{\mathrm{ACP}_c}$ definido en $a, b, c \in \mathcal{A}_\tau$ exceptuando las reglas M5 a M7 y Cm1 a Cm3, más las reglas dadas en la Tabla 8.*

Los siguientes teoremas establecen los modelos para las teorías ecuacionales dadas anteriormente.

Teorema 3.3 *Para todo $x, y \in \mathcal{T}_{\mathrm{ACP}_c}$, $x \underline{\leftrightarrow}\, y$ sii $\mathrm{ACP}_c \vdash x = y$, esto es, $\mathcal{T}_{\mathrm{ACP}_c}/\underline{\leftrightarrow} \models \mathrm{ACP}_c$.*

Teorema 3.4 [D'A94] *Para todo $x, y \in \mathcal{T}_{\mathrm{ACP}_\tau}$, $x \underline{\leftrightarrow}_{r\tau}\, y$ sii $\mathrm{ACP}_\tau \vdash x = y$, esto es, $\mathcal{T}_{\mathrm{ACP}_\tau}/\underline{\leftrightarrow}_{r\tau} \models \mathrm{ACP}_\tau$.*

A1	$x + y = y + x$		CM1	$x \| y = x \mathbin{\|\!\lfloor} y + y \mathbin{\|\!\lfloor} x + x \mid y$
A2	$x + (y + z) = (x + y) + z$		CM2	$a \mathbin{\|\!\lfloor} x = ax$
A3	$x + x = x$		CM3	$ax \mathbin{\|\!\lfloor} y = a(x \| y)$
A4	$(x + y)z = xz + yz$		CM4	$(x + y) \mathbin{\|\!\lfloor} z = x \mathbin{\|\!\lfloor} z + y \mathbin{\|\!\lfloor} z$
A5	$(xy)z = x(yz)$		CM5	$(ax \mid b) = (a \mid b) \cdot x$
A6	$x + \delta = x$		CM6	$(a \mid bx) = (a \mid b) \cdot x$
A7	$\delta x = \delta$		CM7	$(ax \mid by) = (a \mid b) \cdot (x \| y)$
			CM8	$(x + y) \mid z = x \mid z + y \mid z$
CF1	$a \mid b = \gamma(a, b)$	si γ está definida	CM9	$x \mid (y + z) = x \mid y + x \mid z$
CF2	$a \mid b = \delta$	en caso contrario		
D1	$\partial_H(a) = a$	si $a \notin H$	RN1	$\rho_f(\delta) = \delta$
D2	$\partial_H(a) = \delta$	si $a \in H$	RN2	$\rho_f(a) = f(a)$ si $a \neq \delta$
D3	$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$		RN3	$\rho_f(x + y) = \rho_f(x) + \rho_f(y)$
D4	$\partial_H(xy) = \partial_H(x) \cdot \partial_H(y)$		RN4	$\rho_f(xy) = \rho_f(x) \cdot \rho_f(y)$

Tabla 4: Axiomas para ACP_c

D0	$\partial_H(\tau) = \tau$		TM1	$\tau \lfloor\!\lfloor x = \tau x$
RN0	$\rho_f(\tau) = \tau$		TM2	$\tau x \lfloor\!\lfloor y = \tau(x\|\|y)$
TI0	$\tau_I(\tau) = \tau$		TC1	$\tau \| x = \delta$
TI1	$\tau_I(a) = a$	si $a \notin I$	TC2	$x \| \tau = \delta$
TI2	$\tau_I(a) = \tau$	si $a \in I$	TC3	$\tau x \| y = x \| y$
TI3	$\tau_I(x+y) = \tau_I(x) + \tau_I(y)$		TC4	$x \| \tau y = x \| y$
TI4	$\tau_I(xy) = \tau_I(x) \cdot \tau_I(y)$			
T1	$x\tau = x$		T3	$a(\tau x + y) = a(\tau x + y) + ax$
T2	$\tau x = \tau x + x$			

Tabla 5: Axiomas adicionales para ACP_τ

Act	$a \xrightarrow{a} \surd$	End	$\surd \xrightarrow{\surd} \delta$
Alt1	$\dfrac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'}$	Alt2	$\dfrac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$
Sq1	$\dfrac{x \xrightarrow{a} x'}{xy \xrightarrow{a} x'y}\ (x' \neq \surd)$	Sq2	$\dfrac{x \xrightarrow{a} \surd}{xy \xrightarrow{a} y}$
M1	$\dfrac{x \xrightarrow{a} x'}{x\|\|y \xrightarrow{a} x'\|\|y}\ (x' \neq \surd)$	M2	$\dfrac{y \xrightarrow{a} y'}{x\|\|y \xrightarrow{a} x\|\|y'}\ (y' \neq \surd)$
M3	$\dfrac{x \xrightarrow{a} \surd}{x\|\|y \xrightarrow{a} y}$	M4	$\dfrac{y \xrightarrow{a} \surd}{x\|\|y \xrightarrow{a} x}$
M5	$\dfrac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{x\|\|y \xrightarrow{c} x'\|\|y'}\ (\gamma(a,b) = c,\ x', y' \notin \{\surd\})$		
M6	$\dfrac{x \xrightarrow{a} \surd \quad y \xrightarrow{b} y'}{x\|\|y \xrightarrow{c} y'}\ (\gamma(a,b) = c)$	M7	$\dfrac{x \xrightarrow{a} x' \quad y \xrightarrow{b} \surd}{x\|\|y \xrightarrow{c} x'}\ (\gamma(a,b) = c)$
Lm1	$\dfrac{x \xrightarrow{a} x'}{x \lfloor\!\lfloor y \xrightarrow{a} x'\|\|y}\ (x' \neq \surd)$	Lm2	$\dfrac{x \xrightarrow{a} \surd}{x \lfloor\!\lfloor y \xrightarrow{a} y}$

Tabla 6: $\mathcal{R}_{\mathrm{ACP}_c}$

Cm1	$\dfrac{x \xrightarrow{a} x' \;\; y \xrightarrow{b} y'}{x \mid y \xrightarrow{c} x' \Vert y'} \;\; (\gamma(a,b)=c,\; x', y' \notin \{\surd\})$		
Cm2	$\dfrac{x \xrightarrow{a} \surd \;\; y \xrightarrow{b} y'}{x \mid y \xrightarrow{c} y'} \; (\gamma(a,b)=c)$	Cm3	$\dfrac{x \xrightarrow{a} x' \;\; y \xrightarrow{b} \surd}{x \mid y \xrightarrow{c} x'} \; (\gamma(a,b)=c)$
Rel1	$\dfrac{x \xrightarrow{a} x'}{\rho_f(x) \xrightarrow{f(a)} \rho_f(x')} \; (x' \neq \surd)$	Rel2	$\dfrac{x \xrightarrow{a} \surd}{\rho_f(x) \xrightarrow{f(a)} \surd}$
Enc1	$\dfrac{x \xrightarrow{a} x'}{\partial_H(x) \xrightarrow{a} \partial_H(x')} \;\; (a \notin H,\; x' \neq \surd)$	Enc2	$\dfrac{x \xrightarrow{a} \surd}{\partial_H(x) \xrightarrow{a} \surd} \; (a \notin H)$

Tabla 7: $\mathcal{R}_{\mathrm{ACP}_c}$ (continuación)

Abs1	$\dfrac{x \xrightarrow{a} x'}{\tau_I(x) \xrightarrow{a} \tau_I(x')} \;\; (a \notin I,\; x' \neq \surd)$	Abs2	$\dfrac{x \xrightarrow{a} x'}{\tau_I(x) \xrightarrow{\tau} \tau_I(x')} \;\; (a \in I,\; x' \neq \surd)$
Abs3	$\dfrac{x \xrightarrow{a} \surd}{\tau_I(x) \xrightarrow{a} \surd} \; (a \notin I)$	Abs4	$\dfrac{x \xrightarrow{a} \surd}{\tau_I(x) \xrightarrow{\tau} \surd} \; (a \in I)$
M5	$\dfrac{x \overset{\epsilon}{\Longrightarrow} x'' \;\; y \overset{\epsilon}{\Longrightarrow} y'' \;\; x'' \xrightarrow{a} x' \;\; y'' \xrightarrow{b} y'}{x \Vert y \xrightarrow{c} x' \Vert y'}$	$(\gamma(a,b)=c,$ $x', y', x'', y'' \notin \{\surd\})$	
M6	$\dfrac{x \overset{\epsilon}{\Longrightarrow} x'' \;\; y \overset{\epsilon}{\Longrightarrow} y'' \;\; x'' \xrightarrow{a} \surd \;\; y'' \xrightarrow{b} y'}{x \Vert y \xrightarrow{c} y'}$	$(\gamma(a,b)=c,$ $x'', y'' \notin \{\surd\})$	
M7	$\dfrac{x \overset{\epsilon}{\Longrightarrow} x'' \;\; y \overset{\epsilon}{\Longrightarrow} y'' \;\; x'' \xrightarrow{a} x' \;\; y'' \xrightarrow{b} \surd}{x \Vert y \xrightarrow{c} x'}$	$(\gamma(a,b)=c,$ $x'', y'' \notin \{\surd\})$	
Cm1	$\dfrac{x \overset{\epsilon}{\Longrightarrow} x'' \;\; y \overset{\epsilon}{\Longrightarrow} y'' \;\; x'' \xrightarrow{a} x' \;\; y'' \xrightarrow{b} y'}{x \mid y \xrightarrow{c} x' \Vert y'}$	$(\gamma(a,b)=c,$ $x', y', x'', y'' \notin \{\surd\})$	
Cm2	$\dfrac{x \overset{\epsilon}{\Longrightarrow} x'' \;\; y \overset{\epsilon}{\Longrightarrow} y'' \;\; x'' \xrightarrow{a} \surd \;\; y'' \xrightarrow{b} y'}{x \mid y \xrightarrow{c} y'}$	$(\gamma(a,b)=c,$ $x'', y'' \notin \{\surd\})$	
Cm3	$\dfrac{x \overset{\epsilon}{\Longrightarrow} x'' \;\; y \overset{\epsilon}{\Longrightarrow} y'' \;\; x'' \xrightarrow{a} x' \;\; y'' \xrightarrow{b} \surd}{x \mid y \xrightarrow{c} x'}$	$(\gamma(a,b)=c,$ $x'', y'' \notin \{\surd\})$	

Tabla 8: Reglas adicionales para $\mathcal{R}_{\mathrm{ACP}_\tau}$

4 Traducción de CCS en ACP_τ

La relación entre ambas teorías es establecida mediante una función composicional $\Phi : \mathcal{T}_{\mathrm{CCS}} \to \mathcal{T}_{\mathrm{ACP}_\tau}$.

Mostraremos que $\mathbb{T}_{\mathrm{CCS}}$ está "inserto" en $\mathbb{T}_{\mathrm{ACP}_\tau}$ utilizando Φ. Esto nos indicará que, para todo $P \in \mathcal{T}_{\mathrm{CCS}}$, si P puede realizar una acción a, también podrá hacerlo $\Phi(P)$. Veremos también que la recíproca no se cumple y estudiaremos este caso.

Los resultados mostrados a partir de ahora han sido completamente demostrados en [D'A94].

4.1 La función Φ

Como ya hemos visto, ACP_τ no es una teoría completa, ésta depende del conjunto de constantes $\mathcal{A}$ y de la función de comunicación γ. Estableceremos entonces la teoría completa con la que trabajaremos definiendo $\mathcal{A}$ y γ.

Nota 4.1 *Consideraremos la teoría ACP_τ donde el conjunto de constantes $\mathcal{A}$ es el mismo conjunto $\mathcal{A} = \mathbf{A} \cup \overline{\mathbf{A}}$ utilizado en CCS, y $\gamma : \mathcal{A} \times \mathcal{A} \rightharpoonup \{\tau\}$ está definida por $\gamma(a, \overline{a}) = \gamma(\overline{a}, a) = \tau$ (e indefinida en cualquier otro caso).*

En la subsección 3.2 pedíamos que $\gamma : \mathcal{A} \times \mathcal{A} \rightharpoonup \mathcal{A}$ que contradice lo anterior. Al mismo tiempo considerábamos la posibilidad de que más de dos procesos se comuniquen. Esto no ocurre en este caso y entonces γ está bien definida. (Note que cumple asociatividad y conmutatividad.)

Definición 4.1 *Definimos $\Phi : \mathcal{T}_{\mathrm{CCS}} \to \mathcal{T}_{\mathrm{ACP}_\tau}$ inductivamente de acuerdo a la estructura de los términos de $\mathcal{T}_{\mathrm{CCS}}$ como:*

$$\begin{array}{ll}
\Phi(0) \stackrel{\mathrm{def}}{=} \delta & \Phi(P|Q) \stackrel{\mathrm{def}}{=} \Phi(P)||\Phi(Q) \\
\Phi(a.P) \stackrel{\mathrm{def}}{=} a \cdot \Phi(P) & \Phi(P\backslash X) \stackrel{\mathrm{def}}{=} \partial_H(\Phi(P)), \quad \textit{donde } H = \{a | a \in X \vee \overline{a} \in X\} \\
\Phi(P + Q) \stackrel{\mathrm{def}}{=} \Phi(P) + \Phi(Q) & \Phi(P[f]) \stackrel{\mathrm{def}}{=} \rho_f(\Phi(P))
\end{array}$$

Nos abusaremos de la notación y diremos que $H = X \cup \overline{X}$.

De la definición se sigue inmediatamente que Φ está bien definida. Consideremos el proceso a ($\in \mathcal{T}_{\mathrm{ACP}_\tau}$). Está claro que no existe $P \in \mathcal{T}_{\mathrm{CCS}}$ tal que $\Phi(P) = a$; esto implica que Φ no es suryectiva. Por otro lado, puede verificarse simplemente que Φ es inyectiva haciendo inducción estructural.

Φ puede extenderse trivalmente a literales haciendo $\Phi(P \xrightarrow{a} Q) = \Phi(P) \xrightarrow{a} \Phi(Q)$. Análogamente se extiende Φ a reglas y conjunto de reglas.

4.2 Relación entre el comportamiento operacional de $\mathcal{T}_{\mathrm{CCS}}$ y $\mathcal{T}_{\mathrm{ACP}_\tau}$

En esta sección relacionaremos ambos sistemas de transiciones, esto es, $\mathbf{P}_{\mathrm{CCS}}$ y $\mathbf{P}_{\mathrm{ACP}_\tau}$.

Definición 4.2 *Definimos el TSS $\Phi(\mathbb{T}_{\mathrm{CCS}}) = (\Sigma_{\mathrm{ACP}_\tau}, \mathcal{A}_\tau, \Phi(\mathcal{R}_{\mathrm{CCS}}))$. (Ver Tabla 9.)*

Los siguientes lemas son fácilmente demostrables:

Lema 4.1 *Sean $P, Q \in \mathcal{T}_{\mathrm{CCS}}$. Luego $\mathbb{T}_{\mathrm{CCS}} \vdash P \xrightarrow{a} Q$ sii $\Phi(\mathbb{T}_{\mathrm{CCS}}) \vdash \Phi(P) \xrightarrow{a} \Phi(Q)$.*

Lema 4.2 *Toda regla de $\Phi(\mathcal{R}_{\mathrm{CCS}})$ es demostrable en $\mathbb{T}_{\mathrm{ACP}_\tau}$. Además, para todo $x, y \in \mathcal{T}_{\mathrm{ACP}_\tau}$, $\Phi(\mathbb{T}_{\mathrm{CCS}}) \vdash x \xrightarrow{a} y$ implica $\mathbb{T}_{\mathrm{ACP}_\tau} \vdash x \xrightarrow{a} y$.*

Demostración: Ver Tabla 9. La segunda parte sigue inmediatamente. ∎

Corolario 4.2.1 *Sean $P, Q \in \mathcal{T}_{\mathrm{CCS}}$. Luego $\mathbb{T}_{\mathrm{CCS}} \vdash P \xrightarrow{a} Q$ implica $\mathbb{T}_{\mathrm{ACP}_\tau} \vdash \Phi(P) \xrightarrow{a} \Phi(Q)$.*

Corolario 4.2.2 *Sean $P, Q \in \mathcal{T}_{\mathrm{CCS}}$. Luego $\forall a \in \mathcal{A} \cup \{\epsilon\} :\ P \stackrel{a}{\Longrightarrow} Q$ implica $\Phi(P) \stackrel{a}{\Longrightarrow} \Phi(Q)$.*

$\Phi(\mathbf{Act})$	$a \cdot x \xrightarrow{a} x$		
$\Phi(\mathbf{Sum_1})$	$\equiv$ Alt1 con $x' \neq \surd$	$\Phi(\mathbf{Sum_2})$	$\equiv$ Alt2 con $x' \neq \surd$
$\Phi(\mathbf{Com_1})$	$\equiv$ M1	$\Phi(\mathbf{Com_2})$	$\equiv$ M2
$\Phi(\mathbf{Com_3})$	$\dfrac{x \xrightarrow{a} x' \; y \xrightarrow{\overline{a}} y'}{x\|\|y \xrightarrow{\tau} x'\|\|y'} \; (a \neq \tau)$		
$\Phi(\mathbf{Res})$	$\equiv$ Enc1 con $H = X \cup \overline{X}$	$\Phi(\mathbf{Rel})$	$\equiv$ Rel1 con $f \in \Re_{\mathrm{CCS}}$

Tabla 9: $\Phi(\mathcal{R}_{\mathrm{CCS}})$

Demostración: Se sigue inmediatamente de la definición de $\Rightarrow$ y el corolario anterior. ∎

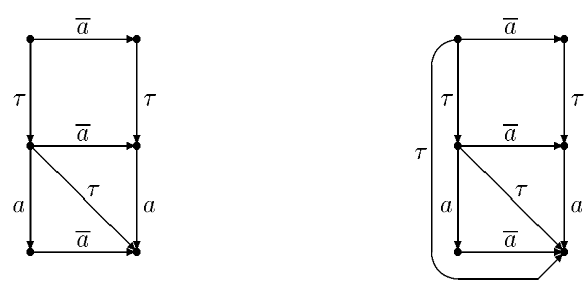
Lema 4.3 *No existe $P \in \mathcal{T}_{\mathrm{CCS}}$ tal que $\Phi(P) \xrightarrow{a} \surd$ en $\mathbf{P}_{\mathrm{ACP}_\tau}$.*

Demostración: Ver Corolario 4.6.1 (y Lema 4.6). ∎

Este lema establece que no hay ningún proceso definido en CCS que pueda terminar (satisfactoriamente). Esto comienza a mostrar diferencias entre CCS y ACP. Entonces nos preguntamos si, siendo $P \in \mathcal{T}_{\mathrm{CCS}}$, para toda transición $\Phi(P) \xrightarrow{a} x$ en $\mathbf{P}_{\mathrm{ACP}_\tau}$ existe una respectiva $P \xrightarrow{a} Q$ con $\Phi(Q) \equiv x$ en $\mathbf{P}_{\mathrm{CCS}}$. La siguiente nota nos responde negativamente.

Nota 4.2 *En general no se cumple que $\Phi(P) \xrightarrow{a} \Phi(Q)$ implique que $P \xrightarrow{a} Q$. Por ejemplo, consideremos el término CCS $\tau.a.0|\overline{a}.0$. Por la definición de Φ, tenemos que $\Phi(\tau.a.0|\overline{a}.0) \stackrel{\mathrm{def}}{=} \tau a\delta||\overline{a}\delta$. Por la regla M5, $\tau a\delta||\overline{a}\delta \xrightarrow{\tau} \delta||\delta$, mientras no es cierto que $\tau.a.0|\overline{a}.0 \xrightarrow{\tau} 0|0$. Sin embargo, $\tau.a.0|\overline{a}.0 \xrightarrow{\tau\tau} 0|0$, lo que implica que $\tau.a.0|\overline{a}.0 \stackrel{\epsilon}{\Longrightarrow} 0|0$. De esta manera, la traducción puede introducir nuevas transiciones. Este ejemplo es clarificado en la Figura 1.*

Esta clase de transiciones (esto es, las obtenidas después de comunicar dos procesos donde al menos uno de ellos comienza realizando un τ) es la única que hace la diferencia. Para el siguiente lema vamos



$\tau.a.0|\overline{a}.0$ $\qquad\qquad$ $\tau a\delta||\overline{a}\delta$

Figura 1: Φ introduce nuevas transiciones a considerar la comunicación en ACP_τ sólo cuando los procesos involucrados se dispongan a realizar las acciones complementarias. Para ello consideremos la regla

$$\mathrm{M5}' \qquad \frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{x||y \xrightarrow{c} x'||y'} \ \ (\gamma(a,b) = c \ \wedge \ x', y' \notin \{\surd\})$$

la cual es claramente demostrable en $\mathbb{T}_{\mathrm{ACP}_\tau}$. Sea $\mathbb{T}^M_{\mathrm{ACP}_\tau}$ el TSS obtenido de $\mathbb{T}_{\mathrm{ACP}_\tau}$ cambiando la regla M5 por la regla M5$'$. Es fácil observar que $\mathbb{T}_{\mathrm{ACP}_\tau}$ permite mas transiciones que $\mathbb{T}^M_{\mathrm{ACP}_\tau}$. Entonces podemos decir:

Lema 4.4 *Sea $P \in \mathcal{T}_{\mathrm{CCS}}$. Si $\mathbb{T}^M_{\mathrm{ACP}_\tau} \vdash \Phi(P) \xrightarrow{a} x$, entonces existe $Q \in \mathcal{T}_{\mathrm{CCS}}$ tal que $\Phi(Q) \equiv x$ y $\mathbb{T}_{\mathrm{CCS}} \vdash P \xrightarrow{a} Q$.*

Demostración: Aplicaremos inducción estructural sobre P. (Obviamente, no consideraremos los casos donde $\Phi(P) \xrightarrow{a} \surd$ porque estos nunca ocurren por Lema 4.3.) Demostraremos el caso en que $P \equiv Q|R$, el resto queda en manos del lector.

Digamos que $\Phi(Q|R) \stackrel{\mathrm{def}}{=} \Phi(Q)||\Phi(R) \xrightarrow{a} x$.

Supongamos que ésta proviene de aplicar la regla M1, entonces $\Phi(Q) \xrightarrow{a} y$ y $x \equiv y||\Phi(R)$. Por hipótesis inductiva $\exists Q' : \ \Phi(Q') \equiv y$ y $Q \xrightarrow{a} Q'$. Aplicando la regla $\mathbf{Com}_1$, $Q|R \xrightarrow{a} Q'|R$. Además $x \equiv \Phi(Q'|R)$ por definición de Φ.

El caso en que M2 es aplicada es similar.

Supongamos que M5$'$ es aplicada. Luego, para algún b, $\Phi(Q) \xrightarrow{b} y$ y $\Phi(R) \xrightarrow{\overline{b}} z$; además $x \equiv y||z$ y $a \equiv \tau$ (por γ). Por inducción, $\exists Q' : \ \Phi(Q') \equiv y$ y $Q \xrightarrow{b} Q'$, y $\exists R' : \ \Phi(R') \equiv z$ y $R \xrightarrow{\overline{b}} R'$. Finalmente, $x \equiv \Phi(Q')||\Phi(R') \stackrel{\mathrm{def}}{=} \Phi(Q'|R')$ y, por $\mathbf{Com}_3$, $Q|R \xrightarrow{\tau} Q'|R'$. ■

Nota 4.3 *Existen ciertos casos patológicos que satisfacen el Lema 4.4 aún si M5 es considerada; por ejemplo, $\tau.a.0|\overline{a}.0 + \tau.(0|0)$ o bien $\tau.a.0|\overline{a}.0 + a.0|\overline{a}.0$.*

A partir de ahora, cada vez que escribamos algo como $P \stackrel{\sigma}{\Longrightarrow} \xrightarrow{b} \xrightarrow{a} \stackrel{\rho}{\Longrightarrow} Q$, querremos significar que existen ciertos P', P'', P''' tal que $P \stackrel{\sigma}{\Longrightarrow} P' \xrightarrow{b} P'' \xrightarrow{a} P''' \stackrel{\rho}{\Longrightarrow} Q$.

Lema 4.5 *Sea $P, Q \in \mathcal{T}_{\mathrm{CCS}}$. Si $P \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{a} P'$ y $Q \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{b} Q'$, luego*

1. $P + Q \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{a} P'$
2. $P + Q \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{b} Q'$
3. $P\backslash X \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{a} P'\backslash X$, *si* $a \notin X \ \wedge \ \overline{a} \notin X$
4. $P[f] \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{f(a)} P'$
5. $P|Q \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{a} P'|Q$
6. $P|Q \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{b} P|Q'$
7. $P|Q \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{\tau} P'|Q'$, *si* $a = \overline{a}$

Demostración: Todos los casos se demuestran por inducción en la definición de $\stackrel{\epsilon}{\Longrightarrow}$ excepto el item 7 que se demuestra a partir de 5 y 6. ■

Reformularemos el Lema 4.4 de manera de considerar el TSS $\mathbb{T}_{\mathrm{ACP}_\tau}$.

Lema 4.6 *Sea $P \in \mathcal{T}_{\mathrm{CCS}}$. Si $\Phi(P) \xrightarrow{a} x$ en $\mathbf{P}_{\mathrm{ACP}_\tau}$, luego $x \not\equiv \surd$ y existe un $Q \in \mathcal{T}_{\mathrm{CCS}}$ tal que $\Phi(Q) \equiv x$ y $P \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{a} Q$ en $\mathbf{P}_{\mathrm{CCS}}$.*

Demostración: Por inducción estructural en P. Nuevamente, sólo consideraremos el caso $P \equiv Q|R$.

Supongamos que $\Phi(Q|R) \stackrel{\mathrm{def}}{=} \Phi(Q)||\Phi(R) \xrightarrow{a} x$.

Supongamos que ésta proviene de aplicar la regla M1, entonces $\Phi(Q) \xrightarrow{a} y$, $y \not\equiv \surd$ y $x \equiv y||\Phi(R)$. Por hipótesis inductiva $\exists Q' : \ \Phi(Q') \equiv y$ y $Q \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{a} Q'$. Por Lema 4.5(5), $Q|R \stackrel{\epsilon}{\Longrightarrow} \xrightarrow{a} Q'|R$. Además $x \equiv \Phi(Q'|R)$ por definición de Φ.

El caso en que la regla M2 es considerada es análogo.

Las reglas M3 y M4 contradicen la hipótesis inductiva.

Supongamos que M5 es aplicada. Luego, para algún b, $\Phi(Q) \overset{\epsilon}{\Longrightarrow} y' \xrightarrow{b} y$ y $\Phi(R) \overset{\epsilon}{\Longrightarrow} z' \xrightarrow{\bar{b}} z$. Además $x \equiv y||z$ y $a \equiv \tau$ (por definición de γ).

Por definición de $\overset{\epsilon}{\Longrightarrow}$, existen $y_0, \ldots, y_n, n \leq 0$ tal que $\Phi(Q) \equiv y_0 \xrightarrow{\tau} y_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} y_n \equiv y' \xrightarrow{b} y$ (Recordemos que estamos tratando con procesos finitos, entonces n es un número finito.) Ahora podemos aplicar la hipótesis inductiva sucesivamente, esto es, $\exists Q_1 : \Phi(Q_1) \equiv y_1$ y $Q \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} Q_1$, luego $\exists Q_2 : \Phi(Q_2) \equiv y_2$ y $Q_1 \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} Q_2$, y así continuando hasta llegar a que $\exists Q' : \Phi(Q') \equiv y$ y $Q_n \overset{\epsilon}{\Longrightarrow} \xrightarrow{b} Q'$. Entonces $Q \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} Q_1 \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} \cdots \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} Q_n \overset{\epsilon}{\Longrightarrow} \xrightarrow{b} Q'$, esto es, $Q \overset{\epsilon}{\Longrightarrow} \xrightarrow{b} Q'$.

Análogamente $\exists R' : \Phi(R') \equiv z$ y $R \overset{\epsilon}{\Longrightarrow} \xrightarrow{\bar{b}} R'$.

Finalmente, por Lema 4.5(7), $Q|R \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} Q'|R'$ donde $\Phi(Q'|R') \overset{\text{def}}{=} \Phi(Q')||\Phi(R') \equiv y||z \equiv x$.

No es simple, pero utilizando la mismas técnicas que para demostrar el caso de M5, podemos ver que M6 y M7 contradicen la hipótesis inductiva. ∎

Corolario 4.6.1 (Lema 4.3) *No existe $P \in \mathcal{T}_{\text{CCS}}$ tal que $\Phi(P) \xrightarrow{a} \surd$ en $\mathbf{P}_{\text{ACP}_\tau}$.*

Corolario 4.6.2 *Para todo $a \in \mathcal{A} \cup \{\epsilon\}$, $\Phi(P) \overset{a}{\Longrightarrow} x$ implica $\exists Q \in \mathcal{T}_{\text{CCS}} : \Phi(Q) \equiv x \wedge P \overset{a}{\Longrightarrow} Q$.*

Demostración: Se sigue inmediatamente de la definición de $\Rightarrow$ y el Lema 4.6. ∎

4.3 Relación entre $\mathcal{T}_{\text{CCS}}$ y $\mathcal{T}_{\text{ACP}_\tau}$ en las distintas semánticas

Los principales conceptos en la comparación entre CCS y ACP$_\tau$ son expuestos en esta sección. Se dan un conjunto de teoremas estableciendo relaciones semánticas entre $\mathcal{T}_{\text{CCS}}$ y $\mathcal{T}_{\text{ACP}_\tau}$. De hecho, veremos que la función Φ es por sí misma muchas de esas relaciones.

Recordemos que una función es una relación. A partir de ahora, consideraremos a Φ como una relación, esto es, $\Phi \in \mathcal{T}_{\text{CCS}} \times \mathcal{T}_{\text{ACP}_\tau}$, y escribiremos $P \Phi x$ por $\Phi(P) = x$.

Utilizaremos la notación $R : \mathcal{P} \underset{\rightarrow}{\subseteq} \mathcal{P}'$ para decir que $R \subseteq \mathcal{P} \times \mathcal{P}'$ es una simulación de términos de $\mathcal{P}$ en términos de $\mathcal{P}'$, y análogamente para las otras equivalencias.

Teorema 4.7 *Para todo $P \in \mathcal{T}_{\text{CCS}}$, $P \underset{\rightarrow}{\subseteq} \Phi(P)$. Más aún, $\Phi : \mathcal{T}_{\text{CCS}} \underset{\rightarrow}{\subseteq} \mathcal{T}_{\text{ACP}_\tau}$.*

Demostración: Inmediato por Corolario 4.2.2. ∎

Nota 4.4 *El ejemplo dado en la Nota 4.2 deja claro que Φ^{-1} no es una simulación de términos de $\mathcal{T}_{\text{ACP}_\tau}$ en términos de $\mathcal{T}_{\text{CCS}}$. Esto implica que Φ no es una bisimulación en $\mathcal{T}_{\text{CCS}} \times \mathcal{T}_{\text{ACP}_\tau}$.*

A pesar de esta nota, podemos obtener un subconjunto de términos CCS en el cual se cumpla que Φ sea una bisimulación. Como vimos en la Nota 4.2, el problema es generado por el operador de concurrencia y sólo en ciertos casos de comunicación, más precisamente al aplicar una instancia de M5 distinta de M5$'$ en la prueba de una transición. Sea $\mathcal{T}_{\text{CCS}\backslash\text{M}} = \{P | P \in \mathcal{T}_{\text{CCS}} \wedge$ no existe derivativa de $\Phi(P)$ probada utilizando una instancia de M5 distinta de M5$'$ $\}$. Luego el siguiente teorema se satisface:

Teorema 4.8 *Para todo $P \in \mathcal{T}_{\text{CCS}\backslash\text{M}}$, $\Phi(P) \underset{\rightarrow}{\subseteq} P$. Más aún, $\Phi^{-1} : \mathcal{T}_{\text{ACP}_\tau} \underset{\rightarrow}{\subseteq} \mathcal{T}_{\text{CCS}\backslash\text{M}}$.*

Corolario 4.8.1 *Para todo $P \in \mathcal{T}_{\text{CCS}\backslash\text{M}}$, $P \leftrightarrow \Phi(P)$. Más aún, $\Phi : \mathcal{T}_{\text{CCS}\backslash\text{M}} \leftrightarrow \mathcal{T}_{\text{ACP}_\tau}$.*

Hasta ahora sólo hemos considerado los procesos concretos (la acción invisible es considerada "tan observable" como el resto de las acciones). Bajo abstracción tenemos nuevas consideraciones.

Teorema 4.9 *Para todo $P \in \mathcal{T}_{\mathrm{CCS}}$, $P \underline{\leftrightarrow}_\tau \Phi(P)$. Más aún, $\Phi : \mathcal{T}_{\mathrm{CCS}} \underline{\leftrightarrow}_\tau \mathcal{T}_{\mathrm{ACP}_\tau}$.*

Demostración: Supongamos que $P \Phi x$.

1. Sea $P \xrightarrow{a} Q$. Dado que $\xrightarrow{a}$ implica $\overset{\hat{a}}{\Longrightarrow}$ para todo a, y considerando el Teorema 4.7, tenemos que $\exists y : x \xrightarrow{a} y$ y $Q \Phi y$.

2. Supongamos que $x \xrightarrow{a} y$. Por el Lema 4.6, existe Q tal que $P \overset{\epsilon}{\Longrightarrow} \xrightarrow{a} Q$ y $Q \Phi y$. Pero $P \overset{\epsilon}{\Longrightarrow} \xrightarrow{a} Q$ implica $P \overset{\hat{a}}{\Longrightarrow} Q$, lo que concluye la demostración. ∎

Teorema 4.10 *Para todo $P \in \mathcal{T}_{\mathrm{CCS}}$, $P \underline{\leftrightarrow}_{r\tau} \Phi(P)$.*

Demostración: Por el Teorema 4.9, $P \underline{\leftrightarrow}_\tau \Phi(P)$. Sólo nos queda demostrar que se cumplen las condiciones sobre la raíz (ver Definición 2.5).

Supongamos que $P \xrightarrow{\tau} Q$. Luego, por el Corolario 4.2.1, $\Phi(P) \xrightarrow{\tau} \Phi(Q)$. Entonces $\Phi(P) \xrightarrow{\tau} \overset{\epsilon}{\Longrightarrow} \Phi(Q)$ y además $Q \underline{\leftrightarrow}_\tau \Phi(Q)$ por el Teorema 4.9.

Supongamos ahora que $\Phi(P) \xrightarrow{\tau} y$. Por el Lema 4.6, existe Q such that $P \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} Q$ and $\Phi(Q) \equiv y$. Si $P \overset{\epsilon}{\Longrightarrow} P \xrightarrow{\tau} Q$, luego $P \xrightarrow{\tau} Q \overset{\epsilon}{\Longrightarrow} Q$. En otro caso, existe P' tal que $P \xrightarrow{\tau} P' \overset{\epsilon}{\Longrightarrow} \xrightarrow{\tau} Q$ que implica $P \xrightarrow{\tau} P' \overset{\epsilon}{\Longrightarrow} Q$. Como $Q \underline{\leftrightarrow}_\tau \Phi(Q) \equiv y$ por el Teorema 4.9, la prueba queda concluida. ∎

El siguiente teorema establece que la teoría ecuacional ACP_τ es una extensión conservativa vía Φ de $\mathcal{E}_\tau$, la teoría ecuacional de CCS para τ-bisimulación enraizada.

Teorema 4.11 *Sean $P, Q \in \mathcal{T}_{\mathrm{CCS}}$. Luego $\mathcal{E}_\tau \vdash P = Q$ sii $\mathrm{ACP}_\tau \vdash \Phi(P) = \Phi(Q)$.*

Demostración:

$$
\begin{array}{llll}
\mathcal{E}_\tau \vdash P = Q & \text{iff} & P \underline{\leftrightarrow}_{r\tau} Q & \text{(Teorema 3.2)} \\
 & \text{iff} & \Phi(P) \underline{\leftrightarrow}_{r\tau} P \underline{\leftrightarrow}_{r\tau} Q \underline{\leftrightarrow}_{r\tau} \Phi(Q) & \text{(Teorema 4.10)} \\
 & \text{iff} & \Phi(P) \underline{\leftrightarrow}_{r\tau} \Phi(Q) & (\underline{\leftrightarrow}_{r\tau} \text{ es transitiva}) \\
 & \text{iff} & \mathrm{ACP}_\tau \vdash \Phi(P) = \Phi(Q) & \text{(Teorema 3.4)}
\end{array}
$$

∎

5 Discusión y conclusión

Hemos presentado una traducción composicional Φ de los términos de CCS a los términos de ACP. Ella nos ha permitido comparar ambas teorías en tres aspectos: qué correspondencia hay entre los operadores de ambas teorías, bajo qué criterios una teoría se adecua a otra y cuán expresiva es una teoría respecto de la otra.

Debido a la restricción impuesta sobre γ, podemos ver que la comunicación en ACP es mucho más poderosa que en CCS. Mientras ACP permite que muchos procesos compuestos en concurrencia se sincronicen en una única acción de comunicación (esto depende de la definición de γ), CCS sólo permite dos. CCS restringe la comunicación a acciones complementadas la cual es ocultada al "emitir" τ como acción de comunicación.

En la concurrencia, no solo la comunicación es un caso de estudio. Si sólo consideráramos el subconjunto de términos de CCS que no contienen el operador de concurrencia, veríamos que cada uno de ellos es fuertemente bisimilar a un término de ACP_τ (esto es implicado por el Teorema 4.8). Sin embargo, como se viera en la Nota 4.2, la composición paralela introduce más transiciones en ACP_τ que en CCS, determinando que la bisimulación fuerte no se preserve. En este caso la adecuación está dada por la τ-bisimulación enraizada; esto es porque todo término CCS es τ-bisimilar enraizado a algún término ACP_τ.

Como vimos oportunamente, los términos de ACP que terminan satisfactoriamente no son representables en CCS. Pero podemos intuir más: todo término de ACP que *no realice una secuencia de acciones y termine satisfactoriamente* es representable en CCS. Un poco más formalmente podemos establecer la siguiente conjetura:

> *Para todo $x \in \mathcal{T}(\mathrm{ACP}_\tau)$ tal que no exista una secuencia de derivación $x \xrightarrow{a_1} \cdots \xrightarrow{a_n} \surd$, entonces existe un $P \in \mathcal{T}(\mathrm{CCS})$ tal que, para algún $y \in \mathcal{T}(\mathrm{ACP}_\tau)$, $ACP_\tau \vdash x = y$ y $\Phi(P) \equiv y$.*

Esta conjetura no sería válida si pretendieramos que $\Phi(P) \equiv x$. (Considerar como ejemplo $x \equiv (a + b) \cdot \delta$.)

Si bien la composición secuencial no existe en CCS, ésta puede ser definida a partir del operador $|$ y considerando una acción especial que refleje la noción de terminación satisfactoria (ver [Mil89]). Sin embargo, en determinados casos no se comporta de la forma esperada.

CCS no posee un operador de abstracción (τ_I en ACP_τ). Una definición más restringida fue dada en [Gla86] (abstrae tanto una acción como su complementaria).

Ambas teorías concuerdan en que el no-determinismo no debe ser una propiedad de los operadores (como ocurre en CSP [BHR84, Hoa85]), sino de los procesos involucrados: $\tau.P + \tau.Q$ representa una elección no-determinística entre los procesos P y Q, mientras $a.P+b.Q$ es absolutamente determinístico.

En [D'A94] se obtienen conclusiones semejantes a las obtenidas en este artículo para las teorías ACP y CCS sobre bisimulación fuerte y bisimulación de bifurcación [GW89]. [Bro83] establece una relación entre CCS y CSP sobre un dominio de árboles en el modelo de semántica de falla (failure semantics [BHR84]). [Gla86] da una axiomatización conjunta de CCS y CSP. En [D'A94] se establece una relación entre ACP y CSP en el estilo seguido por este artículo.

Agradecimientos.

Deseamos agradecer a Javier Blanco que nos ha animado a trabajar en este tema y ha corregido algunos de los primeros y numerosos borradores de este trabajo. Tambien agradecemos a Cecilia Pertino su calidez y colaboración en los últimos tiempos de la elaboración de este trabajo. Debemos reconocer, además, los aportes de Daniel Yankelevich, Luis Sierra y Gabriel Fialco en exposiciones previas.

Durante la primera parte de la este trabajo el primero de los autores trabajó en el Laboratorio de Investigación y Desarrollo en Informática del Departamento de Informática, Fac. de Ciencias Exactas, Universidad Nacional de La Plata.

Este trabajo fue financiado parcialmente por el "Programa de Desarrollo de Ciencias Básicas" de Uruguay, y por la Facultad de Ciencias Exactas de la Universidad de La Plata a través del proyecto "Una herramienta para la manipulación de Especificaciones en Semántica Operacional Estructurada".

Referencias

[BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, July 1984.

[BIM88] B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced: preliminary report. In *Conference Record of the 15th ACM Symposium on Principles of Programming Languages*, pages 229–239, California, San Diego, 1988.

[BK85] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, (37):77–121, 1985.

[Bro83] S.D. Brookes. On the relationship of CCS an CSP. In J. Diaz, editor, *Proceedings 10th ICALP*, pages 83–96, Barcelona, 1983. LNCS 154, Springer-Verlag.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process algebra.* Cambridge University Press, 1990.

[D'A94] Pedro R. D'Argenio. A comparative analisys of concurrency theories. Graduation thesis, Departamento de Informática, Facultad de Ciencias Exactas, Universidad Nacional de La Plata, May 1994.

[Gla86] R.J. van Glabbeek. Notes on the metodology of CCS and CSP. Report CS-R8624, Centrum voor Wiskunde en Informatica, Amsterdam, August 1986.

[GV89] J.F. Groote and F.W. Vaandrager. Structured operational semantics and bisimulation as a congruence (extended abstract). In G. Ausiello, M. Dezani-Ciancaglini, and S. Ronchi Della Rocca, editors, *Proceedings ICALP 89*, pages 423–438, Stressa, 1989. LNCS 372, Springer-Verlag.

[GW89] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics (extended abstract). In G.X. Ritter, editor, *Information Processing 89*, pages 613–618. North-Holland, 1989.

[Hoa85] C.A.R. Hoare. *Communicating sequential process.* Prentice Hall, 1985.

[Kel76] R.M. Keller. Formal verification of parallel programs. *Communications of the ACM*, 8(19):371–384, 1976.

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

[Mil83] Robin Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, (25):267–310, 1983.

[Mil85] Robin Milner. Lectures notes on a calculus for communicating systems. In M Broy, editor, *Control flow and data flow*, pages 205–228. International Summer School at Marktoberdorf, Springer-Verlag, 1985.

[Mil89] Robin Milner. *Communication and concurrency.* Prentice Hall, 1989.

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

[NRSV90] X. Nicollin, J.-L. Richier, J. Sifakis, and J. Voiron. ATP: an algebra for timed processes. In M. Broy and C.B. Jones, editors, *Proceedings of the IFIP TC 2 Working Conference on Programming Concepts and Methods*, Sea of Gallilee,Israel, April 1990. North-Holland.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequence. In P. Deussen, editor, *Proceedings 5th. GI Conference*, pages 167–183. LNCS 104, Springer-Verlag, 1981.

[Plo81] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI-FN-19, Computer Science Department, University of Århus, 1981.

[Tho89] Bent Thomsen. A calculus of higher order communicating systems. In *Proceedings 16th POPL*, pages 143–154, 1989.