24

2y

Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

TEORIA ALGEBRAICA DE PROCESOS BAJO EL MODELO DEL CALCULO DE SISTEMAS COMUNICANTES

TESIS
Que para obtener el título de
MATEMATICO
Presenta
Joaquín Fernando Mendoza Blanco

Ciudad Universitaria, Julio de 1991.

FALLA DE ORIGEN

INDICE

Prefacio.

En contraposición con los sistemas más tradicionales en ambientes secuenciales, el entendimiento de una aplicación concurrente y/o paralela se logra mediante una percepción relativista del sistema en la cual las dimensiones de espacio y tiempo juegan un papel esencial en el comportamiento visible hacia el exterior.

En años recientes, las técnicas y los modelos orientados al tratamiento concurrente y paralelo de problemas destacan principalmente tres aspectos básicos, a saber, la descomposición del problema en subproblemas utilizando algún tipo de estrategia, el manejo de la complejidad como una medida de la eficacia de la descomposición y el control de las comunicaciones entre las diferentes subtareas.

El diseño, análisis y razonamiento de estrategias para dividir un problema en pequeños subproblemas que puedan ser resueltos en forma concurrente y/o paralela, a partir de los cuales pueda construirse una solución al problema original, necesita de herramientas formales de expresión en las que puedan ser manifestados conceptos como el no determinismo, la sincronización, la observación, la comunicación, la concurrencia, el paralelismo, el manejo de la complejidad,etc.

La división de un problema complejo en diversas subtareas obliga a establecer cierta coordinación entre ellas, la cual se puede lograr mediante canales de comunicación interconectando a las diferentes tareas.

Entre las herramientas formales y semi-formales existentes para el estudio de la concurrencia y el paralelismo destacan las redes de Petri, la teoría de trazas, los árboles de sincronización, los sistemas de transición-estado, CCS,CSP, álgebras de procesos, etc. La diversidad de modelos y lenguajes es consecuencia del hecho de que cada uno captura y define los conceptos de proceso y concurrencia en distintas formas y en diversos niveles de abstracción.

El lenguaje especializado en esta clase de sistemas y aquí presentado es denominado Cálculo de Sistemas Comunicantes (CCS).

En otro orden de ideas, una forma de aproximarse a la comprensión de los sistemas concurrentes y paralelos es mediante el análisis de problemas específicos con el fin de detectar los elementos fundamentales que son abarcados por este tipo de sistemas.

A raíz del estudio de soluciones algorítmicas de algunos problemas básicos de concurrencia, surgió en mí el incentivo y motivación para asimilar herramientas formales en las cuales se puediera dar un tratamiento adecuado a los sistemas rebasando las barreras de la secuencialidad.He dirigido un especial interés en la aplicación de CCS para la expresión y desarrollo de problemas tradicionales de concurrencia y paralelismo, también he enfocado la atención en la demostración de propiedades básicas y avanzadas del lenguaje CCS, así como de algunos resultados que facilitan la denotación y manipulación de los sistemas concurrentes y paralelos mediante el tratamiento algebraico provisto por CCS.

Quiero además aprovechar estas líneas para agradecer profundamente a mí maestro y amigo, Dr. Víctor Germán Sánchez, su dirección para la elaboración de esta tésis. También agradezco sinceramente a todos mis asesores, Dra. Hanna Oktaba, M.C. Sergio Castro, Mat. Salvador López y Mat. Carlos Velarde por la enseñanza y amistad que me ofrecen.

1. Introducción.

La necesidad de procesar computacionalmente con mayor velocidad y mayor eficiencia es una de las características de muchos problemas prácticos y de investigación en la actualidad. Una forma básica de acelerar el procesamiento es mediante el paralelismo y la concurrencia. Lo anterior se puede lograr tanto físicamente a través de la utilización de múltiples procesadores como a nivel lógico estructurando a los sistemas en forma de procesos que cooperan para lograr algún objetivo definido. De hecho, es posible establecer dos categorías en las que se puede clasificar la naturaleza de las tareas computacionales: la primera de procesos mutuamente independientes (o no interactuantes) y la segunda de procesos mutuamente dependientes.

Para diseñar , analizar y razonar sobre un sistema particular existe un problema fundamental de orden el cual es referido como sincronización de procesos, y para enfrentarlo se utilizan dos métodos diferentes :

1) Variables compartidas.

En este método la sincronización y la comunicación se logra utilizando una memoria común con un control centralizado.

2) Paso de mensajes.

En este método los procesos intercambian mensajes (envían y reciben) sin compartir datos y son controlados mediante operaciones de espera y retardo.

En el pasado se han desarrollado diferentes modelos formales de computación como lo constituyen los autómatas, los lenguajes formales, las funciones recursivas, etc.,pero existe una necesidad de extender estos modelos y desarrollar algunos nuevos para comprender los efectos que se producen al acoplar las diversas acciones en las que participan los distintos procesos y de integrar las nociones del no determinismo al diseño y a la programación.

Dotar a los lenguajes con mecanismos para controlar la concurrencia y el paralelismo ha sido un objetivo principal en la investigación en años recientes, pero la programación constituye sólo un nivel en el cual se pueden abstraer las ideas y materializar soluciones (algunas veces en forma no muy clara y aceptable) de los problemas prácticos que deben ser resueltos. En general es preferible establecer las ideas, los criterios y las decisiones sobre las posibles soluciones a un problema en un nivel de abstracción más alto al impuesto por un lenguaje de programación, el cual en muchas formas limita la expresividad y la libertad de pensamiento para desarrollar soluciones. En limpio , existe una necesidad real de crear modelos que faciliten el diseño y el análisis de diversos sistemas que pueden y/o deben ser enfocados a través de una perspectiva paralela o concurrente.

En otro orden de ideas,como resultado de la división lógica de un sistema en varios procesos, las acciones en las que participa cada uno de ellos se pueden entrelazar o combinar de diversas formas, siendo una cualidad indispensable para un modelo del paralelismo tener la habilidad de condicionar las combinaciones de eventos y de eliminar aquéllas que no conducen a solución alguna.

Básicamente el problema anterior es una cuestión de orden , para lo cual se requieren modelos formales que posibiliten la descripción de condiciones tales como la sincronización, la comunicación , el no determinismo, etc.

En general, hay dos formas distintas en que se puede considerar como correcto un comportamiento compuesto por una colección de procesos :

i) Si existe una secuencia de eventos que produce un resultado satisfaciendo ciertas condiciones que lo acreditan como solución.

ii) Si el resultado de todas las combinaciones de eventos permitidas es el mismo, el cual es considerado como solución al problema.

En cualquiera de las perspectivas anteriores, existe la necesidad de establecer equivalencias entre las distintas descripciones que se pueden plantear para expresar un problema y su solución. Existe en CCS una noción de equivalencia observacional que identifica a los procesos cuyas comunicaciones externas siguen un mismo patrón pero cuyos comportamientos internos pueden ser distintos.

Una de las metas en el diseño de CCS fué desarrollar propiedades estructurales que permitieran componer sistemas a partir de componentes básicos interactuantes.

El cálculo de sistemas comunicantes constituye un modelo matemático (en el cual pueden ser expresados con claridad muchos sistemas concurrentes) que consiste de dos tipos de objetos, el primero constituído por agentes o procesos con la capacidad de realizar y participar en eventos , y el segundo compuesto por un conjunto de operadores (estáticos y dinámicos) para construir y estructurar a los procesos.

Para denotar a los eventos o acciones en los que participan los procesos, se considera una colección numerable de etiquetas $\{\alpha,\beta,\gamma,..\}$ ya que de hecho el modelo de CCS no tiene especial interés en la naturaleza misma de los eventos que constituyen a los procesos, pero sí en el orden y la composición de esos eventos.

El desarrollo de un cálculo nuevo requiere y se beneficia de algunas propiedades en su diseño como son el poseer una flexibilidad de manipulación, tener un alto nivel de articulación, dotarse de riqueza de expresión y equiparse con las capacidades necesarias para poder describir sistemas y argumentar matemáticamente sobre ellos.

El cálculo de sistemas comunicantes desarrollado por Robin Milner durante los años de 1979 y 1980, sobre cuyo estudio versará este trabajo , está basado en dos ideas centrales :

a) Observación.

El objetivo de la observación es la descripción de los sistemas concurrentes para determinar el comportamiento que será visto o experimentado por un observador externo.

b) Comunicación de valores.

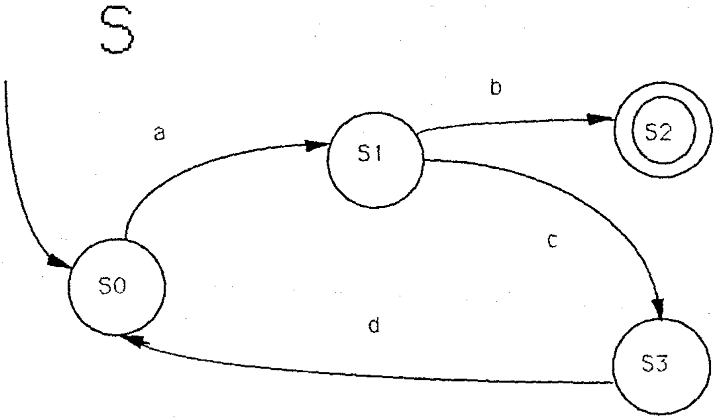
En general, un sistema se construye mediante agentes independientes interactuando mediante una acción indivisible que es la comunicación.

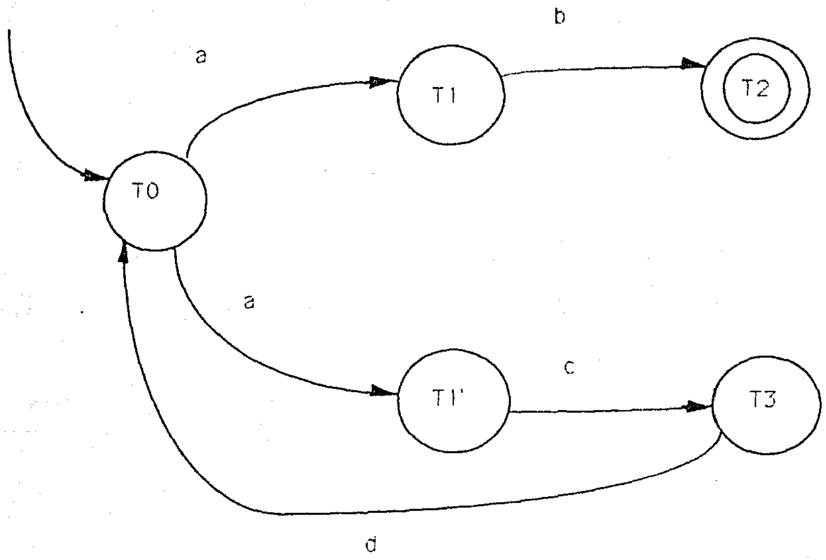
El cálculo de sistemas comunicantes CCS está enmarcado en un ambiente algebraico, siendo la operación central del álgebra de sistemas la composición concurrente. De hecho, los términos del cálculo representan programas cuya estructura se refleja en la estructura del término; también es conveniente mencionar que los términos del cálculo representan comportamientos de sistemas referidos como extensiones.

El cálculo CCS opera bajo la suposición de que en la composición de dos agentes, las acciones independientes de cada uno pueden ocurrir en cualquier orden pero no simultáneamente, esto es, un observador externo que vea el sistema sólo puede realizar una observación a la vez.

Para comenzar con algunos rudimentos técnicos se consideran algunos experimentos sobre máquinas no determinísticas.

Dados los autómatas S y T sobre el alfabeto $\Sigma = \{ a,b,c,d \}$, se presenta la siguiente proposición :





Proposición 1. Los autómatas S y T, con estados iniciales S_0 y T_0 respectivamente, son equivalentes en el sentido de que reconocen el mismo lenguaje, a saber, el denotado por la expresión regular (acd)*abε.

Demostración.

Denotemos también con S_i al lenguaje reconocido por el autómata S tomando como estado inicial a S_i y análogamente para el autómata T. De la figura obtenemos

$S0 = aS1$ $\qquad$ $T0 = aT1 + aT1'$

$S1 = bS2 + cS3$ $\qquad$ $T1 = bT2$

$S2 = \varepsilon$ $\qquad$ $T1' = cT3$

$S3 = dS0$ $\qquad$ $T2 = \varepsilon$

$T3 = dT0$

en donde el símbolo + denota la unión de lenguajes y el símbolo ε representa al lenguaje $\{ \varepsilon \}$ cuyo único elemento es la cuerda nula,también simbolizado por ε. Por substitución obtenemos

$$S0 = a (b\varepsilon \ + \ cdS0) = ab\varepsilon + acdS0$$

$$T0 = ab\varepsilon + acdT0$$

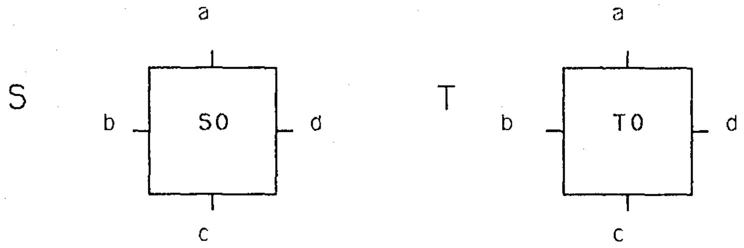
Utilizando la regla de Arden, se sabe que la solución a cualquiera de estas ecuaciones es única, precisamente $(acd)^*ab\varepsilon$, lo cual demuestra el resultado ofrecido .

Como siguiente paso , representamos a los autómatas S y T mediante cajas negras equipadas con 4 botones a los que corresponden sendos experimentos atómicos consistiendo en la opresión de los botones. Los posibles resultados de la opresión de un botón pueden ser :

éxito : se presenta cuando el botón está desbloqueado y se puede oprimir,

fracaso : se presenta cuando el botón está bloqueado.

Por ejemplo, en el estado inicial, sólo se puede oprimir exitosamente el botón a, los otros se encuentran bloqueados.



Consideremos la siguiente definición de equivalencia desde el punto de vista observacional.

Definición.

Los autómatas S y T son equivalentes observacionalmente si la aceptación de cualquier secuencia de opresiones de botones en el autómata S, partiendo del estado inicial, también es aceptada por el autómata T y viceversa.

Bajo está definición analizamos si los autómatas S y T son equivalentes, para lo cuál se ofrece una discusión breve sobre el comportamiento que presentan estos autómatas en particular.

Dados los comportamientos iniciales de las cajas S y T, regidos por los autómatas respectivos, del hecho de que S siempre acepta la secuencia ab y el autómata T no siempre la acepta, no se puede concluír la no equivalencia observacional de las cajas (debido al no determinismo de los autómatas, el significado de la palabra "siempre" envuelve la idea de un análisis exhaustivo de casos posibles ,este análisis puede ser basado en la suposición de que los comportamientos de S y T están regulados por los autómatas correspondientes).

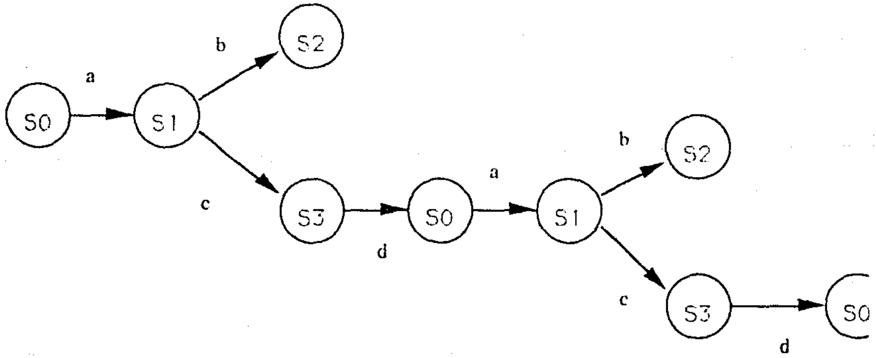
Sin el apoyo de un análisis exhaustivo de casos posibles, el éxito obtenido en una caja y el fracaso obtenido en la otra no es un hecho concluyente para aseverar la no equivalencia observacional.

Para distinguir observacionalmente a las cajas S y T, se aprecia que si se ocultaran los botones etiquetados con la letra b en ambas cajas, entonces nunca sería posible bloquear completamente a la caja S , a diferencia del comportamiento que presenta la caja T.

El comportamiento no determinista de una caja negra se resuelve invariable e irreversiblemente por información interna contenida en la caja .

Nota: La equivalencia observacional definida anteriormente rechaza la ley distributiva a(x+y) = ax + ay , propiedad que ha sido utilizada en la demostración de la proposición 1.

La representación del comportamiento de una caja negra se realiza mediante un árbol de comportamiento etiquetado con símbolos de un conjunto Λ, obtenido a través del desdoblamiento de la gráfica de transición de estados de la siguiente forma :



Λ = { a,b,c,d,...}

A continuación se introducen algunos conceptos básicos.

Definición.

Una etiqueta es un miembro de un conjunto prefijado Λ y se utilizan para denotar acciones o eventos.

Definición.

Un género L es cualquier subconjunto de Λ .

Definición.

Un árbol de sincronización rígido de género L (elemento del conjunto RST) es un árbol no ordenado y finitamente ramificado cuyos arcos están etiquetados por elementos de L.

Para los autómatas no determinísticos se emplea una transición usualmente etiquetada por ε y que está asociada con la cuerda nula. En el caso de las cajas negras, esta etiqueta ε se traduce en una acción interna y no observable que puede hacer cambiar el comportamiento de la caja. El símbolo τ denota este nuevo evento.

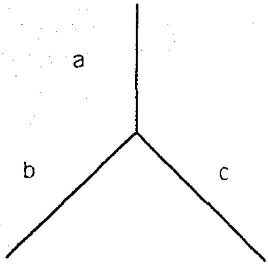
Definición.

Un árbol de sincronización de género L (elemento de ST) es un árbol no ordenado y finitamente ramificado cuyos arcos están etiquetados por elementos de $L \cup \{ \tau \}$.

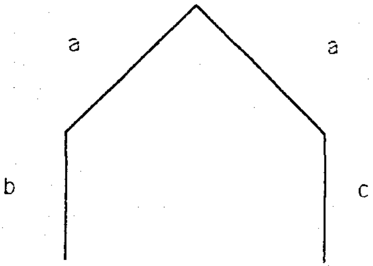
Nota : $\tau \notin \Lambda$, RST $\subsetneq$ ST.

Nota: Un árbol ST es rígido cuando todas sus etiquetas corresponden a acciones observables.

Los árboles de sincronización se asocian en forma natural con expresiones de comportamiento como se ilustra a continuación.



a (bNIL + cNIL)



abNIL + acNIL

• NIL

Notación.

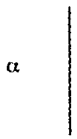
RST_L denota al conjunto de RST 's de género L.

$NIL \in RST_L \qquad \forall L \subseteq \Lambda.$

$+ \quad \in RST_L \times RST_M \rightarrow RST_{L \cup M}$

$\lambda \quad \in RST_L \rightarrow RST_{L \cup \{\lambda\}}$

En relación al concepto de equivalencia observacional sobre árboles de sincronización, se aprecia que los siguientes dos son gráficamente distintos pero observacionalmente equivalentes .



α $\qquad\qquad$ $\alpha + \alpha$

Una definición precisa de la equivalencia observacional se dará en las próximas secciones.

2. Sincronización.

Con el propósito de obtener una notación conveniente para expresar la interacción entre varios comportamientos, se selecciona un conjunto Λ con una estructura especial.

Definición.

$\Lambda = \Delta \cup \overline{\Delta}$ con $\Delta \cap \overline{\Delta} = \Phi$

Δ es el conjunto de nombres

$\overline{\Delta}$ es el conjunto de co - nombres

y $\overline{(\;)} : \Lambda \rightarrow \Lambda$ es una biyección en Λ, dada por

$\alpha \quad (\in \Delta) \mapsto \overline{\alpha} \quad (\in \overline{\Delta})$

$\overline{\alpha} \quad (\in \overline{\Delta}) \mapsto \alpha \quad (\in \Delta) \qquad (\overline{\overline{\alpha}} = \alpha)$

α y $\overline{\alpha}$ se llaman etiquetas complementarias .

Definición .

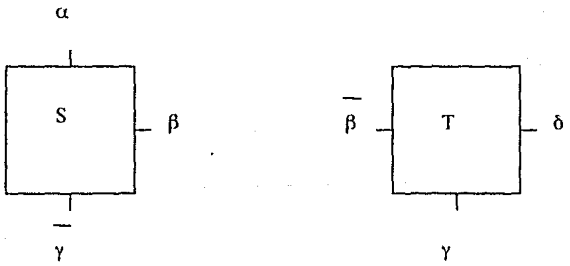
nombre ( ) : $\Lambda \rightarrow \Delta$

nombre (α) = nombre $(\overline{\alpha})$ = α

y para un género L, $\overline{L} = \{ \overline{\lambda} : \lambda \in L \}$,

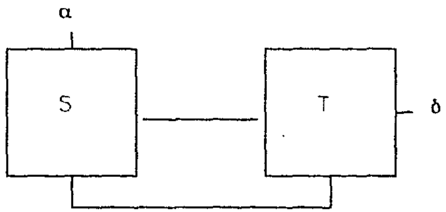
nombre (L) = $\{$nombre$(\lambda) : \lambda \in L \}$.

Considere el siguiente par de cajas negras,



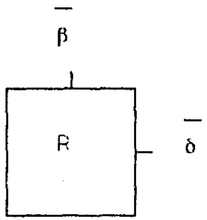
S : $\{ \alpha, \beta, \overline{\gamma} \}$ T : $\{ \overline{\beta}, \gamma, \delta \}$

Un resultado natural para la composición de S y T (denotada S || T), es ligar los puertos complementarios ocultando sus respectivas etiquetas.



S || T

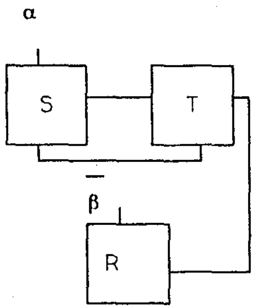
Esta operación de composición no es asociativa,como lo prueba el siguiente contraejemplo :

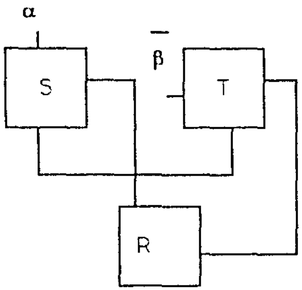


R : { $\overline{\beta}$,$\overline{\delta}$ }

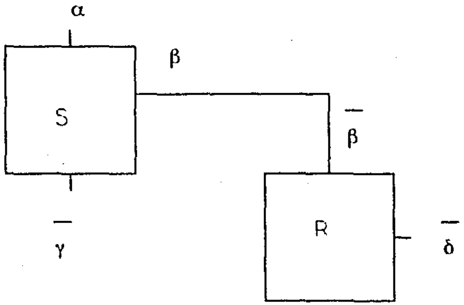
El resultado de la composición R || (S || T) es distinto del obtenido de la composición (R || S) || T.

$R \parallel (S \parallel T)$ $(R \parallel S) \parallel T$

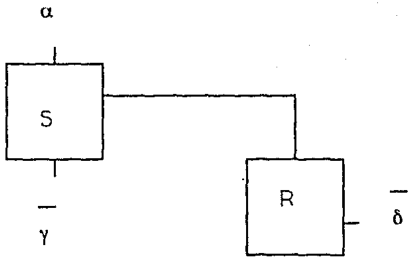




Para la obtención de mejores propiedades de la operación de composición, conviene introducir dos operadores más elementales en términos de los cuales se puede construir $\parallel$, uno para ligar puertos complementarios sin eliminar las etiquetas (composición $|$) y otro para el ocultamiento (restricción $\backslash$). Así por ejemplo $R \mid S$ representa al siguiente diagrama:

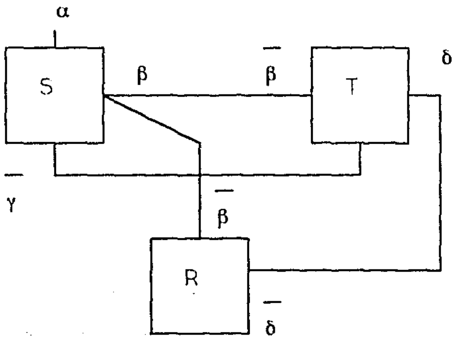


y $(R \mid S) \setminus \beta$ denota al siguiente:



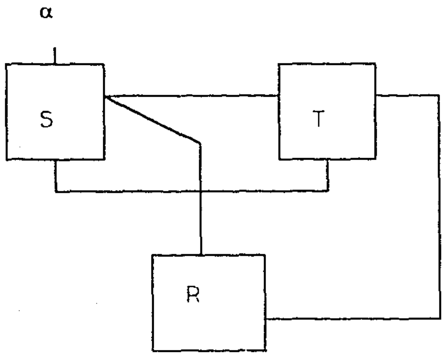
Obsérvese que $R \| S = (R|S)\setminus\beta$ y similarmente $S \| T = (S \mid T)\setminus\beta\setminus\gamma$.

La operación de composición $\mid$ es asociativa y conmutativa.



$$R \mid (S \mid T) = (R \mid S) \mid T = R \mid S \mid T$$

$((R \mid S) \mid T)\backslash\beta\backslash\gamma\backslash\delta$ corresponde a ocultar las etiquetas con nombres β, γ, δ.



La operación de ocultamiento asociada con una etiqueta α ($\backslash\alpha$) corresponde a interiorizar los puertos con nombre α de tal forma que los eventos mencionados no pueden ser observados desde el exterior.

El operador || se puede definir en términos de | y \ de la siguiente manera:

Definición .

Sean $S : L$ y $T : M$, entonces

$S \| T = (S | T) \setminus \alpha_1 .. \setminus \alpha_n$ en donde $\{ \alpha_1, .., \alpha_n \} =$ nombre $(L \cap \overline{M})$.

Definición .

Una función $S : L \rightarrow M$ es un re - etiquetamiento de L a M sí

i. S es biyectiva,

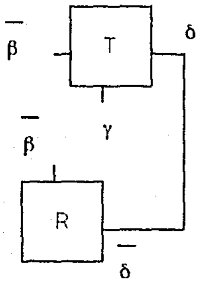
ii. $\overline{S(\alpha)} = S(\overline{\alpha})$ cuando $\alpha, \overline{\alpha} \in L$ (S respeta complementos).

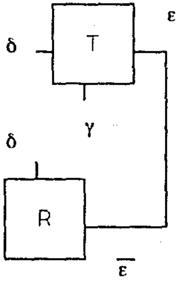
Un ejemplo de re - etiquetamiento es el siguiente :

$$S : \{ \overline{\beta}, \gamma, \delta, \overline{\delta} \} \rightarrow \{ \delta, \gamma, \varepsilon, \overline{\varepsilon} \}$$

$S(\overline{\beta}) = \delta$, $S(\gamma) = \gamma$, $S(\delta) = \varepsilon$, $S(\overline{\delta}) = \overline{\varepsilon}$

R | T (R | T) [S]





Definición.

Un re-etiquetamiento [S] sobre una caja R de género L (denotado R[S]), consiste en reemplazar cada etiqueta $\lambda \in \Lambda$ por $S(\lambda)$.

Notación.

$\lambda_1 / \alpha_1, \ldots, \lambda_n / \alpha_n$ o $\lambda_1\lambda_2 \ldots \lambda_n / \alpha_1\alpha_2 \ldots \alpha_n$ en donde los α_i son distintos y $\lambda_i \in \Lambda$ significa

i. $S(\alpha_i) = \lambda_i \qquad \alpha_i \in L$

ii. $S(\overline{\alpha_i}) = \overline{\lambda_i} \qquad \overline{\alpha_i} \in L$

iii. $S(\lambda) = \lambda \qquad \lambda \notin \{ \alpha_1, \ldots, \alpha_n \}$

y $r[\lambda_1 / \alpha_1, \ldots, \lambda_n / \alpha_n]$ denota a $r[S]$.

Nota: No es difícil probar que el re-etiquetamiento distribuye sobre la composición |,

$(r \mid t) [S] = r[S] \mid t[S]$.

Hasta este punto tenemos definidos tres operadores actuando sobre los dibujos de cajas negras, a saber | , $\backslash \alpha$ y [S]. Estos mismos operadores se introducen ahora sobre ST's.

EXTENSION DEL ALGEBRA A LOS ARBOLES DE SINCRONIZACION.

Notación.

En lo sucesivo λ denotará un elemento en Λ y μ ,υ denotarán elementos del conjunto $\Lambda \cup \{ \tau \}$.

Notación.

Un árbol de sincronización t se denotará mediante $t = \sum_{1 \le i \le m} \mu_i t_i$ en donde $\mu_i \in \Lambda \cup \{\tau\}$, $t_i \in ST$. Por convención, el caso especial cuando $m = 0$ denota al árbol NIL.

Las operaciones NIL, de adición y prefijo se definen en la misma forma que para los árboles RST.

Definición.

Operación de composición $| : ST_L \times ST_M \rightarrow ST_{L \cup M}$.

Sean $t = \sum_i \mu_i t_i$ y $u = \sum_j \upsilon_j u_j$ árboles de sincronización, entonces

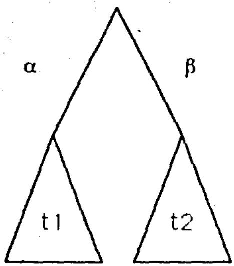
$$t \mid u = \sum_i \mu_i (t_i | u) + \sum_j \upsilon_j (t | u_j) + \sum_{\mu_i = \overline{\upsilon_j}} \tau (t_i | u_j) .$$

Nótese que de acuerdo a $\sum_{1 \le i \le 0} \mu_i t_i = NIL$, casos particulares de la definición anterior son:

$$NIL \mid NIL = NIL \quad y \quad NIL \mid t = t.$$

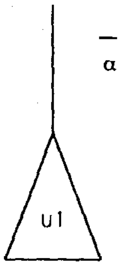
Un ejemplo de composición es el siguiente :

t =



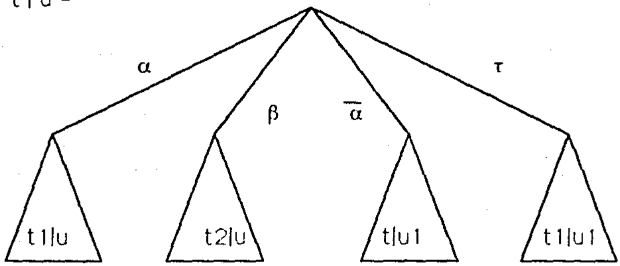
$\alpha t_1 + \alpha t_2$

u =



$\bar{\alpha} u_1$

t | u =



$\alpha(t_1|u) \quad + \quad \beta(t_2|u) \quad + \quad \bar{\alpha}(t|u_1) \quad + \quad \tau(t_1|u_1)$

En la definición de $t \mid u$, los dos primeros términos corresponden a eventos independientes de t y u respectivamente, el último término corresponde a una interacción entre t y u. Como se observa, la composición se define recursivamente a través de los descendientes.

Nota: Sin grandes dificultades se pueden demostrar

$t \mid u = u \mid t, \quad t \mid (u \mid v) = (t \mid u) \mid v.$

Definición .

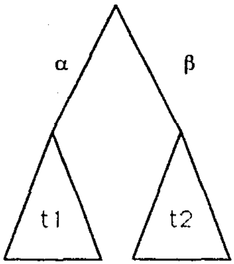
Operación de restricción $\backslash \alpha : ST_L \rightarrow ST_{L - \{\alpha, \overline{\alpha}\}}$, $\alpha \in \Lambda$.

Sea $t = \sum_i \mu_i t_i$ un árbol de sincronización , entonces

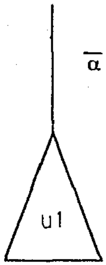
$$t \backslash \alpha = \sum_{\mu_i \notin \{\alpha, \overline{\alpha}\}} \mu_i (t_i \backslash \alpha) \quad .$$

Ejemplo.

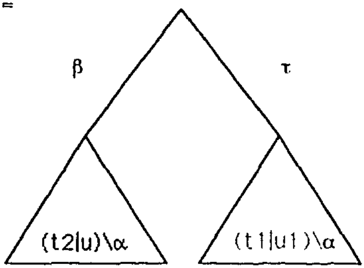
t =



u =



(t | u) \ α =



Esta operación corresponde a podar todas las ramas etiquetadas con α y $\overline{\alpha}$.

Operación de re - etiquetamiento $[S] : ST_L \rightarrow ST_M$, con $S : L \rightarrow M$.
Definición .
Sea $t = \sum_i \mu_i t_i$ un árbol de sincronización , entonces

$$t[S] = \sum_i S(\mu_i)\,(t_i[S]) \quad .$$

TEOREMA DE EXPANSION DE ST 's.

Sea $t_i = \sum_j \mu^j t_i^j$ y sea $t = (t_1 \mid t_2 \mid \ldots \mid t_m) \setminus A$ en donde $A = \{\alpha_1, \ldots, \alpha_k\}$ y $\setminus A$ denota $\setminus \alpha_1 \setminus \alpha_2 \ldots \setminus \alpha_k$,entonces

$$\begin{aligned} t = & \sum \{\mu^j((t_1|\ldots|t_i^j|\ldots|t_m) \setminus A);\ 1 \le i \le m,\ \mu^j t_i^j \text{ un sumando de } t_i,\ \text{nombre}(\mu^j) \notin A\} \\ + & \sum \{\tau((t_1|\ldots|t_i^k|\ldots|t_j^l|\ldots|t_m) \setminus A);\ 1 \le i < j \le m,\ \mu^k t_i^k \text{ un sumando de } t_i, \\ & \qquad \mu^l t_j^l \text{ un sumando de } t_j,\ \overline{\mu^k} = \mu^l\}. \end{aligned}$$

Demostración .

Inducción sobre m ,

i. Para m = 1 o 2 , el resultado corresponde a las definiciones de los operadores | y \ A.

ii. Para m > 2. Suponga el resultado válido para m - 1 y para todo conjunto A.

$$t = (t_1 | t_2 | \ldots | t_m) \setminus A$$
$$= (\sum \mu^k t_1^k \mid (t_2 | \ldots | t_m)) \setminus A$$
$$= [\sum \mu^k t_1^k \mid (\sum \{ \mu^j (t_2 | \ldots | t_i^j | \ldots | t_m); \ 2 \le i \le m, \ \mu^j t_i^j \text{ un sumando de } t_i \}$$
$$+ \sum \{ \tau (t_2 | \ldots | t_i^k | \ldots | t_j^l \ldots | t_m); \ 2 \le i < j \le m, \ \mu^k t_i^k \text{ un sumando de } t_i,$$
$$\mu^l t_j^l \text{ un sumando de } t_j, \overline{\mu^k} = \mu^l \})] \setminus A$$

lo anterior aplicando la hipótesis inductiva con A = {},

$$t = [\sum \mu^k (t_1^k | (t_2 | \ldots | t_m))$$
$$+ \sum \{ \mu^j (t_1 | t_2 | \ldots | t_i^j | \ldots | t_m); \ 2 \le i \le m, \ \mu^j t_i^j \text{ un sumando de } t_i \}$$
$$+ \sum \{ \tau (t_1 | t_2 | \ldots | t_i^k | \ldots | t_j^l \ldots | t_m); \ 2 \le i < j \le m, \ \mu^k t_i^k \text{ un sumando de } t_i,$$
$$\mu^l t_j^l \text{ un sumando de } t_j, \overline{\mu^k} = \mu^l \}$$
$$+ \sum \{ \tau (t_1^k | t_2 | \ldots | t_i^j | \ldots | t_m); \ 2 \le i \le m, \ \mu^k t_1^k \text{ un sumando de } t_1,$$
$$\mu^j t_i^j \text{ un sumando de } t_i, \overline{\mu^k} = \mu^j \}] \setminus A$$

$$= [\sum \{ \mu^j (t_1 | \ldots | t_i^j | \ldots | t_m); \ 1 \le i \le m, \ \mu^j t_i^j \text{ un sumando de } t_i \}$$
$$+ \sum \{ \tau (t_1 | \ldots | t_i^k | \ldots | t_j^l \ldots | t_m); \ 1 \le i < j \le m, \ \mu^k t_i^k \text{ un sumando de } t_i,$$
$$\mu^l t_j^l \text{ un sumando de } t_j, \overline{\mu^k} = \mu^l \}] \setminus A$$

El resultado se sigue de la definición de \ A y de que $(\sum t_i) \setminus A = \sum t_i \setminus A$.

3. Aplicación de CCS en ejemplos tradicionales.

3.1. Semáforos.

Dijkstra introduce el concepto de semáforo en el año de 1965 para enfrentar problemas de sincronización entre procesos que cooperan para la consecución de un objetivo común o de procesos que compiten por la obtención de recursos no compartibles.
Una de las bondades de los semáforos radica en la suspensión efectiva de los procesos, lo cual tiene como consecuencia una mejor utilización de los recursos de procesamiento. Algunas implementaciones típicas de los semáforos como objetos de sincronización entre procesos están basadas en la definición y control de variables de tipos discretos (como enteros, boolenanos, enumerados,etc) y de un par de operaciones primitivas indivisibles ejecutadas en forma atómica, referidas originalmente por Dijkstra mediante la notación P(s) y V(s) en donde s representa al semáforo.
Concibiendo a un semáforo s como una variable entera, la semántica de las operaciones P(s) y V(s) es la siguiente :

P(s)	:	disminuir el valor de s en 1 tan pronto como la condición $s > 0$ sea cierta.
V(s)	:	incrementar el valor de s en 1.

Todo proceso que realice una invocación a la operación P(s) cuando la condición $s > 0$ es falsa es suspendido hasta el momento en el cual algún otro proceso realice la operación V(s) (incrementando el valor del semáforo s) y el proceso suspendido sea seleccionado para concluir la ejecución de la operación P(s).

Un semáforo visualizado como una variable que admite solamente valores en {0,1} es denominado un semáforo binario.

El siguiente ejemplo presenta una especificación de sincronización entre procesos a través de la utilización de semáforos.

Sea $S = \overline{\pi}\overline{\phi}S$ un semáforo en donde la operación P de Dijkstra corresponde al experimento $\overline{\pi}$ y la operación V de Dijkstra corresponde al experimento $\overline{\phi}$.

Sea P un proceso con sección crítica $< \alpha, \beta >$

$$P = \pi \alpha \beta \ \phi P$$

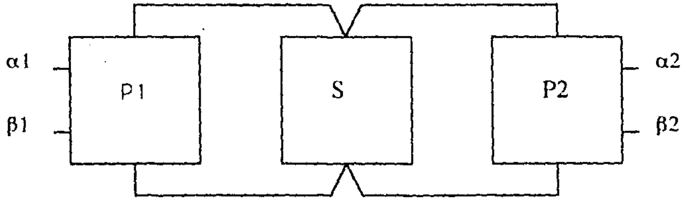
y obtenga instancias (copias) de P mediante el re - etiquetamiento

$$S_i = \alpha_i \beta_i / \alpha\beta$$

$$P_i = P[S_i] = \pi \alpha_i \beta_i \phi P_i \ .$$

Un sistema con dos procesos y un semáforo binario está descrito por

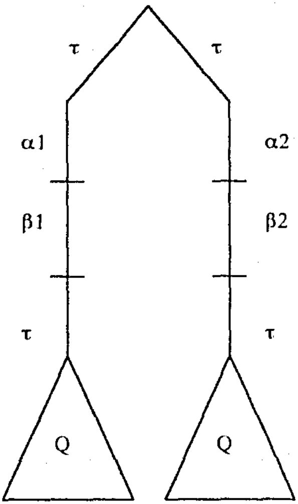
$$Q = (P_1 \mid P_2 \mid S) \setminus \pi \setminus \phi$$



$$
\begin{aligned}
Q &= (\pi \alpha_1 \beta_1\ \phi P_1 \mid \pi \alpha_2 \beta_2\ \phi P_2 \mid \overline{\pi}\ \overline{\phi} S)\ \backslash \pi \backslash \phi \\
&= \tau[\,(\alpha_1 \beta_1\ \phi P_1 \mid P_2 \mid \overline{\phi} S)\ \backslash \pi \backslash \phi\,] \;+\; \tau[\,(P_1 \mid \alpha_2 \beta_2\ \phi P_2 \mid \overline{\phi} S)\ \backslash \pi \backslash \phi\,] \\
&= \tau \alpha_1 \beta_1 [\,(\phi P_1 \mid P_2 \mid \overline{\phi} S)\ \backslash \pi \backslash \phi\,] \;+\; \tau \alpha_2 \beta_2 [\,(P_1 \mid \phi P_2 \mid \overline{\phi} S)\ \backslash \pi \backslash \phi\,] \\
&= \tau \alpha_1 \beta_1 \tau [\,(P_1 \mid P_2 \mid S)\ \backslash \pi \backslash \phi\,] \;+\; \tau \alpha_2 \beta_2 \tau [\,(P_1 \mid P_2 \mid S)\ \backslash \pi \backslash \phi\,] \\
&= \tau \alpha_1 \beta_1 \tau Q \;+\; \tau \alpha_2 \beta_2 \tau Q
\end{aligned}
$$

El árbol de sincronización muestra que las secciones críticas nunca se entrelazan.

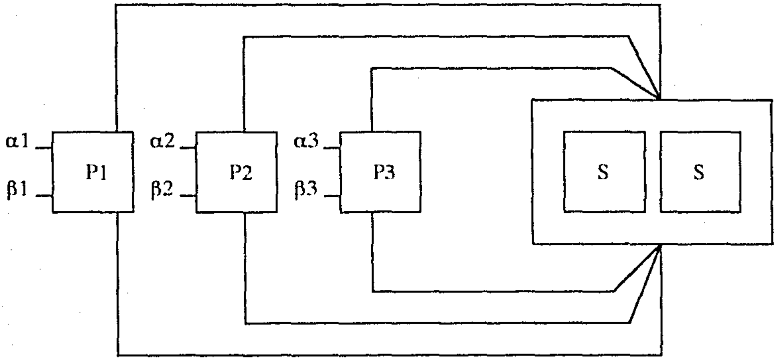
Q



Como una propiedad agradable de CCS , el caso de un sistema más general se obtiene mediante la composición de copias de los elementos anteriormente definidos.

$S_n = S \mid S \mid S \mid \ldots \mid S$ (n veces) semáforo general .

La figura correspondiente a un sistema con tres procesos y S_2 es



Un sistema con m procesos y n semáforos binarios se especifica con la composición

$Q = (P_1 \mid P_2 \mid \ldots \mid P_m \mid S_n) \setminus \pi \setminus \phi$.

3.2. Productor - Consumidor.

El problema de productor -consumidor puede describirse en forma sencilla como un sistema constituido de una entidad que produce elementos de consumo (algún tipo de datos), un almacenamiento temporal (buffer) que guarda los datos producidos y una tercera entidad que los consume .

Existen variantes para la definición de este problema dependiendo del número de productores y del número de consumidores involucrados ,así como de la capacidad de almacenamiento del buffer.

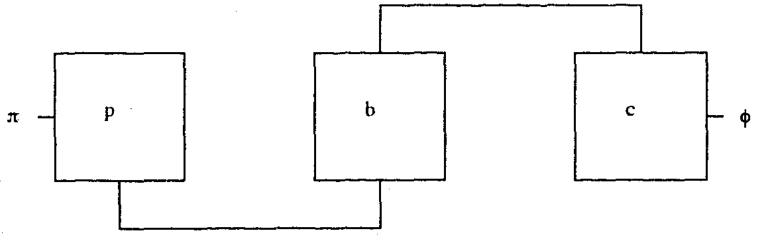
El siguiente ejemplo especifica un sistema bajo el modelo de productor-consumidor.

Sea $b = w\bar{r}b$ un buffer unitario en donde w representa al evento escritura y $\bar{r}$ denota al evento lectura .

Sea $p = \pi\bar{w}p$ un proceso productor en donde π denota al evento producción y sea $c = r\phi c$ un proceso consumidor en donde ϕ denota al evento consumo .

Un sistema con un productor , un consumidor y un buffer unitario se especifica como

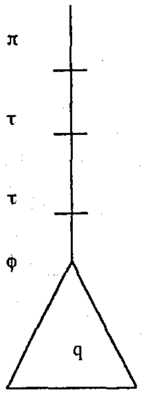
$$q = (b \mid p \mid c) \setminus w \setminus r$$



$$
\begin{aligned}
q &= ((\, w\bar{r}b \mid \pi\overline{w}p \mid r\phi c\,) \setminus w \ \setminus r) \\
&= \pi((\, w\bar{r}b \mid \overline{w}p \mid r\phi c\,) \setminus w \ \setminus r) \\
&= \pi\tau((\, \bar{r}b \mid p \mid r\phi c\,) \setminus w \ \setminus r\,) \\
&= \pi\tau^2((\, b \mid p \mid \phi c\,) \setminus w \ \setminus r\,) \\
&= \pi\tau^2\phi((\, b \mid p \mid c\,) \setminus w \ \setminus r\,) \\
&= \pi\tau^2\phi\, q
\end{aligned}
$$

El árbol de sincronización muestra la secuencia obigada de eventos

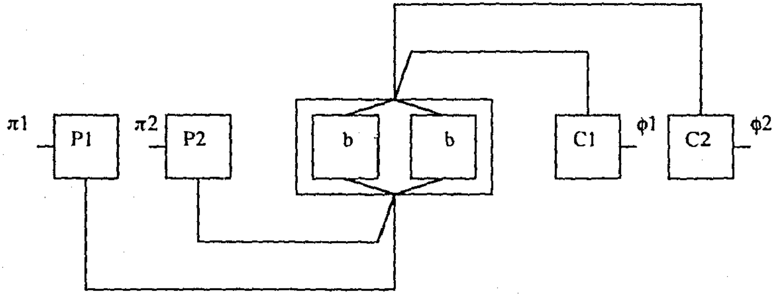
q



Un buffer general con capacidad n se obtiene componiendo n buffer's unitarios.

$$b_n = b \mid b \mid \ldots \mid b$$

Una figura que representa con cajas un sistema con dos productores, dos consumidores y un buffer con capacidad dos es la siguiente :



en donde $p_i = p[S_i] = \pi_i \overline{w} p_i \qquad S_i = \pi_i / \pi$

$c_i = c[S_i] = r\phi_i c_i \qquad S_i = \phi_i / \phi$

Un sistema general consistente de n productores, k buffer's unitarios y m consumidores queda especificado mediante

$q = (p_1 \mid p_2 \mid ... \mid p_n \mid b_k \mid c_1 \mid c_2 \mid ... \mid c_m) \setminus w \setminus r$

Nota. Para que el sistema funcione con propiedad y asegurar que todos los datos producidos sean consumidos, se debe asumir equitatividad en la participación de todos los buffer's unitarios ya que el orden de envío puede ser distinto al orden de recepción.

3.3. Filósofos.

El problema de los 5 filósofos constituye un ejemplo tradicional sobre el análisis de la concurrencia entre procesos.

El ejemplo consiste en la existencia de un monasterio en el cual viven 5 filósofos, cada uno de los cuáles se puede encontrar meditando o comiendo. Existe también un comedor común para todos los filósofos que consiste de una mesa en la cual están dispuestos 5 platos y 5 tenedores colocados en forma alternada.

Cada tenedor es compartido entre los dos filósofos con platos adyacentes al tenedor y para que un filósofo pueda comer es indispensable que utilize los dos tenedores adyacentes a su plato, de tal suerte que cuando un filósofo está comiendo imposibilita a sus comensales vecinos a saciar su apetito. La solución requerida para este problema debe evitar la posibilidad de que alguno de los filósofos muera por inanición. La especificación del sistema mediante la utilización de CCS es la siguiente :

Sea $t = \overline{\alpha}\overline{\beta}t$ el comportamiento asociado con un tenedor y sea

$f = k\,(\omega\rho + \rho\omega)\,\lambda\,(\varepsilon\theta + \theta\varepsilon)$ el comportamiento asociado con un filósofo .

Sea $t_i = t[S_i] = \overline{\mathrm{g}_i}\,\overline{\mathrm{p}_i}\,t_i$ en donde $S_i = \mathrm{g}_i p_i \,/\, \alpha\beta$ y sea

$f_i = f[S_i] = k_i\,(\mathrm{g}_i g_{i+1} + g_{i+1}\mathrm{g}_i)\,\lambda_i\,(\mathrm{p}_i p_{i+1} + p_{i+1}\mathrm{p}_i)\,f_i$

en donde $S_i = k_i\lambda_i \mathrm{g}_i g_{i+1} \mathrm{p}_i p_{i+1} \,/\, k\lambda\omega\rho\varepsilon\theta$ $\quad i = 0 .. 4$ y

con la aritmética de los subíndices realizada módulo 5.

g_i denota al evento tomar de la mesa al tenedor i

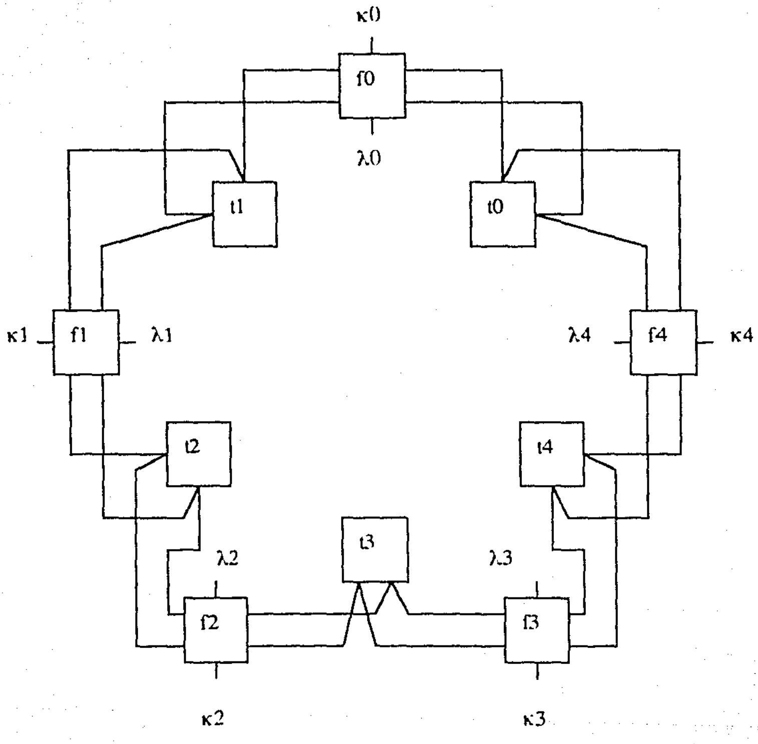
p_i denota al evento devolver a la mesa el tenedor i

k_i denota al evento filósofo i piensa

λ_i denota al evento filósofo i se alimenta .

Nota . En la definición de f se ha utilizado una operación de secuencia postfija que no ha sido presentada pero que se interpreta en la forma natural . Estrictamente sólo se ha introducido la operación prefijo .

La composición $(t_0 \mid t_1 \mid t_2 \mid t_3 \mid t_4 \mid f_0 \mid f_1 \mid f_2 \mid f_3 \mid f_4) \setminus A$
en donde $A = \{ g_i : i = 0..4 \} \cup \{ p_i : i = 0.. \}$ describe el problema de los cinco filósofos.


κ0
f0
λ0
t1
t0
κ1
f1
λ1
λ4
f4
κ4
t2
t4
t3
λ2
λ3
f2
f3
κ2
κ3

Esta solución no evita que los filósofos mueran por inanición ni tampoco que se produzca un comportamiento de interbloqueo total. Para evitar el interbloqueo se adiciona un semáforo

$$S_4 = (S \mid S \mid S \mid S)$$

describiendo la especificación "4 a la mesa" que limita el número de filósofos que simultáneamente pueden competir por la obtención de los tenedores.

Recuerde que un semáforo binario tiene el comportamiento

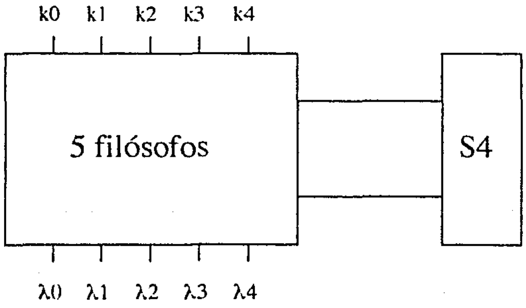
$S = \overline{\pi}\overline{\phi}S$

Sea $F_i = k_i \, \pi (\mathrm{g}_i \mathit{g}_{i+1} + \mathit{g}_{i+1} \mathrm{g}_i) \, \lambda_i (\mathrm{p}_i \mathit{p}_{i+1} + \mathit{p}_{i+1} \mathrm{p}_i) \, \phi F_i \quad , i = 0..4$

la nueva descripción está dada por la composición

$$q = (t_0 \mid t_1 \mid t_2 \mid t_3 \mid t_4 \mid f_0 \mid f_1 \mid f_2 \mid f_3 \mid f_4 \mid S_4) \setminus A \setminus \pi \setminus \phi$$

Un diagrama con cajas para éste sistema puede ser el siguiente,



Los únicos eventos que pueden ser observados desde el exterior corresponden a k_i, λ_i, $i = 0..4$.

El conjunto de acciones internas (no observables) regula los problemas de sincronía de los 5 filósofos.

3.4. Despachador.

Dado un conjunto de tareas

$$\{p_i : 1 \leq i \leq n\}$$

se desea diseñar un despachador que asegure que las tareas se ejecuten en forma rotativa.

Sea α_i el evento que describe la solicitud de iniciación de la tarea p_i y sea β_i la señal que indica su terminación ($1 \leq i \leq n$).

El despachador Sch de género $A \cup B$ (Sch: $A \cup B$) con $A = \{\alpha_i\}$ y $B = \{\beta_i\}$ está sujeto a las siguientes restricciones sobre cualquier secuencia de eventos posibles en $(A \cup B)^w$:

i. la eliminación de todas las ocurrencias β_i ($1 \leq i \leq n$) produce la secuencia

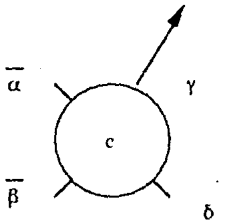
$$(\alpha_1 \alpha_2 \ldots \alpha_n)^w \quad ,$$

ii. para cada i, la eliminación de todas las ocurrencias α_j, β_j ($j \neq i$) produce la secuencia $(\alpha_i \beta_i)^w$.

Notación. $s^w = s(s)^w$.

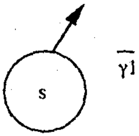
La representación gráfica del Sch utilizando rotores y un botón de arranque es

rotor



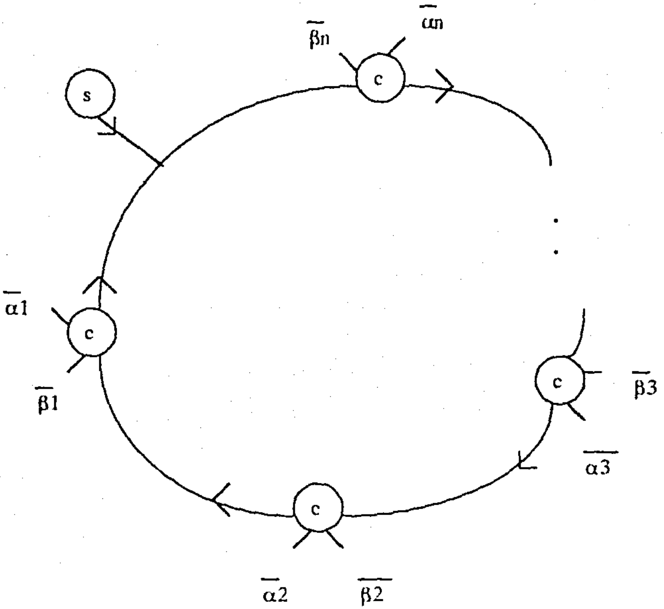
$$c = \gamma\,\overline{\alpha}(\,\overline{\beta}\,\delta\, c + \delta\,\overline{\beta}\ c)$$

botón de arranque



$$s = \overline{\gamma 1}\,\mathrm{NIL}$$

Sch



Nota. La marca ^ indica el primer evento a ocurrir en cada componente.

Sea $c_i = c[\alpha_i/\alpha, \beta_i/\beta, \gamma_i/\gamma, \overline{\gamma_{i+1}}/\delta]$

$= \gamma_i\overline{\alpha_i}(\overline{\beta_i}\,\overline{\gamma_{i+1}}c_i + \overline{\gamma_{i+1}}\,\overline{\beta_i}c_i)$, i = 1,,n

y la aritmética de los subíndices se realiza modularmente .

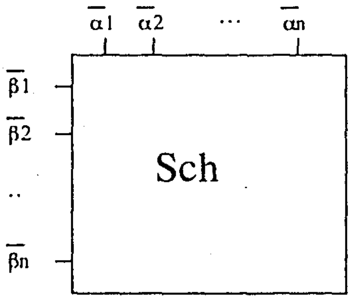
Cada rotor gira infinitamente como sigue :

i. se habilita por el rotor predecesor en γ,

ii.recibe solicitud de iniciación en $\overline{\alpha}$,

iii.recibe señal de terminación en $\overline{\beta}$ y habilita al rotor sucesor en δ , estos dos eventos en cualquier orden .

El botón de arranque interviene una única vez para iniciar al primer rotor.



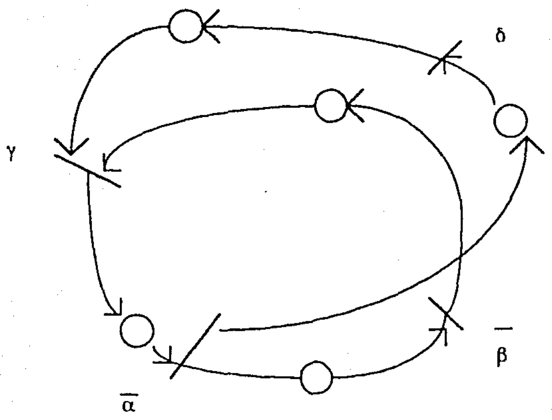
El despachador Sch está descrito mediante la composición

$$Sch = (s \mid c_1 \mid c_2 \mid \ldots \mid c_n) \setminus \gamma_1 \setminus \gamma_2 \setminus \ldots \setminus \gamma_n$$

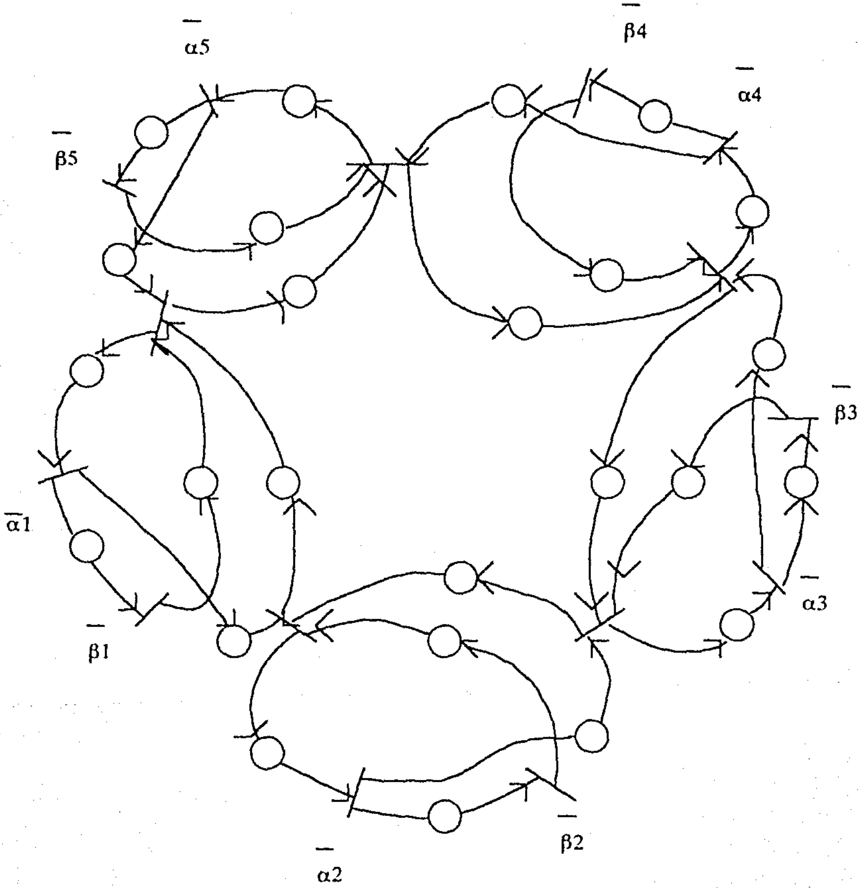
Una demostración de que Sch funciona correctamente será dada posteriormente.

Utililizando redes de Petri para modelar el despachador,un rotor está dado por

C



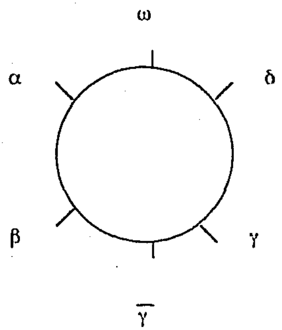
Despachador con cinco rotores.



Nota. En ésta red el botón de arranque ha sido eliminado por lo cual se ha tomado una configuración inicial de tokens especial.

3.5. Buffer cíclico.

La especificación del comportamiento de un buffer cíclico está dada mediante la composición de elementos básicos de la forma



en donde el significado de los eventos asociados a las etiquetas es el siguiente :

i. se recibe habilitación de escritura del predecesor en ∂
ii. se habilita la escritura en el sucesor en α
iii. se recibe habilitación de lectura del predecesor en β
iv. se habilita la lectura en el sucesor en γ
v. operación de escritura en ω
vi. operación de lectura en $\bar{r}$.

El elemento básico b corresponde a un

$$b = \delta\omega(\alpha\beta + \beta\alpha)\bar{r}\gamma b$$

Para un buffer con capacidad n considere

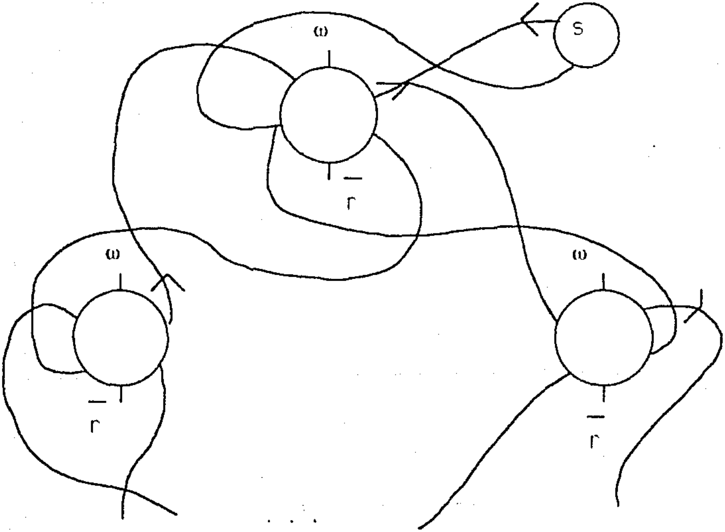
$$\begin{aligned} b_i &= b \mid \alpha_i / \alpha, \overline{\beta_{i-1}} / \beta . \beta_i / \gamma , \overline{\alpha_{i-1}} / \delta \mid \\ &= \overline{\alpha_{i-1}}\omega(\alpha_i\overline{\beta_{i-1}} + \overline{\beta_{i-1}}\alpha_i)\ \bar{r}\beta_i b_i \qquad , 1 \le i \le n \end{aligned}$$

y un elemento de arranque $S = \alpha_n\beta_n$ NIL.

Nota. La aritmética de los subíndices se realiza en forma modular.

La composición Buffer $= (s \mid b_1 \mid b_2 \mid \ldots \mid b_n) \setminus A$ en donde

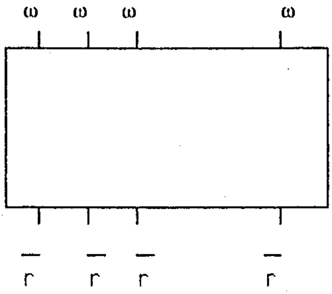
$A = \{ \alpha_1, \ldots, \alpha_n, \beta_1, \ldots \beta_n\}$ representa al sistema y la figura correspondiente es



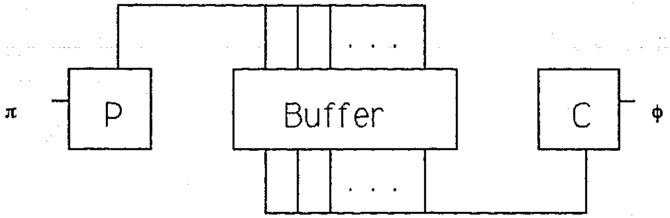
La marca ^ señala el evento inicial en cada componente básico.
Si $p = \pi\overline{\omega}p$ es un proceso productor (π denota al evento producción) y $c = r\phi c$ es un proceso consumidor (ϕ denota al evento consumo) entonces un sistema con un consumidor, un productor y un buffer cíclico se especifica como

$$q = (p \mid \text{Buffer} \mid c) \setminus \omega \setminus r$$

Buffer:



q

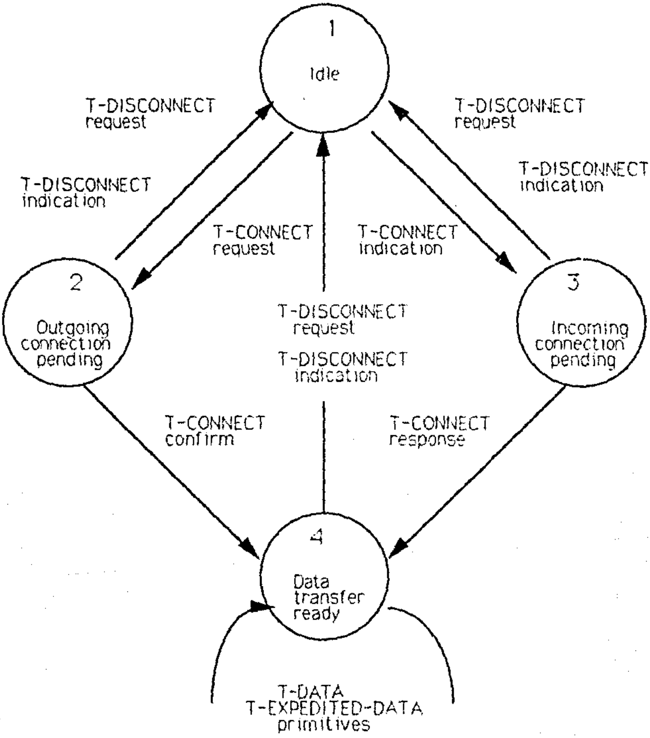


3.6. Servicio de transporte.

El nivel de transporte en la jerarquía de protocolos OSI, está destinado a proveer un servicio de transmisión a nivel de la red en forma eficiente y confiable a los procesos usuarios en el nivel de sesión. Existen dos tipos para el servicio de transporte, uno orientado a la conexión y el otro no orientado a la conexión, en el primero de ellos se distinguen 3 fases : El establecimiento de la conexión, la transferencia de los datos y la liberación de la conexión.

El modelo OSI provee una serie de primitivas de servicio en ambas modalidades de transporte, en particular, para el establecimiento de una conexión se utilizan 4 primitivas invocadas en varias posibles secuencias, por ejemplo, una entidad fuente puede ejecutar una primitiva T-CONNECT.request para indicar el deseo de establecer una conexión con una entidad destino la cual debe estar ligada a una dirección particular a través de un TSAP (transport service access point) . Como consecuencia de lo anterior se produce una primitiva T-CONNECT.indication dirigida a la entidad destino, misma que puede aceptar la conexión mediante el uso de la primitiva T-CONNECT.response o rechazarla con la primitiva T-DISCONNECT.request, en el primer caso la entidad fuente recibe la primitiva T-CONNECT.confirm y en el segundo caso la primitiva T-DISCONNECT.indication. El siguiente ejemplo ilustra la especificación del servicio a nivel de transporte de los protocolos OSI mediante la utilización de CCS.

En la recomendación X.214 de CCITT (TRANSPORT SERVICE DEFINITION FOR OPEN SYSTEM INTERCONNECTION (OSI) FOR CCITT APPLICATIONS), aparece la siguiente información relativa a las primitivas del servicio de transporte,



State Transition Diagram for Possible Allowed Sequence of TS Primitives at a Tc Endpoint

TRANSPORT SERVICE PRIMITIVES

Phase	Service	Primitive	Parameters
TC establishment	TC establishment	T-CONNECT request	(Called address, calling address, expedited data option, quality of service TS user-data)
		T-CONNECT indication	(Called address, calling address, expedited data option, quality of service TS user-data)
		T-CONNECT response	(Quality of service, responding address, expedited data option, TS user-data)
		T-CONNECT confirm	(Quality of service, responding address, expedited data option, TS user-data)
Data transfer	Normal data transfer	T-DATA request	(TS user data)
		T-DATA indication	(TS user-data
	Expedited data transfer	T-EXPEDITED DATA request	(TS user data)
		T-EXPEDITED DATA indication	(TS user data)
TC release	TC release	T-DISCONNECT request	(TS user-data)
		T-DISCONNECT indication	(Disconnect reason, TS user-data)

The TS primitive X / may be followed by the TS primitive Y	T-CONNECT request	T-CONNECT confirm	T-CONNECT indication	T-CONNECT response	T-DATA request	T-DATA indication	T-EXPEDITED-DATA request	T-EXPEDITED-DATA indication	T-DISCONNECT request	T-DISCONNECT indication
T-CONNECT request										
T-CONNECT confirm	+									
T-CONNECT indication										
T-CONNECT response			+							
T-DATA request		+		+	+	+	+	+		
T-DATA indication		+		+	+	+	+	+		
T-EXPEDITED DATA request		+		+	+	+	+	+		
T-EXPEDITED DATA indication		+		+	+	+	+	+		
T-DISCONNECT request	+	+	+	+	+	+	+	+		
T-DISCONNECT indication	+	+	+	+	+	+	+	+		

Sequences of TS Primitives at one End of aTC

Utilizando CCS para la descripción de la información anterior, se puede proceder de la siguiente manera .

Sea

D = (T.DATA.req) D + (T.DATA.ind) D

+ (T-EXPEDITED-DATA.req) D + (T-EXPEDITED-DATA.ind) D

+ (T-DISCONNECT.ind) + ((T.DISCONNECT.req)

y sea

T = (T-CONNECT.req) [(T.DISCONNECT.req) + (T.DISCONNECT.ind) +

(T.CONNECT.conf) D] +

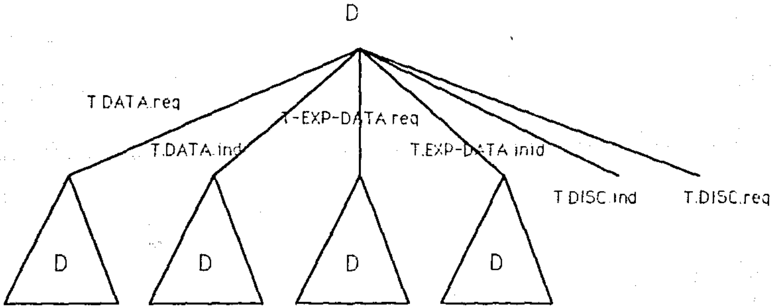
(T-CONNECT.ind) [(T.DISCONNECT.req) + (T.DISCONNECT.ind) +

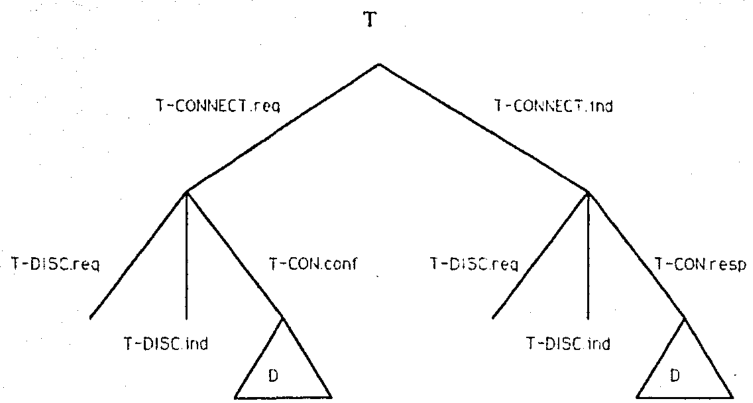
(T.CONNECT.resp) D]

D describe el comportamiento del estado DATA TRANSFER READY y T describe el comportamiento del servicio de transporte.

Nota. En este ejemplo se ha hecho uso de identificadores completos de acuerdo con la recomendación X.210 en lugar de utilizar etiquetas simples.

Para los anteriores comportamientos, los árboles de sincronización son





Lo anterior corresponde a otra forma de definir las posibles secuencias en las cuales las primitivas para el servicio de transporte pueden ocurrir.

4. Equivalencia observacional y comunicación de valores.

Para formalizar el concepto de equivalencia observacional considere un conjunto de comportamientos P y una familia de relaciones binarias sobre P denotada por

$$\{ \xrightarrow{\mu} ; \mu \in \Lambda \cup \{\tau\}\}$$

Definición .

i. Sea $\lambda \in \Lambda$, $p \xrightarrow{\lambda} p'$ significa que p admite un experimeto λ y se transforma en p' como resultado.

ii. $p \xrightarrow{\tau} p'$ significa que p puede transformarse en p ' en forma no observable.

iii. $\forall\ s = \mu_1, \ldots, \mu_n \in (\Lambda \cup \{\tau\})^*$, $p \xrightarrow{s} p'$ significa que existen $p_0, \ldots, p_n$ $(n \geq 0)$ tales que

$$p = p_0 \xrightarrow{\mu_1} p_1 \xrightarrow{\mu_2} p_2 \ldots \xrightarrow{\mu n} p_n = p'.$$

iv. $\forall\ s = \lambda_1, \ldots, \lambda_n \in \Lambda^*$, $p \overset{s}{\Longrightarrow} p'$ significa que existen $k_0, \ldots, k_n \geq 0$ tales que

$$p \xrightarrow{\tau^{k_0}\ \lambda_1 \tau^{k_1}\ \lambda_2 \ldots\ \lambda_n \tau^{kn}} p'$$

Nota. Sea $\varepsilon \in \Lambda^*$ la secuencia vacía , $p \overset{\varepsilon}{\Longrightarrow} p'$ sii $\exists\ k \geq 0$ tal que $p \xrightarrow{\tau^k} p'$

y $p \overset{\varepsilon}{\Longrightarrow} p'$ con $k = 0$.

A continuación se presenta la definición de equivalencia observacional a través de una secuencia $\approx_0$, $\approx_1$,..., $\approx_k$,.. de relaciones de equivalencia .

Definición .

Sean $p,q \in P$,

i. $p \approx_0 q$ siempre es cierto ,

ii. $p \approx_{k+1} q$ sii $\forall\ s \in \Lambda^*$

a) si $p \overset{s}{\Rightarrow} p'$ entonces $\exists\ q'$ tal que $q \overset{s}{\Rightarrow} q'$ y $p' \approx_k q'$

b) si $q \overset{s}{\Rightarrow} q'$ entonces $\exists\ p'$ tal que $p \overset{s}{\Rightarrow} p'$ y $p' \approx_k q'$

iii. $p \approx q$ sii $\forall k \geq 0$ $p \approx_k q$

Nota. $\approx\ = \bigcap_k \approx_k$

Dado $\mu \in \Lambda \cup \{\tau\}$ y $p \xrightarrow{\mu} p'$, a p' se le llama un comportamiento derivado de p. La definición de equivalencia observacional $p \approx q$ no sólo implica equivalencia en un primer nivel sino además la equivalencia de los comportamientos derivados .

Nota. $\forall\ k,\ \approx_k$ es una relación de equivalencia y $\approx_{k+1} \subseteq \approx_k$,

y $\therefore\ p \approx_{k+1} q$ implica $p \approx_k q$.

Nota. En términos de árboles de sincronización , $t \xrightarrow{\mu} t'$ sii t tiene una rama $\mu t'$.

Un resultado sencillo y directo es que $t \approx \tau t$ ya que

i. $t \approx_0 \tau t$ es cierto,

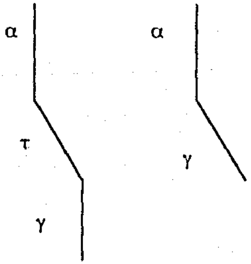
ii. $t \stackrel{s}{\Rightarrow} t'$ sii $\tau t \stackrel{s}{\Rightarrow} t'$ por aplicación directa de la definición de $\stackrel{s}{\Rightarrow}$, y $t' \approx_k t'$ $\forall k \geq 0$.

Una relación de congruencia es aquélla en la cual al reemplazar t por t' (cuando $t \approx t'$) en u para obtener u' implica que $u \approx u'$.

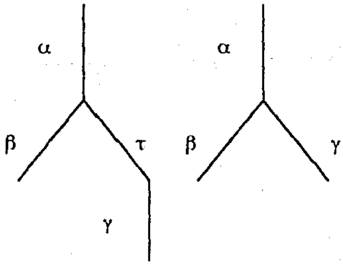
La equivalencia observacional no es una congruencia ya que $NIL \approx \tau NIL$ pero $\alpha NIL + NIL \not\approx \alpha NIL + \tau NIL$.

Como ejemplos de equivalencia observacional se presentan los siguientes árboles de sincronización :

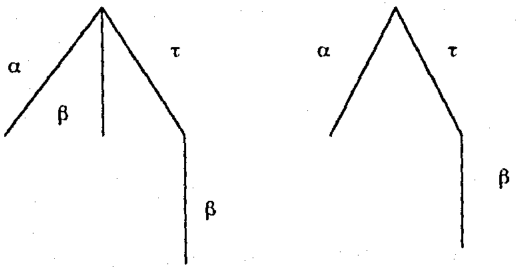
a)



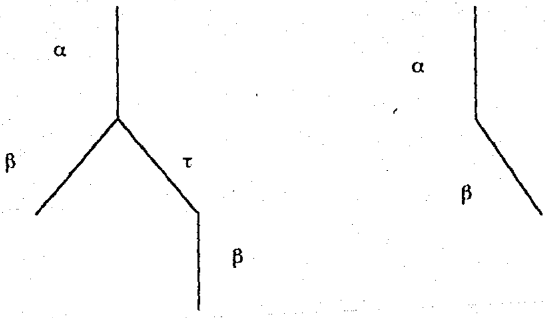
b)



c)

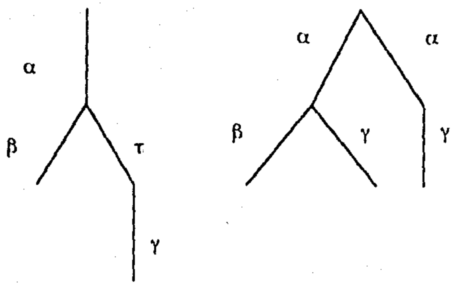


d)

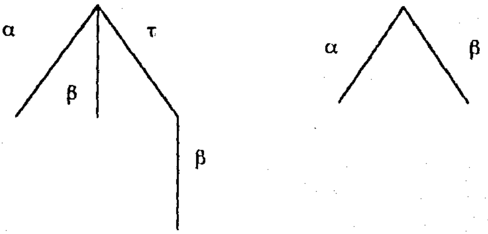


Como ejemplos de árboles de sincronización que no son equivalentes observacionalmente tenemos :

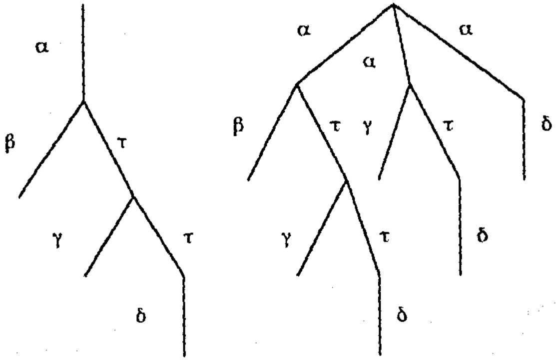
a)



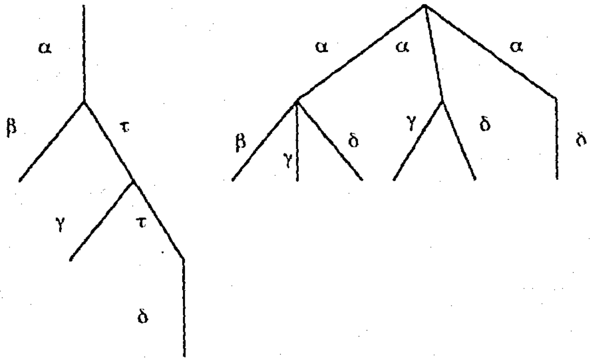
b)



c)



d)



COMUNICACION DE VALORES.

Hasta este punto se han considerado operaciones que establecen comportamientos secuenciales no determinísticos,

i. inacción NIL
ii. adición $+$
iii. acción prefijo μ $(\in \Lambda \cup \{\tau\})$

y que se denominan operaciones dinámicas. También se han presentado operaciones estáticas que establecen una estructura ligada fija entre comportamientos activos concurrentes, a saber

iv. composición $|$
v. restricción $\backslash \alpha$ $(\alpha \in \Delta)$
vi. re - etiquetamiento $[S]$.

Para introducir el intercambio de valores entre un conjunto de comportamientos . considere $p = \alpha\beta\,\overline{\gamma}\,p$ y distinga a las etiquetas positivas (α y β) como aceptadoras de valores y a las etiquetas negativas ($\overline{\gamma}$) como emisoras de valores, entonces p recibe un par de valores (en un orden específico) y emite un tercer valor .
Un α - experimento en p consiste en ofrecer a p un valor en α y un $\overline{\gamma}$ - experimento en p consiste en demandar de p un valor en $\overline{\gamma}$.

La expresión $p = \alpha x.B$, en donde x representa una variable (receptora de un valor en un α - experimento), y B es una expresión de comportamiento conteniendo a x como variable libre , denota el ofrecimiento a p de un valor en α. Para esta expresión se dice que la variable x es acotada por α y que su alcance en B .

La expresión $p = \overline{\beta}\, v.B$, en donde v representa un valor (emitido en un $\overline{\beta}$ - experimento) y B es una expresión de comportamiento denota la demanda a p de un valor en $\overline{\beta}$.

Para generalizar la familia $\{ \xrightarrow{\mu} ; \mu \in \Lambda \cup \{\tau\} \}$ de relaciones binarias , se proporciona la siguiente

Definición .

Sea $\lambda \in \Lambda$, v un valor

i. $p \xrightarrow{\overline{\lambda v}} p'$ significa que p admite un $\overline{\lambda}$ v - experimento y se transforma en p ' como resultado,

ii. $p \xrightarrow{\lambda x} p'$ signfica que p admite un λ x - experimento y se transforma en p ' como resultado.

Como regla general se tiene que

$$\alpha\, x.B \xrightarrow{\alpha v} B\{ v / x\}$$

en donde $B\{ v / x\}$ denota el resultado de sustituir v en todas las ocurrencias libres de x en B, además se tiene también que

$$\overline{\gamma}\, v.B \xrightarrow{\overline{\gamma v}} B$$

para cualquier valor particular v .

Ejemplo.

Sea $p = \alpha x.\ \beta y.\ \overline{\gamma}\,(x + y).\ p$

$$p \xrightarrow{\alpha 3} \beta\, y.\overline{\gamma}\,(3 + y).\ p$$

$$\xrightarrow{\beta 4} \overline{\gamma}\,(3 + 4).\ p$$

$$\xrightarrow{\overline{\gamma} 7} p.$$

Una expresión típica de un comportamiento conteniendo la comunicación de valores está dada por

$$B = \sum_i \alpha_i x_i . B_i + \sum_j \overline{\beta}_j E_j . B_j' + \sum_k \tau . B_k''$$

en donde B_i, B_j', B_k'' denotan expresiones de comportamiento, x_i son variables y E_j son expresiones de valor.

Notación. $\tilde{x} = x_1 \ldots x_n$, $\tilde{E} = E_1 \ldots E_n$.

El término guardia denota cualquiera de los prefijos $\alpha \tilde{x}$, $\overline{\beta}\,\tilde{E}$, τ . El término guardia es creación de Dijkstra y representa alguna condición que debe ser cumplida como requisito, para la ejecución de una parte de algún programa. Hoare adaptó el concepto de guardia cuando la condición es la aceptación de una comunicación ofrecida como lo muestra el lenguaje CSP. Considerando el concepto de guardia, una expresión de comportamiento general es

$$\sum_k g_k . B_k$$

en donde nombre(g_k) corresponde a la etiqueta en g_k.

$B \overset{\varepsilon}{\Rightarrow} B'$ denota $B \xrightarrow{\tau^n} B'$ $(n \geq 0)$ y

$B \overset{\mu v}{\Rightarrow} B'$ representa $B \xrightarrow{\tau^n.\mu v.\tau^m} B'$ $(m,n \geq 0)$

Ejemplo.

Sea $B = ((\alpha x.B_1 + \beta y.B_2) \mid \overline{\beta} v.\gamma z.B_3) \setminus \beta$

$$B \xrightarrow{\alpha 5} (B_1\{5/x\} \mid \overline{\beta} v.\gamma z.B_3) \setminus \beta$$

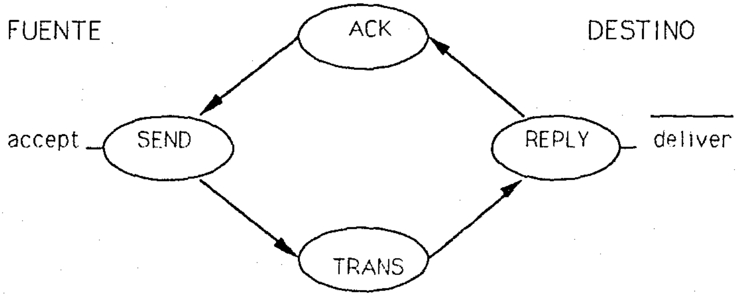
$$B \xrightarrow{\tau} (B_2\{v/y\} \mid \gamma z.B_3) \setminus \beta \xrightarrow{\gamma 7} (B_2\{v/y\} \mid B_3\{7/z\}) \setminus \beta$$

5. Ejemplos utilizando comunicación de valores.

5.1. Protocolo bit alternante.

Un protocolo puede ser descrito como un conjunto de reglas y formatos que constituyen una disciplina para establecer una comunicación. En general los medios de comunicación son de naturaleza adversa en el sentido de que los mensajes transmitidos pueden ser alterados, perdidos o duplicados. Para evitar estos inconvenientes existen algunos algoritmos orientados al control del flujo, al control de errores, al envío de mensajes especiales de reconocimiento o acuse de recepción, etc.

En este ejemplo se asumen canales de comunicación con buffer's no acotados. El esquema del sistema es el siguiente :



Los canales de comunicación (TRANS, ACK) se asumen ser no confiables permitiendo la pérdida y duplicación de mensajes pero no su alteración.

El protocolo de bit alternante utiliza los bits 0 y 1 en forma alterna para realizar el reconocimiento de mensajes transmitidos.

El emisor (SEND) después de aceptar un mensaje, lo transmite conjuntamente con el bit b sobre la línea TRANS e inicializa un temporizador (timer). A continuación se presentan tres posibilidades :

a) se recibe un tiempo fuera (timeout) y el emisor reenvía el mensaje junto con el bit b,

b) se recibe un reconocimiento b de la línea ACK y el emisor queda listo para aceptar otro mensaje que será transmitido junto con el bit $\hat{b} = 1 - b$.

c) se recibe un reconocimiento $\hat{b}$ de la línea ACK el cual es ignorado .

En el caso del receptor (REPLY) , después de entregar el mensaje recibido procede a enviar un reconocimiento con el bit b por la línea ACK e inicializa un temporizador .

Posteriormente se presentan tres posibles eventos :

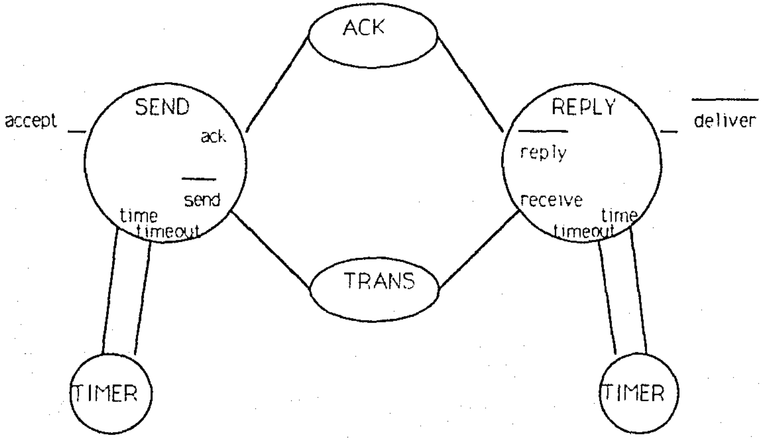
a) se recibe un tiempo fuera y el receptor reenvía el reconocimiento con el bit b ,

b) se recibe un nuevo mensaje con el bit $\hat{b}$ de la línea TRANS , se hace entrega del mensaje y se envía un reconocimiento con el bit $\hat{b} = 1 - b$,

c) se recibe una retransmisión del mensaje previo con el bit b el cual es ignorado .

El orden de transmisión de los mensajes no es alterado por las líneas de comunicación . El canal TRANS transporta una secuencia de mensajes de la forma $(b_1,m_1) \ldots (b_n,b_n)$ con $b_i \in \{0,1\}$ y el canal ACK transporta una secuencia de la forma $b_1 \ldots b_n$ con $b_i \in \{0,1\}$.

Nota. Para este protocolo, sólo es posible que circulen a lo más dos mensajes distintos en los medios de comunicación en forma simultánea.



$$\mathrm{ACCEPT}(b) \Leftarrow \mathrm{accept}(m)\,.\,\mathrm{SEND}(b,m)$$

$$\mathrm{SEND}(b,m) \Leftarrow \overline{\mathrm{send}}(b,m)\,.\,\overline{\mathrm{time}}\,.\,\mathrm{SENDING}(b,m)$$

$$\mathrm{SENDING}(b,m) \Leftarrow \mathrm{timeout}\,.\,\mathrm{SEND}(b,m) + \mathrm{ack}(x)\,.\,\mathrm{CHECK1}(x,b,m)$$

$$\mathrm{CHECK1}(x,b,m) \Leftarrow \text{if } x = b \text{ then timeout }.\,\mathrm{ACCEPT}(\hat{b}) \text{ elseSENDING}(b,m)$$

$$\mathrm{TIMER} \Leftarrow \mathrm{time} \, . \, \overline{\mathrm{timeout}} \, . \, \mathrm{TIMER}$$

$$\mathrm{DELIVER}\,(b) \Leftarrow \overline{\mathrm{deliver}}(m) \, . \, \mathrm{REPLY}\,(b)$$

$$\mathrm{REPLY}\,(b) \Leftarrow \overline{\mathrm{reply}}(b) \, . \, \overline{\mathrm{time}} \, . \, \mathrm{REPLYING}(b)$$

$$\mathrm{REPLYING}(b) \Leftarrow \mathrm{timeout} \, . \, \mathrm{REPLY}(b) + \mathrm{receive}(y,m) \, . \, \mathrm{CHECK2}(y,b)$$

$$\mathrm{CHECK2}(b,m) \Leftarrow \text{if } y = b \text{ then } \mathrm{REPLYING}(b) \text{ else timeout} \, . \, \mathrm{DELIVER}\,(\hat{b})$$

$$\begin{aligned} \mathrm{TRANS}(\,(b_1,m_1) \ldots (b_n,b_n)\,) \Leftarrow\ & \mathrm{send}(b,m) \, . \, \mathrm{TRANS}(\,(b,m)(b_1,m_1) \ldots (b_n,m_n)\,) \\ & + \overline{\mathrm{receive}}\,(b_n,m_n) \, . \, \mathrm{TRANS}(\,(b_1,m_1) \ldots (b_{n-1},m_{n-1})\,) \\ & + \sum_i \tau \, . \, \mathrm{TRANS}(\,(b_1,m_1) \ldots (b_{i-1},m_{i-1})(b_{i+1},m_{i+1}) \ldots (b_n,m_n)\,) \\ & + \sum_i \tau \, . \, \mathrm{TRANS}(\,(b_1,m_1) \ldots (b_i,m_i)(b_i,m_i) \ldots (b_n,m_n)\,) \end{aligned}$$

Los dos últimos sumandos representan la posibilidad de pérdida y duplicación de mensajes.

$$\begin{aligned} \mathrm{ACK}(b_1 \ldots b_n) \Leftarrow\ & \overline{\mathrm{ack}}(b_1) \, . \, \mathrm{ACK}(b_2 \ldots b_n) \\ & + \mathrm{reply}(b_{n+1}) \, . \, \mathrm{ACK}(b_1 \ldots b_n b_{n+1}) \\ & + \sum \tau \, . \, \mathrm{ACK}(b_1 \ldots b_{i-1} b_{i+1} \ldots b_{n+1}) \\ & + \sum \tau \, . \, \mathrm{ACK}(b_1 \ldots b_i b_i \ldots b_{n+1}) \end{aligned}$$

Los dos últimos sumandos representan la posibilidad de pérdida y duplicación de los reconocimientos.

La composición del emisor con su temporizador resulta en las siguientes ecuaciones :

$SEND'(b,m) = (SEND(b,m) \mid TIMER) \setminus \{time, timeout\}$

$ACCEPT'(b) = accept(m) \,.\, SEND'(b,m)$

$SEND'(b,m) = \overline{send}(b,m) \,.\, \tau \,.\, SENDING'(b,m)$

$SENDING'(b,m) = \tau \,.\, SEND'(b,m) + ack(x) \,.\, CHECK1'(x,b,m)$

$CHECK1'(x,b,m) = \text{if } x = b \text{ then } \quad \tau \,.\, ACCEPT'(\hat{b}) \text{ else } SENDING'(b,m).$

La composición del receptor con su temporizador resulta en las siguientes ecuaciones :

$REPLY'(b) = (REPLY(b) \mid TIMER) \setminus \{time, timeout\}$

$DELIVER'(b) = \overline{deliver}(m) \,.\, REPLY'(b)$

$REPLY'(b) = \overline{reply}(b) \,.\, \tau \,.\, REPLYING'(b)$

$REPLYING'(b) = \tau \,.\, REPLY'(b) + receive(y,m) \,.\, CHECK2'(y,b)$

$CHECK2'(y,b) = \text{if } y = b \text{ then } REPLYING'(b) \text{ else } \tau \,.\, DELIVER'(\hat{b})$

La definición completa del sistema es

$$AB \Leftarrow (ACCEPT'(\hat{b}) \mid TRANS(\varepsilon) \mid ACK(\varepsilon) \mid REPLY'(b)) \setminus A$$

con $A = \{ send, receive, reply, ack \}$.

Este sistema es equivalente al comportamiento dado por

$$Buff \Leftarrow accept \,.\, \overline{deliver} \,.\, Buff.$$

Notación. Se ha utilizado la forma $\alpha(x) \,.\, B$ en vez de $\alpha x.B$ a lo largo de este ejemplo .

5.2. Ordenamiento.

Se desea contruir una máquina Ordena $_n$:{ in,$\overline{out}$ } para ordenar n números enteros positivos ($n \geq 0$) .Si S es un multiconjunto (en el cual se permite la repetición de elementos) conteniendo los números a ordenar , entonces la especificación de la máquina de ordenamiento es

$$\begin{aligned}
\mathrm{EspOrdena}_n &\Leftarrow \mathrm{in}\ x_1 . \ \ldots \ .\mathrm{in}\ x_n \ . \mathrm{Ct}(\ \{x_1,..,x_n\}\ ) \\
\mathrm{Ct}(S) &\Leftarrow \overline{\mathrm{out}}(\max S) . \mathrm{Ct}(\ S - \{\max S\}) \qquad ,(S \neq \Phi) \\
\mathrm{Ct}(\Phi) &\Leftarrow \overline{\mathrm{out}}(0) . \mathrm{EspOrdena}_n \ .
\end{aligned}$$

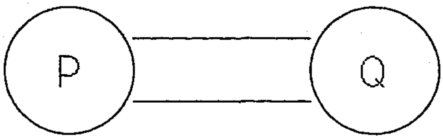
Los números ordenados se entregan en forma descendiente terminando con un cero.

Ordena $_n$ se construye mediante n celdas C y una celda B con la operación de atadura dada a continuación

$$P \frown Q = (\ P[\alpha / \mathrm{down}, \beta / \mathrm{up}] \mid Q[\alpha / \mathrm{in}, \beta / \mathrm{out}]\ ) \setminus \{\alpha, \beta\}$$

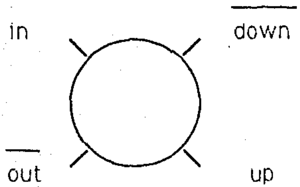
$$\alpha, \beta \notin L(P) \cup L(Q)$$

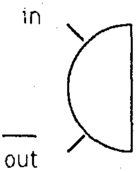
$P \frown Q$



Las celdas C y B se representan mediante

C **B**

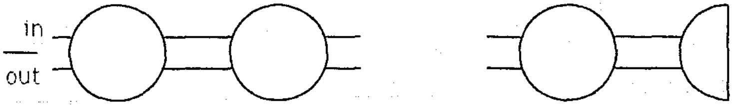




La máquina Ordena_n corresponde a

$$\text{Ordena}_n = \text{C- C- ... - C- C- B} \qquad (\text{n veces C})$$

$$\text{Ordena}_{n+1} = \text{C- Ordena}_n$$



en donde

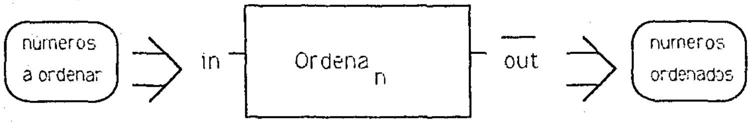
$$C \Leftarrow \text{in}\, x\,.\,C'(x)$$

$$C'(x) \Leftarrow \overline{\text{down}}\, x\,.\,C + \text{up}\, y.\overline{\text{out}}(\max\{x,y\})\,.C''(\min\{x,y\})$$

$$C''(x) \Leftarrow \text{if } x = 0 \text{ then } \overline{\text{out}}\ 0.C \text{ else } C'(x)$$

$$B \Leftarrow \overline{\text{out}}\ 0.B\ .$$

Nota. Ordena $_n$ opera en dos fases , en la primera se aceptan los números en in y se desplazan por $\overline{\text{down}}$, en la segunda se reciben las entradas en up y se proporciona la salida en $\overline{\text{out}}$.



6. Sintaxis y semántica de CCS.

Como ya se ha mencionado, las expresiones de CCS representan tanto programas como especificaciones.

La relación existente entre las expresiones de comportamiento y los árboles de sincronización se ha extendido al agregar al cálculo la comunicación de valores, a este último concepto corresponde la noción de árboles de comunicación que será revisada posteriormente.

En esta sección se introducen varias relaciones entre las expresiones de comportamiento (de hecho ya ha sido definida la equivalencia observacional) entre las que destacan la congruencia fuerte, la congruencia observacional y la equivalencia directa, todas ellas de ayuda en el estudio y análisis de las correspondencias existentes entre los diferentes términos del cálculo CCS.

6.1. Sintaxis.

Las expresiones de valor se construyen a paritr de

i. variables,

ii. símbolos de constante y símbolos de función de las funciones conocidas ,

iii. vectores de expresiones de valor ($E_1, \ldots, E_n$).

Nota. Se asume que el tratamiento de los tipos de valores y variables es el usualmete utilizado.

Como ya ha sido descrito , se utiliza un conjunto de etiquetas

$$\Lambda = \Delta \cup \overline{\Delta}$$

y un símbolo τ que denota a la acción no observable . También se considera una colección de identificadores que tienen preasignados los siguientes atributos :

i. un número de argumentos n(b) ,

ii. un género L(b).

Las expresiones de comportamiento se construyen utilizando los operadores siguientes :

forma	B "	L(B ")	FV(B ")
inacción	NIL	Φ	Φ
suma	B + B'	$L(B) \cup L(B')$	$FV(B) \cup FV(B')$
acción	$\alpha x_{1,\ldots,} x_n.B$	$L(B) \cup \{\alpha\}$	$FV(B) - \{x_{1,\ldots,} x_n\}$
	$\overline{\alpha} E_{1,\ldots,} E_n.B$	$L(B) \cup \{\overline{\alpha}\}$	$FV(B) \cup \bigcup_i FV(E_i)$
	$\tau.B$	$L(B)$	$FV(B)$
composición	B | B'	$L(B) \cup L(B')$	$FV(B) \cup FV(B')$
restricción	$B \setminus \alpha$	$L(B) \cup \{\alpha, \overline{\alpha}\}$	$FV(B)$
re - etiquetamiento	B [S]	$S(L(B))$	$FV(B)$
identificador	$b(E_{1,\ldots,} E_{n(b)})$	$L(b)$	$\bigcup_i FV(E_i)$
condicional	<u>if</u> E <u>then</u> B <u>else</u> B'	$L(B) \cup L(B')$	$FV(E) \cup FV(B) \cup FV(B')$

Notación. FV(B) denota a las variables libres del comportamiento B.

Las precedencias de operadores están regidas por

$$\left\{ \begin{array}{l} \text{restricción} \\ \text{re - etiquetamiento} \end{array} \right\} \quad > \quad \text{acción} \quad > \quad \text{composición} \quad > \quad \text{suma}$$

6.2. Semántica por derivaciones.

Para cada $\mu \in \Lambda \cup \{\tau\}$ y v un valor del tipo asociado a μ, se define la relación binaria $B \xrightarrow{\mu v} B'$ que se lee B produce B ' mediante μv.

La colección de relaciones $\{ \xrightarrow{\mu v} ; \mu \in \Lambda \cup \{\tau\}$, v un valor asociado a μ } se definen por inducción sobre la estructura de las expresiones de comportamiento , de lo cual se obtiene que las acciones atómicas de expresiones compuestas pueden ser inferidas a partir de las acciones atómicas de sus componentes .

Nota. La presentación se realiza a través de reglas de reducción $\dfrac{\alpha}{\beta}$ (de α se puede inferir β).

1. Inacción.

NIL no tiene acciones atómicas .

2. Suma.

Las acciones atómicas de una suma son exactamente las acciones atómicas de los sumandos .

i) $\dfrac{B_1 \xrightarrow{\mu v} B_1'}{B_1 + B_2 \xrightarrow{\mu v} B_1'}$ ii) $\dfrac{B_2 \xrightarrow{\mu v} B_2'}{B_1 + B_2 \xrightarrow{\mu v} B_2'}$

3. Acción.

i) $\alpha x_1, \dots, x_n.B \xrightarrow{\alpha(v_1, \dots, v_n)} B\{v_1 / x_1, \dots, v_n / x_n\}$

ii) $\overline{\alpha}\, v_1, \dots, v_n .B \xrightarrow{\overline{\alpha}\,(v_1, \dots, v_n)} B$

iii) $\tau . B \xrightarrow{\tau} B$

Nota. El axioma i) es para todos los vectores $(v_{1..}, v_n)$, en cambio ii) es únicamente para el vector calificado por $\overline{\alpha}$.

4. Composición.

i)
$$\frac{B_1 \xrightarrow{\mu v} B_1'}{B_1 \mid B_2 \xrightarrow{\mu v} B_1' \mid B_2}$$

ii)
$$\frac{B_2 \xrightarrow{\mu v} B_2'}{B_1 \mid B_2 \xrightarrow{\mu v} B_1 \mid B_2'}$$

iii)
$$\frac{B_1 \xrightarrow{\lambda v} B_1' \quad B_2 \xrightarrow{\overline{\lambda} v} B_2'}{B_1 \mid B_2 \xrightarrow{\tau} B_1' \mid B_2'}$$

5.Restricción.

$$\frac{B \xrightarrow{\mu v} B'}{B \setminus \alpha \xrightarrow{\mu v} B' \setminus \alpha} \qquad \mu \notin \{\alpha, \overline{\alpha}\}$$

6.Re - etiquetamiento .

$$\frac{B \xrightarrow{\mu v} B'}{B\,[S] \xrightarrow{(S\mu)v} B'[S]} \qquad (S\tau = \tau)$$

7. Identificador.

Sea $b(x_1, \ldots, x_{n(b)}) \Leftarrow B_b$. $FV(B_b) \subseteq \{ x_1, \ldots, x_{n(b)} \}$ un identificador.

$$\frac{B_b \{ v_1 / x_1 , \ldots, v_{n(b)} / x_{n(b)} \} \xrightarrow{\mu v} B'}{b(v_1 , \ldots, v_{n(b)}) \xrightarrow{\mu v} B'}$$

8. Condicional.

i) $$\frac{B_1 \xrightarrow{\mu v} B_1'}{\text{if true then } B_1 \text{ else } B_2 \xrightarrow{\mu v} B_1'}$$

ii) $$\frac{B_2 \xrightarrow{\mu v} B_2'}{\text{if false then } B_1 \text{ else } B_2 \xrightarrow{\mu v} B_2'}$$

Nota. Para todas las expresiones de valor se asume una evaluación automática .

Nota. Para un identificador de comportamiento b definido mediante

$$b(x_1 , \ldots, x_{n(b)}) \Leftarrow B_b$$

se dice que b no está custodiado en B si existe una ocurrencia de b en B que no esté prefijada por un guardia. En la definición

$$b(x) \Leftarrow \bar{\alpha}\, x.NIL + b(x+1)$$

b no está custodiado en B lo cual construye una secuencia $b_1 b_1 b_1 \ldots$ que no conduce a algo razonable . Si

$$b(x) \Leftarrow \bar{\alpha}\, x.NIL + \tau.b(x+1)$$

entonces b sí está custodiado en B .

Para el correcto tratamiento de los géneros se requiere que $L(B_b) \subseteq L(b)$.

Definición .

Sea $b(x_1, ..., x_{n(b)}) \Leftarrow B_b$, se dice que b tiene una custodia bien definida si

i. b está custodiado en B

ii. $L(B_b) \subseteq L(b)$.

En forma general , se asume que los términos de CCS están construídos mediante custodias bien definidas lo que implica la finitud del conjunto de comportamientos derivados de un término dado .

Nota. Dado $B : L(B)$ un comportamiento de género $L(B)$, por conveniencia se declara que si $B : L$ y $L \subseteq M$ entonces $B : M$.

Definición .

Un programa es una expresión de comportamiento cerrada (i.e. sin variables libres).

El significado de una expresión de comportamiento B que contiene variables libres está determinado por los significados de las substituciones $B \{ \tilde{v} / \tilde{x} \}$ para todos los vectores de valor $\tilde{v}$. Es en este marco de substitución que debe enfocarse el axioma de acción ii) el cual está dado en términos de un vector de valores $\tilde{v}$ y no en térmnos de un vector de expresiones generales $\tilde{E} = (E_1 ,..., E_n)$.

Nota. Dos resultados directos de los axiomas y reglas de inferencia presentados son

i. Si $B \xrightarrow{\mu v} B'$,$B : L$ entonces $\mu \in L \cup \{\tau\}$ y $B' : L$

ii. Si B es un programa y $B \xrightarrow{\mu v} B'$ entonces B ' también es un programa .

6.3. Equivalencia directa.

Esta subsección corresponde sólo a programas. Para analizar y responder a la pregunta de cuáles programas poseen las mismas derivaciones se presentan a continuación varias definiciones y varios teoremas.

Definición .

Sean B_1, B_2 expresiones en CCS. $B_1 \equiv B_2$ (B_1 y B_2 son directamente equivalentes) sii $\forall\ \mu \in \Lambda \cup \{\tau\}$, v un valor del tipo asociado a μ y B un comportamiento

$$B_1 \xrightarrow{\mu v} B \quad \Leftrightarrow \quad B_2 \xrightarrow{\mu v} B .$$

Nota. La equivalencia directa no es una relación de congruencia . Por ejemplo, si B_1 y B_2 son B y $B + NIL$ respectivamente, entonces $B_1 \equiv B_2$ pero

$$\alpha.NIL \mid B_1 \xrightarrow{\alpha} NIL \mid B_1$$

$$\alpha.NIL \mid B_2 \xrightarrow{\alpha} NIL \mid B_2$$

no son idénticos

Teorema de Equivalencias Directas .

Las siguientes equivalencias directas son ciertas :

(suma $\equiv$)

i. $B_1 + B_2 \equiv B_2 + B_1$

ii. $B_1 + (B_2 + B_3) \equiv (B_1 + B_2) + B_3$

iii. $B + NIL \equiv B$

iv. $B + B \equiv B$

(acción $\equiv$)

$\alpha \tilde{x} .B \equiv \alpha \tilde{y}.B \{\tilde{y} / \tilde{x}\}$ (cambio de variables acotadas , $\tilde{y}$ son variables distintas que no aparecen en B.

(restricción $\equiv$)

i. $NIL \setminus \beta \equiv NIL$

ii. $(B_1 + B_2) \setminus \beta \equiv B_1 \setminus \beta + B_2 \setminus \beta$

iii. $(g.B) \setminus \beta \equiv \begin{cases} NIL & \text{si } \beta = \text{nombre}(g) \\ g.B \setminus \beta & \text{otro caso} \end{cases}$

(re - etiquetamiento $\equiv$)

$$\text{i. } NIL[S] \equiv NIL$$

$$\text{ii. } (B_1 + B_2)[S] \equiv B_1[S] + B_2[S]$$

$$\text{iii. } (g.B)[S] \equiv Sg.B[S]$$

(composición $\equiv$)

Sean $B = \sum g.B_i'$ y $C = \sum g.C_i'$,

$$B \mid C \equiv \sum\{ g.(B' \mid C) \text{ ; } g.B' \text{ un sumando de } B \}$$
$$+ \sum\{ g.(B \mid C') \text{ ; } g.C' \text{ un sumando de } C \}$$
$$+ \sum\{\tau.(B'\{\tilde{v} / \tilde{x}\} \mid C') \text{ ; } \alpha\tilde{x}.B' \text{ un sumando de } B \text{ y } \overline{\alpha}\tilde{x}.C' \text{ un sumando de } C \}$$
$$+ \sum\{\tau.(B' \mid C'\{\tilde{v} / \tilde{x}\}) \text{ ; } \overline{\alpha}\tilde{x}.B' \text{ un sumando de } B \text{ y } \alpha\tilde{x}.C' \text{ un sumando de } C \}$$

(identificador $\equiv$)

Sea $b(\tilde{x}) \Leftarrow B_b$ un identificador

$$b(\tilde{v}) \equiv B_b\{\tilde{v} / \tilde{x}\}$$

(condicional $\equiv$)

$$\text{i. if true then } B_1 \text{ else } B_2 \equiv B_1$$

$$\text{ii. if false then } B_1 \text{ else } B_2 \equiv B_2$$

Notación. g representa un guardia arbitrario entre $\alpha\tilde{x}$, $\overline{\alpha}\tilde{E}$, τ y Sg representa un re - etiquetamiento dado por

$$S(\alpha\tilde{x}) = (S\alpha)\tilde{x}$$

$$S(\overline{\alpha}\tilde{E}) = (S\overline{\alpha})\tilde{E}$$

$$S\tau = \tau$$

Demostración .

(suma $\equiv$)

i. $B_1 + B_2 \equiv B_2 + B_1$

Si $B_1 + B_2 \xrightarrow{\mu v} B$ entonces $B_1 \xrightarrow{\mu v} B$ o $B_2 \xrightarrow{\mu v} B$, de donde $B_2 + B_1 \xrightarrow{\mu v} B$. Similarmente en el otro sentido .

ii. $B_1 + (B_2 + B_3) \equiv (B_1 + B_2) + B_3$

Si $B_1 + (B_2 + B_3) \xrightarrow{\mu v} B$ entonces $B_1 \xrightarrow{\mu v} B$ o $(B_2 + B_3) \xrightarrow{\mu v} B$ y $\therefore$ $B_1 \xrightarrow{\mu v} B$ o $B_2 \xrightarrow{\mu v} B$ o $B_3 \xrightarrow{\mu v} B$. De esto último se puede inferir que $(B_1 + B_2) + B_3 \xrightarrow{\mu v} B$.

iii. $B + NIL \equiv B$

Si $B + NIL \xrightarrow{\mu v} B'$, como NIL no tiene acciones atómicas entonces $B \xrightarrow{\mu v} B'$. En el otro sentido el resultado es inmediato .

iv. $B + B \equiv B$

$$B + B \xrightarrow{\mu v} B' \quad \Leftrightarrow \quad B \xrightarrow{\mu v} B'$$

(acción $\equiv$)

$\alpha\,\tilde{x}.B \equiv \alpha\,\tilde{y}.B\,\{\tilde{y}/\tilde{x}\}$ (cambio de variables acotadas , $\tilde{y}$ son variables distintas y que no aparecen en B . Se observa que $\alpha\,\tilde{x}.B \xrightarrow{\alpha\tilde{v}} B\,\{\tilde{v}/\tilde{x}\}$ con $\tilde{v}$ arbitrario es el único evento posible y también se tiene que

$\alpha\,\tilde{y}.B\,\{\tilde{y}/\tilde{x}\} \xrightarrow{\alpha\tilde{v}} B\{\tilde{y}/\tilde{x}\}\{\tilde{v}/\tilde{y}\} = B\,\{\tilde{v}/\tilde{x}\}$. En el otro sentido es la misma idea .

(restricción $\equiv$)

i. $NIL \backslash \beta \equiv NIL$

$NIL \backslash \beta$ no posee acciones atómicas ni tampoco NIL .

ii. $(B_1+B_2)\setminus\beta \equiv B_1\setminus\beta + B_2\setminus\beta$

Si $(B_1+B_2)\setminus\beta \xrightarrow{\mu\tilde{v}} B' = B\setminus\beta$ entonces $B_1+B_2 \xrightarrow{\mu\tilde{v}} B$ y $\beta \neq \mu$, de donde $B_1 \xrightarrow{\mu\tilde{v}} B$ o $B_2 \xrightarrow{\mu\tilde{v}} B$. Finalmente $B_1\setminus\beta \xrightarrow{\mu\tilde{v}} B\setminus\beta$ o $B_2\setminus\beta \xrightarrow{\mu\tilde{v}} B\setminus\beta$ y por lo tanto $B_1\setminus\beta + B_2\setminus\beta \xrightarrow{\mu\tilde{v}} B\setminus\beta \equiv B'$.

iii. $(g.B)\setminus\beta \equiv \begin{cases} NIL & \text{si } \beta = \text{nombre}(g) \\ g.B\setminus\beta & \text{otro caso} \end{cases}$

Caso 1. $g = \alpha\tilde{x}$.

Si β = nombre(g) entonces $(\alpha\tilde{x}.B)\setminus\alpha$ no tiene acciones ni tampoco NIL . Si $\beta \neq$ nombre(g) entonces $\alpha\tilde{x}.B \xrightarrow{\alpha\tilde{v}} B\{\tilde{v}/\tilde{x}\}$ con $\tilde{v}$ arbitrario es el único evento posible , se sigue que

$$(\alpha\tilde{x}.B)\setminus\beta \xrightarrow{\alpha\tilde{v}} B\{\tilde{v}/\tilde{x}\}\setminus\beta = (B\setminus\beta)\{\tilde{v}/\tilde{x}\}$$

y como $\alpha\tilde{x}.B\setminus\beta \xrightarrow{\alpha\tilde{v}} (B\setminus\beta)\{\tilde{v}/\tilde{x}\}$ se obtiene el resultado .

Caso 2. $g = \overline{\alpha}\tilde{v}$.

Si β = nombre(g) entonces $(\overline{\alpha}\tilde{v}.B)\setminus\alpha$ no tiene acciones atómicas ni tampoco las tiene NIL . Si $\beta \neq$ nombre(g) entonces $\overline{\alpha}\tilde{v}.B \xrightarrow{\overline{\alpha}\tilde{v}} B$ es el único evento posible, de ello se sigue que $(\overline{\alpha}\tilde{v}.B)\setminus\beta \xrightarrow{\overline{\alpha}\tilde{v}} B\setminus\beta$. Como $\overline{\alpha}\tilde{v}.(B\setminus\beta) \xrightarrow{\overline{\alpha}\tilde{v}} B\setminus\beta$ se obtiene el resultado .

Caso 3. $g = \tau$.

$(\tau.B) \setminus \beta \xrightarrow{\tau} B' = B'' \setminus \beta$ es el único evento posible , de donde

$\tau.B \xrightarrow{\tau} B'' = B$ ya que $\tau.B \xrightarrow{\tau} B$. Además $\tau.(B \setminus \beta) \xrightarrow{\tau} B \setminus \beta = B'$

lo cual prueba el resultado .

(re - etiquetamiento $\equiv$)

i. $NIL[S] \equiv NIL$

$NIL[S]$ no tiene acciones atómicas ni tampoco NIL .

ii. $(B_1 + B_2)[S] \equiv B_1[S] + B_2[S]$

Si $(B_1 + B_2)[S] \xrightarrow{\mu v} B' = B[S]$ con $\mu = S(\mu')$ entonces

$B_1 + B_2 \xrightarrow{\mu' v} B$, por lo tanto $B_1 \xrightarrow{\mu' v} B$ o $B_2 \xrightarrow{\mu' v} B$

de donde $B_1[S] \xrightarrow{\mu v} B[S]$ o $B_2[S] \xrightarrow{\mu v} B[S]$, finalmente

$B_1[S] + B_2[S] \xrightarrow{\mu v} B[S] = B'$. La misma idea se aplica en el otro sentido .

iii. $(g.B)[S] \equiv Sg.B[S]$

Caso 1. $g = \alpha\tilde{x}$

Se tiene que $\alpha\tilde{x}.B \xrightarrow{\alpha\tilde{v}} B\{\tilde{v} / \tilde{x}\}$ con $\tilde{v}$ arbitrario es el único evento posible y por lo tanto

$$(\alpha\tilde{x}.B)[S] \xrightarrow{S(\alpha)\tilde{v}} B\{\tilde{v} / \tilde{x}\}[S] = B[S]\{\tilde{v} / \tilde{x}\} .$$

Como $(S\alpha)\tilde{x}.B[S] \xrightarrow{S(\alpha)\tilde{v}} B[S]\{\tilde{v} / \tilde{x}\}$ tiene el mismo evento se obtiene el resultado .

Caso 2. $g = \overline{\alpha}\tilde{v}$.

Se tiene que $\overline{\alpha}\tilde{v}.B \xrightarrow{\overline{\alpha}\tilde{v}} B$ y por lo tanto

$$(\overline{\alpha}\tilde{v}.B)[S] \xrightarrow{S(\overline{\alpha})\tilde{v}} B[S] \text{ es la única derivación posible .}$$

Como $(S\overline{\alpha})\tilde{v}.B[S] \xrightarrow{S(\overline{\alpha})\tilde{v}} B[S]$ el resultado queda demostrado .

Caso 3. $g = \tau$.

De $\tau.B \xrightarrow{\tau} B$ se obtiene que $(\tau.B)|S| \xrightarrow{\tau} B|S|$ $(S\tau = \tau)$. Como

$Sg.B|S| = \tau.B|S| \xrightarrow{\tau} B|S|$ se obtiene el resultado .

(composición $\blacksquare$)

Sean $B = \sum g.B_i'$ y $C = \sum g.C_i'$,

$$
\begin{aligned}
B \mid C = & \sum\{ g.(B' \mid C) ;\ g.B' \text{ un sumando de } B \} \\
& + \sum\{ g.(B \mid C') ;\ g.C' \text{ un sumando de } C \} \\
& + \sum\{\tau.(B'\{\tilde{v} / \tilde{x}\} \mid C') ;\ \alpha\tilde{x}.B' \text{ un sumando de } B \text{ y } \overline{\alpha}\tilde{x}.C' \text{ un sumando de } C \} \\
& + \sum\{\tau.(B' \mid C'\{\tilde{v} / \tilde{x}\}) ;\ \overline{\alpha}\tilde{x}.B' \text{ un sumando de } B \text{ y } \alpha\tilde{x}.C' \text{ un sumando de } C \}
\end{aligned}
$$

$B \mid C \blacksquare \Sigma + \Sigma + \Sigma + \Sigma$

Sea $B \mid C \xrightarrow{\mu v} D$ y $\Sigma + \Sigma + \Sigma + \Sigma = X$.

Caso 1. $B \xrightarrow{\mu v} B''$ y $D = B'' \mid C$

En este caso B debe poseer un sumando $g.B'$ tal que $g.B' \xrightarrow{\mu v} B''$ y por lo tanto $g.(B' \mid C) \xrightarrow{\mu v} B'' \mid C$. Finalmente se tiene que $X \xrightarrow{\mu v} B'' \mid C = D$.

Caso 2. $C \xrightarrow{\mu v} C''$ y $D = B \mid C''$.

Similar al caso anterior .

Caso 3. $B \xrightarrow{\alpha\tilde{u}} B''$ y $C \xrightarrow{\overline{\alpha}\tilde{u}} C'$ y $\mu v = \tau$ y $D = B'' \mid C'$.

Para estas condiciones B debe de poseer un sumando $\alpha\tilde{x}.B'$ con $B'' = B'\{\tilde{u} / \tilde{x}\}$ y C debe de poseer un sumando $\overline{\alpha}\tilde{u}.C'$. Ya que X tiene un sumando $\tau.(B'\{\tilde{u} / \tilde{x}\} \mid C')$ resulta finalmente que $X \xrightarrow{\tau} B'' \mid C' = D$.

Caso 4. $B \xrightarrow{\overline{\alpha}\tilde{u}} B'$ y $C \xrightarrow{\alpha\tilde{u}} C''$ y $\mu v = \tau$ y $D = B' \mid C''$.

Similar al caso 3.

(identificador $\equiv$)

Sea $b(\tilde{x}) \Leftarrow B_b$ un identificador , entonces

$$b(\tilde{v}) \equiv B_b\{\tilde{v} / \tilde{x}\}$$

De $b(\tilde{v}) \xrightarrow{\mu v} B'$ se sigue que $B_b\{\tilde{v} / \tilde{x}\} \xrightarrow{\mu v} B'$ por la regla de inferencia de identificadores .

(condicional $\equiv$)

i. if true then B_1 else $B_2 \equiv B_1$

Si (if true then B_1 else B_2)$\xrightarrow{\mu v} B$ entonces se debe tener que $B_1 \xrightarrow{\mu v} B$ e inversamente por la regla de inferencia sobre condicionales .

ii. if false then B_1 else $B_2 \equiv B_2$

El mismo procedimiento de i .

6.4. Congruencia de programas.

Con el objetivo de presentar una equivalencia congruencial de programas, a continuación se define otra relación de equivalencia entre programas que resulta ser más fuerte que la equivalencia observacional ya que no permite que acciones τ se entrelacen con las acciones observables en forma arbitraria.

Análogamente a la definición de equivalencia observacional , la siguiente equivalencia fuerte (denotada por $B \sim C$) se introduce mediante una secuencia decreciente $\sim_0$, $\sim_1$, $\sim_2$, ... , $\sim_k$, ... de relaciones de equivalencia .

Definición .

Sean B,C términos de CCS, $\mu \in \Lambda \cup \{\tau\}$.

i. $B \sim_0 C$ siempre es cierto.

ii. $B \sim_{k+1} C$ sii $\forall \mu, v$

a) Si $B \xrightarrow{\mu v} B'$ entonces $\exists C'$ tal que $C \xrightarrow{\mu v} C'$ y $B' \sim_k C'$

b) Si $C \xrightarrow{\mu v} C'$ entonces $\exists B'$ tal que $B \xrightarrow{\mu v} B'$ y $B' \sim_k C'$

iii. $B \sim C$ sii $\forall k \geq 0 \ B \sim_k C$.

Nota. $\sim \ = \bigcap_k \sim_k$ y $\sim_k$ es de equivalencia .

Nota. Si $B \equiv C$ entonces $B \sim_k C \ \forall k \geq 0$, ya que si $B \xrightarrow{\mu v} B'$, como $B \equiv C$ se tiene que $C \xrightarrow{\mu v} B'$ y claramente $B' \sim_k B' \ \forall k \geq 0$.

De este resultado se concluye que la partición inducida por $\equiv$ refina a la inducida por $\sim$.

Teorema .

La relación $\sim$ es de congruencia , más precisamente si $B_1 \sim B_2$ entonces

i. $B_1 + C \sim B_2 + C$, $C + B_1 \sim C + B_2$

ii. $\overline{\alpha}\tilde{v}.B_1 \sim \overline{\alpha}\tilde{v}.B_2$, $\tau.B_1 \sim \tau.B_2$

iii. $B_1 \mid C \sim B_2 \mid C$, $C \mid B_1 \sim C \mid B_2$

iv. $B_1 \setminus \alpha \sim B_2 \setminus \alpha$

v. $B_1[S] \sim B_2[S]$

vi. Si $B_1\{\tilde{v} / \tilde{x}\} \sim B_2\{\tilde{v} / \tilde{x}\}$ $\forall \tilde{v}$ entonces $\alpha\tilde{x}.B_1 \sim \alpha\tilde{x}.B_2$

Demostración .

i. $B_1 + C \sim B_2 + C$, $C + B_1 \sim C + B_2$

Se prueba por inducción sobre k que $B_1 \sim_k B_2$ implica $B_1 + C \sim_k B_2 + C$.

Para $k = 0$ es inmediato . Suponiendo $B_1 \sim_{k+1} B_2$ y $B_1 + C \xrightarrow{\mu\tilde{v}} D_1$ se desea encontrar D_2 tal que $B_2 + C \xrightarrow{\mu\tilde{v}} D_2 \sim_k D_1$.

Caso 1. $B_1 \xrightarrow{\mu\bar{v}} D_1$

Ya que $B_1 \sim_{k+1} B_2$ se tiene $B_2 \xrightarrow{\mu\bar{v}} D_2 \sim_k D_1$ de donde

$B_2 + C \xrightarrow{\mu\bar{v}} D_2 \sim_k D_1$.

Caso 2. $C \xrightarrow{\mu\bar{v}} D_1$

Se tiene que $B_2 + C \xrightarrow{\mu\bar{v}} D_1 = D_2 \sim_k D_1$. El resto de la prueba es en forma simétrica .

ii. $\overline{\alpha}\bar{v}.B_1 \sim \overline{\alpha}\bar{v}.B_2$, $\tau.B_1 \sim \tau.B_2$

Se prueba por inducción sobre k que $B_1 \sim_k B_2$ implica $\overline{\alpha}\bar{v}.B_1 \sim_k \overline{\alpha}\bar{v}.B_2$.
Para $k = 0$ es inmediato . Supóngase que $B_1 \sim_{k+1} B_2$.

Se tiene que $\overline{\alpha}\bar{v}.B_1 \xrightarrow{\overline{\alpha}\bar{v}} B_1$ y $\overline{\alpha}\bar{v}.B_2 \xrightarrow{\overline{\alpha}\bar{v}} B_2 \sim_k B_1$ ya que $B_1 \sim_{k+1} B_2$, lo cual prueba el resultado .

La demostración de que $\tau.B_1 \sim_{k+1} \tau.B_2$ se obtiene de que $\tau.B_1 \xrightarrow{\tau} B_1$ y $\tau.B_2 \xrightarrow{\tau} B_2 \sim_k B_1$ $(B_1 \sim_{k+1} B_2)$.

En el otro sentido la demostración es simétrica .

iii. $B_1 \mid C \sim B_2 \mid C$, $C \mid B_1 \sim C \mid B_2$

Se prueba por inducción sobre k que $B_1 \sim_k B_2$ implica $B_1 \mid C \sim_k B_2 \mid C$.
Para $k = 0$ es inmediato . Suponiendo que $B_1 \sim_{k+1} B_2$ y $B_1 \mid C \xrightarrow{\mu\bar{v}} D_1$ se desea encontrar D_2 tal que $B_2 \mid C \xrightarrow{\mu\bar{v}} D_2 \sim_k D_1$.

Caso 1. $B_1 \xrightarrow{\mu\bar{v}} B_1'$ y $D_1 = B_1' \mid C$

Como $B_1 \sim_{k+1} B_2$ se tiene que $B_2 \xrightarrow{\mu\bar{v}} B_2' \sim B_1'$ de donde

$B_2 \mid C \xrightarrow{\mu\bar{v}} B_2' \mid C = D_2 \sim_k D_1 = B_1' \mid C$ por la hipótesis inductiva .

Caso 2. $C \xrightarrow{\mu\bar{v}} C'$ y $D_1 = B_1 \mid C'$

Se sigue que $B_2 \mid C \xrightarrow{\mu\bar{v}} B_2 \mid C' = D_2 \sim_k D_1 = B_1 \mid C'$ nuevamente por la hipótesis de inducción .

Caso 3. $B_1 \xrightarrow{\lambda\bar{u}} B_1'$ y $C \xrightarrow{\bar{\lambda}\bar{u}} C'$ y $\mu\bar{v} = \tau$ y $D_1 = B_1' \mid C'$.

Se sigue que $B_2 \xrightarrow{\lambda\bar{u}} B_2' \sim_k B_1'$ de donde

$B_2 \mid C \xrightarrow{\tau} B_2' \mid C = D_2 \sim_k D_1$ por la hipótesis de inducción .

Caso 4. $B_1 \xrightarrow{\bar{\lambda}\bar{u}} B_1'$ y $C \xrightarrow{\lambda\bar{u}} C'$ y $\mu\bar{v} = \tau$ y $D_1 = B_1' \mid C'$.

Es similar al caso anterior .

iv. $B_1 \setminus \alpha \sim B_2 \setminus \alpha$

Se prueba por inducción sobre k que $B_1 \sim_k B_2$ implica $B_1 \setminus \alpha \sim_k B_2 \setminus \alpha$.

Para k = 0 es inmediato . Suponiendo que $B_1 \sim_{k+1} B_2$ y $B_1 \setminus \alpha \xrightarrow{\mu\bar{v}} D_1$ se desea encontrar D_2 tal que $B_2 \setminus \alpha \xrightarrow{\mu\bar{v}} D_2 \sim_k D_1$.

Como $B_1 \setminus \alpha \xrightarrow{\mu\bar{v}} D_1 = D_1' \setminus \alpha$ se sigue que $B_1 \xrightarrow{\mu\bar{v}} D_1'$ $(\mu \neq \alpha)$

y por lo tanto $B_2 \xrightarrow{\mu\bar{v}} D_2' \sim D_1'$ $(B_1 \sim_k B_2)$ de donde

$B_2 \setminus \alpha \xrightarrow{\mu\bar{v}} D_2' \setminus \alpha = D_2 \sim_k D_1 = D_1' \setminus \alpha$ por la hipótesis de inducción .

v. $B_1[S] \sim B_2[S]$

Se prueba por inducción sobre k que $B_1 \sim_k B_2$ implica $B_1[S] \sim_k B_2[S]$.

Para k = 0 es inmediato . Suponiendo que $B_1 \sim_{k+1} B_2$ y

$B_1[S] \xrightarrow{\mu\bar{v}} D_1$ se desea encontrar D_2 tal que $B_2[S] \xrightarrow{\mu\bar{v}} D_2 \sim_k D_1$.

Como $B_1[S] \xrightarrow{\mu\bar{v}} D_1 = D_1'[S]$ $(\mu \neq S(\mu'))$ se sigue que

$B_1 \xrightarrow{\mu'\bar{v}} D_1'$, y por lo tanto $B_2 \xrightarrow{\mu'\bar{v}} D_2'$ $(B_1 \sim_k B_2)$

de donde $B_2[S] \xrightarrow{\mu\bar{v}} D_2'[S] = D_2 \sim_k D_1 = D_1'[S]$ por la hipótesis inductiva .

vi. Si $B_1\{\tilde v / \tilde x\} \sim B_2\{\tilde v / \tilde x\} \quad \forall \tilde v$ entonces $\alpha\tilde x.B_1 \sim \alpha\tilde x.B_2$

Se prueba por inducción sobre k que $B_1\{\tilde v / \tilde x\} \sim_k B_2\{\tilde v / \tilde x\} \quad \forall \tilde v$ implica $\alpha\tilde x.B_1 \sim_k \alpha\tilde x.B_2$.

Para $k = 0$ es inmediato. Súpongase que $B_1\{\tilde v / \tilde x\} \sim_{k+1} B_2\{\tilde v / \tilde x\} \; \forall \tilde v$, se tiene que $\alpha\tilde x.B_1 \xrightarrow{\alpha\tilde v} B_1\{\tilde v / \tilde x\}$ y $\alpha\tilde x.B_2 \xrightarrow{\alpha\tilde v} B_2\{\tilde v / \tilde x\} \sim_k B_1\{\tilde v / \tilde x\}$ lo cual prueba el resultado .

Nota. Si $B_1 \sim_k B_2$ entonces $\alpha\tilde v.B_1 \sim_{k+1} \alpha\tilde v.B_2$, de donde se concluye que $B_1 \sim B_2$ implica $\alpha\tilde v.B_1 \sim \alpha\tilde v.B_2$. En general, prefijar un comportamiento con un guardia incrementa el índice k tratándose de la congruencia fuerte .

Algunas relaciones adicionales de la congruencia fuerte se presentan en el siguiente teorema.

Teorema.

Las siguientes congruencias fuertes son ciertas .

(composición $\sim$)

i. $B_1 \mid B_2 \sim B_2 \mid B_1$

ii. $B_1 \mid (B_2 \mid B_3) \sim (B_1 \mid B_2) \mid B_3$

iii. $B \mid NIL \sim B$

(restricción $\sim$)

i. $B \setminus \alpha \sim B$ $\qquad$ ($B : L$ y $\alpha \notin$ nombre(L))

ii. $B \setminus \alpha \setminus \beta \sim B \setminus \beta \setminus \alpha$

iii. $(B_1 \mid B_2) \setminus \alpha \sim B_2 \setminus \alpha \mid B_1 \setminus \alpha$ $\qquad$ ($B_1 : L_1$, $B_2 : L_2$, $\alpha \notin$ nomre($L_1 \cap \overline{L_2}$))

(re - etiquetamiento $\sim$)

i. $B[I] \sim B$ $\quad$ ($I : L \rightarrow L$ la función identidad)

ii. $B[S] \sim B[S']$ $\quad$ ($B : L$, $S \uparrow L = S' \uparrow L$)

iii. $B[S][S'] \sim B[S' o S]$

iv. $B[S] \setminus \beta \sim B \setminus \alpha [S]$ $\quad$ (β = nombre($S(\alpha)$))

v. $(B_1 \mid B_2)[S] \sim B_1[S] \mid B_2[S]$

Demostración .

(composición $\sim$)

i. $B_1 \mid B_2 \sim B_2 \mid B_1$

Se prueba por inducción sobre k que $B_1 \mid B_2 \sim_k B_2 \mid B_1$.

Para k = 0 es inmediato . Supóngase $\sim_k$ y $B_1 \mid B_2 \xrightarrow{\mu\tilde{v}} D_1$,

se requiere D_2 tal que $B_2 \mid B_1 \xrightarrow{\mu\tilde{v}} D_2 \sim D_1$.

Caso 1. $B_1 \xrightarrow{\mu\tilde{v}} B_1'$ y $D_1 = B_1' \mid B_2$.

Se tiene que $B_2 \mid B_1 \xrightarrow{\mu\tilde{v}} B_2 \mid B_1' = D_2 \sim_k B_1' \mid B_2 = D_1$ por la hipótesis de inducción .

Caso 2. $B_2 \xrightarrow{\mu\tilde{v}} B_2'$ y $D_1 = B_1 \mid B_2'$.

Es simétrico al caso anterior .

Caso 3. $B_1 \xrightarrow{\lambda\tilde{v}} B_1'$ y $B_2 \xrightarrow{\bar{\lambda}\tilde{v}} B_2'$ y $\mu\tilde{v} = \tau$ y $D_1 = B_1' \mid B_2'$.

Se tiene que $B_2 \mid B_1 \xrightarrow{\tau} B_2' \mid B_1' = D_2 \sim_k B_1' \mid B_2' = D_1$

Caso 4. $B_1 \xrightarrow{\bar{\lambda}\tilde{v}} B_1'$ y $B_2 \xrightarrow{\lambda\tilde{v}} B_2'$ y $\mu\tilde{v} = \tau$ y $D_1 = B_1' \mid B_2'$.

Simétrico al caso anterior .

ii. $B_1 \mid (B_2 \mid B_3) \sim (B_1 \mid B_2) \mid B_3$

Se prueba por inducción sobre k que $B_1 \mid (B_2 \mid B_3) \sim_k (B_1 \mid B_2) \mid B_3$.

Para k = 0 es inmediato . Supóngase $B_1 \mid (B_2 \mid B_3) \xrightarrow{\mu\tilde{v}} D_1$, se requiere D_2 tal que $(B_1 \mid B_2) \mid B_3 \xrightarrow{\mu\tilde{v}} D_2 \sim_k D_1$.

Caso 1. $B_1 \xrightarrow{\mu\bar{v}} B_1'$ y $D = B_1' \mid (B_2 \mid B_3)$.

Se tiene que $(B_1 \mid B_2) \mid B_3 \xrightarrow{\mu\bar{v}} (B_1' \mid B_2) \mid B_3 = D_2 \sim_k D_1$ por la hipótesis de inducción .

Caso 2. $B_2 \mid B_3 \xrightarrow{\mu\bar{v}} C$ y $D_1 = B_1 \mid C$.

Subcasos:

a) $B_2 \xrightarrow{\mu\bar{v}} B_2'$ y $C = B_2' \mid B_3$ $(D_1 = B_1 \mid (B_2' \mid B_3))$.

Se tiene que $B_1 \mid B_2 \xrightarrow{\mu\bar{v}} B_1 \mid B_2'$ de donde

$(B_1 \mid B_2) \mid B_3 \xrightarrow{\mu\bar{v}} (B_1 \mid B_2') \mid B_3 = D_2 \sim_k D_1$ por la hipótesis inductiva .

b) $B_3 \xrightarrow{\mu\bar{v}} B_3'$ y $C = B_2 \mid B_3'$.

Similar al subcaso a).

c) $B_2 \xrightarrow{\lambda\bar{u}} B_2'$ y $B_3 \xrightarrow{\bar{\lambda}\bar{u}} B_3'$ y $C = B_2' \mid B_3'$ y $\mu\bar{v} = \tau$.

(por lo tanto $D = B_1 \mid (B_2' \mid B_3')$).

Se tiene que $B_1 \mid B_2 \xrightarrow{\lambda\bar{u}} B_1 \mid B_2'$ de donde

$(B_1 \mid B_2) \mid B_3 \xrightarrow{\tau} (B_1 \mid B_2') \mid B_3' = D_2 \sim_k D_1$ por la hipótesis inductiva .

Caso 3. $B_1 \xrightarrow{\lambda\bar{u}} B_1'$ y $B_2 \mid B_3 \xrightarrow{\bar{\lambda}\bar{u}} C$ y $D = B_1' \mid C$ y $\mu\bar{v} = \tau$.

Subcasos.

a) $B_2 \xrightarrow{\bar{\lambda}\bar{u}} B_2'$ y $C = B_2' \mid B_3$ y $D = B_1' \mid (B_2' \mid B_3)$

Se tiene $B_1 \mid B_2 \xrightarrow{\tau} B_1' \mid B_2'$ de donde

$(B_1 \mid B_2) \mid B_3 \xrightarrow{\tau} (B_1' \mid B_2') \mid B_3 = D_2 \sim_k D_1$ por la hipótesis inductiva .

b) $B_3 \xrightarrow{\bar{\lambda}\bar{u}} B_3'$ y $C = B_2 \mid B_3'$.

Es similar al caso anterior .

El resto de la prueba se obtiene por simetría .

iii. $B \mid NIL \sim B$

Se prueba por inducción sobre k que $B \mid NIL \sim_k B$.

Para k = 0 es inmediato. Supóngase $B \mid NIL \xrightarrow{\mu\tilde{v}} D_1$, se desea encontrar D_2 tal que $B \xrightarrow{\mu\tilde{v}} D_2 \sim_k D_1$.

Como NIL carece de acciones atómicas entonces $B \mid NIL \xrightarrow{\mu\tilde{v}} D_1$ implica $B \xrightarrow{\mu\tilde{v}} D_1 = D_2$ (claramente $D_1 \sim_k D_2$).

(restricción $\sim$)

i. $B \setminus \alpha \sim B$ $\quad$ $(B : L$ y $\alpha \notin$ nombre$(L))$

Es inmediato del hecho $\alpha \notin$ nombre(L).

ii. $B \setminus \alpha \setminus \beta \sim B \setminus \beta \setminus \alpha$

Se prueba por inducción sobre k que $B \setminus \alpha \setminus \beta \sim_k B \setminus \beta \setminus \alpha$.

Para k = 0 es inmediato. Supóngase $\sim_k$ y $B \setminus \alpha \setminus \beta \xrightarrow{\mu\tilde{v}} D_1$, se requiere D_2 tal que $B \setminus \beta \setminus \alpha \xrightarrow{\mu\tilde{v}} D_2 \sim_k D_1$.

De $B \setminus \alpha \setminus \beta \xrightarrow{\mu\tilde{v}} D_1 = D_1' \setminus \alpha \setminus \beta$ implica $\mu \neq \alpha, \beta$ y $B \setminus \alpha \xrightarrow{\mu\tilde{v}} D_1' \setminus \alpha$ de donde $B \xrightarrow{\mu\tilde{v}} D_1'$. Finalmente se tiene $B \setminus \beta \xrightarrow{\mu\tilde{v}} D_1' \setminus \beta$ y $B \setminus \beta \setminus \alpha \xrightarrow{\mu\tilde{v}} D_1' \setminus \beta \setminus \alpha = D_2 \sim_k D_1$ por la hipótesis de inducción.

iii. $(B_1 \mid B_2) \setminus \alpha \sim B_2 \setminus \alpha \mid B_1 \setminus \alpha$ $\quad$ $(B_1 : L_1, B_2 : L_2, \alpha \notin$ nombre$(L_1 \cap \overline{L_2}))$

Se prueba por inducción sobre k que $(B_1 \mid B_2) \setminus \alpha \sim_k B_2 \setminus \alpha \mid B_1 \setminus \alpha$.

Para k = 0 es inmediato. Supóngase $\sim_k$ y $(B_1 \mid B_2) \setminus \alpha \xrightarrow{\mu\tilde{v}} D_1$, se requiere D_2 tal que $B_1 \setminus \alpha \mid B_2 \setminus \alpha \xrightarrow{\mu\tilde{v}} D_2 \sim_k D_1$.

De $(B_1 \mid B_2) \setminus \alpha \xrightarrow{\mu\tilde{v}} D_1 = D_1' \setminus \alpha$ $(\mu \neq \alpha)$ se tiene $B_1 \mid B_2 \xrightarrow{\mu\tilde{v}} D_1'$. Se presentan varios casos:

Caso 1. $B_1 \xrightarrow{\mu\bar{\nu}} B_1'$ y $D_1' = B_1' \mid B_2$.

$B_1 \setminus \alpha \xrightarrow{\mu\bar{\nu}} B_1' \setminus \alpha$ y por lo tanto

$B_1 \setminus \alpha \mid B_2 \setminus \alpha \xrightarrow{\mu\bar{\nu}} B_1' \setminus \alpha \mid B_2 \setminus \alpha = D_2 \sim_k (B_1' \mid B_2) \setminus \alpha = D_1' \setminus \alpha = D_1$

por la hipótesis de inducción.

Caso 2. $B_2 \xrightarrow{\mu\bar{\nu}} B_2'$ y $D_1' = B_1 \mid B_2'$.

Es simétrico al caso anterior .

Caso 3. $B_1 \xrightarrow{\lambda\bar{\nu}} B_1'$ y $B_2 \xrightarrow{\bar{\lambda}\bar{\nu}} B_2'$ y $\mu\bar{\nu} = \tau$ y $D_1' = B_1' \mid B_2'$.

Como $\alpha \notin$ nombre$(L_1 \cap \bar{L}_2)$ implica $\alpha \neq \lambda$ de donde

$B_1 \setminus \alpha \xrightarrow{\lambda\bar{\nu}} B_1' \setminus \alpha$ y $B_2 \setminus \alpha \xrightarrow{\bar{\lambda}\bar{\nu}} B_2' \setminus \alpha$, por lo tanto

$B_1 \setminus \alpha \mid B_2 \setminus \alpha \xrightarrow{\tau} B_1' \setminus \alpha \mid B_2' \setminus \alpha = D_2 \sim_k (B_1' \mid B_2') \setminus \alpha = D_1' \setminus \alpha = D_1$

por la hipótesis de inducción.

(re - etiquetamiento $\sim$)

i. $B[I] \sim B$ ($B : L$, $I : L \rightarrow L$ la función identidad)

ii. $B[S] \sim B[S']$ ($B : L$, $S \uparrow L = S' \uparrow L$)

iii. $B[S][S'] \sim B[S' o S]$

Estos resultados son inmediatos.

iv. $B[S] \setminus \beta \sim B \setminus \alpha[S]$ ($\beta =$ nombre$(S(\alpha))$)

Se prueba por inducción sobre k que $B[S] \setminus \beta \sim_k B \setminus \alpha[S]$.

Para $k = 0$ es inmediato. Supóngase $\sim_k$ cierto y $B[S] \setminus \beta \xrightarrow{\mu\bar{\nu}} D_1$,

se requiere D_2 tal que $B \setminus \alpha[S] \xrightarrow{\mu\bar{\nu}} D_2 \sim_k D_1$.

De $B[S] \setminus \beta \xrightarrow{\mu\bar{\nu}} D_1 = D_1'[S] \setminus \beta$ se tiene que $\mu \neq \beta$ y

$B[S] \xrightarrow{\mu\bar{\nu}} D_1'[S]$, por lo tanto $B \xrightarrow{\mu'\bar{\nu}} D_1'$ con $\mu = S(\mu')$.

Como S es inyectiva implica $\mu' \neq \alpha$ de donde

$B \setminus \alpha \xrightarrow{\mu'\tilde{v}} D_1' \setminus \alpha$ y finalmente $B \setminus \alpha[S] \xrightarrow{\mu\tilde{v}} D_1' \setminus \alpha[S] = D_2 \sim_k D_1$

por la hipótesis de inducción.

v. $(B_1 \mid B_2)[S] \sim B_1[S] \mid B_2[S]$

Se prueba por inducción sobre k que $(B_1 \mid B_2)[S] \sim_k B_1[S] \mid B_2[S]$.

Para $k = 0$ es inmediato. Supóngase $\sim_k$ y $(B_1 \mid B_2)[S] \xrightarrow{\mu\tilde{v}} D_1$,

se requiere D_2 tal que $B_1[S] \mid B_2[S] \xrightarrow{\mu\tilde{v}} D_2 \sim_k D_1$.

De $(B_1 \mid B_2)[S] \xrightarrow{\mu\tilde{v}} D_1 = D_1'[S]$ se obtiene $B_1 \mid B_2 \xrightarrow{\mu'\tilde{v}} D_1'$

con $\mu = S(\mu')$. Se presentan varios casos :

Caso 1. $B_1 \xrightarrow{\mu'\tilde{v}} B_1'$ y $D_1' = B_1' \mid B_2$.

Se sigue $B_1[S] \xrightarrow{\mu\tilde{v}} B_1'[S]$ y por lo tanto

$B_1[S] \mid B_2[S] \xrightarrow{\mu\tilde{v}} B_1'[S] \mid B_2[S] = D_2 \sim_k (B_1' \mid B_2)[S] = D_1'[S] = D_1$

por la hipótesis inductiva.

Caso 2. $B_2 \xrightarrow{\mu'\tilde{v}} B_2'$ y $D_1' = B_1 \mid B_2'$.

Es simétrico al caso anterior.

Caso 3. $B_1 \xrightarrow{\lambda\tilde{v}} B_1'$ y $B_2 \xrightarrow{\bar{\lambda}\tilde{v}} B_2'$ y $\mu\tilde{v} = \tau$ y $D_1' = B_1' \mid B_2'$.

Se tiene $B_1[S] \xrightarrow{S(\lambda)\tilde{v}} B_1'[S]$, $B_2[S] \xrightarrow{S(\bar{\lambda})\tilde{v}} B_2[S]$ y por lo tanto

$B_1[S] \mid B_2[S] \xrightarrow{\tau} B_1'[S] \mid B_2[S] = D_2 \sim_k (B_1' \mid B_2')[S] = D_1[S] = D_1$

por la hipótesis inductiva.

Por simetría se concluye el resultado.

Como siguiente resultado, se demuestra que la congruencia fuerte "satisface su definición" como se muestra a continuación. Este teorema está basado en la noción de custodias bien definidas para identificadores de comportamiento.

Teorema.

$B \sim C$ sii $\forall \mu \in \Lambda \cup \{\tau\}$, $\tilde{v}$

i. Si $B \xrightarrow{\mu\tilde{v}} B'$ entonces $\exists\, C'$ tal que $C \xrightarrow{\mu\tilde{v}} C'$ y $B' \sim C'$.

ii. Si $C \xrightarrow{\mu\tilde{v}} C'$ entonces $\exists\, B'$ tal que $B \xrightarrow{\mu\tilde{v}} B'$ y $B' \sim C'$.

Demostración.

$\Leftarrow$) $B' \sim C'$ implica $B' \sim_k C'$ $\forall k$, de i y ii se obtiene $B \sim_{k+1} C$ $\forall k$ y por lo tanto $B \sim C$ (prefijar con un guardia incrementa el valor del subíndice en $\sim_k$).

$\Rightarrow$) Ya que $B \sim_{k+1} C$ $\forall k$, si $B \xrightarrow{\mu\tilde{v}} B'$ entonces $\forall k$ $\exists\, C_k$ tal que $C \xrightarrow{\mu\tilde{v}} C_k$ y $B' \sim_k C_k$. El conjunto $\{ C' : C \xrightarrow{\mu\tilde{v}} C' \}$ es finito si los identificadores de comportamiento tienen custodias bien definidas, por lo tanto $\exists\, C'$ tal que $C \xrightarrow{\mu\tilde{v}} C'$ y $B' \sim_k C'$ para una infinidad de k, de donde $B' \sim_k C'$ $\forall k$ ya que las relaciones $\sim_k$ son decrecientes.

La prueba de ii. es similar.

Hasta este punto sólo se ha hecho referencia a programas, para extender los conceptos a expresiones de comportamiento conteniendo variables libres se ofrece la siguiente definición.

Definición.

Sea $\tilde{x}$ las variables libres contenidas en un comportamiento B_1, B_2 o ambos, entonces

$$B_1 \approx B_2 \quad \text{sii} \quad \forall \tilde{v} \quad B_1\{\tilde{v} / \tilde{x}\} \approx B_2\{\tilde{v} / \tilde{x}\}$$

$$B_1 \sim B_2 \quad \text{sii} \quad \forall \tilde{v} \quad B_1\{\tilde{v} / \tilde{x}\} \sim B_2\{\tilde{v} / \tilde{x}\}.$$

Nota.

Todos los resultados sobre equivalencias presentados de la congruencia fuerte y la equivalencia directa son ciertos sustituyendo $\tilde{v}$ por una expresión general $\tilde{E}$, a excepción de (composición $\approx$) que se puede reformular de la siguiente manera :

(composición $\equiv$)

Sean B y C sumas de guardias, entonces

$$
\begin{aligned}
B \mid C \equiv\ & \sum\{g.(B' \mid C)\ ;\ g.B' \text{ un sumando de } B\} \\
+\ & \sum\{g.(B \mid C')\ ;\ g.C' \text{ un sumando de } C\} \\
+\ & \sum\{\tau.(B'\{\tilde{v}/\tilde{x}\} \mid C')\ ;\ \alpha\tilde{x}.B' \text{ y } \overline{\alpha}\tilde{v}.C' \text{ sumandos de B y C respectivamente}\} \\
+\ & \sum\{\tau.(B' \mid C'\{\tilde{v}/\tilde{x}\})\ ;\ \overline{\alpha}\tilde{v}.B' \text{ y } \alpha\tilde{x}.C' \text{ sumandos de B y C respectivamente}\}
\end{aligned}
$$

bajo el supuesto de que en el primer término ninguna variable libre de C es acotada por g y ninguna variable libre de B es acotada por g en el segundo término .

A continuación se presenta el teorema de expansión para una composición de comportamientos .

Teorema .

Sea $B = (B_1 \mid ... \mid B_m) \setminus A$, B_i una suma de guardias. Entonces

$$
\begin{aligned}
B \sim\ & \sum\{g.((B_1 \mid ... \mid B_i' \mid ... \mid B_m) \setminus A)\ ;\ g.B_i' \text{ un sumando de } B_i,\ \text{nombre}(g) \notin A\} \\
+\ & \sum\{\tau.((B_1 \mid ... \mid B_i'\{\tilde{E}/\tilde{x}\} \mid ... \mid B_j' \mid ... \mid B_m) \setminus A)\ ; \\
& \qquad \alpha\tilde{x}.B_i' \text{ un sumando de } B_i,\ \overline{\alpha}\tilde{E}.B_j' \text{ un sumando de } B_j,\ i \neq j\}
\end{aligned}
$$

bajo el supuesto que en el primer término ninguna variable libre en B_k $(k \neq i)$ es acotada por g.

Demostración .

Por inducción sobre m .

Para m = 1 el resultado es directo . Supóngase

$$
\begin{aligned}
B_1 \mid ... \mid B_m \sim\ & \sum\{g.(B_1 \mid ... \mid B_i' \mid ... \mid B_m)\ ;\ g.B_i' \text{ de } B_i\ ,\ 1 \leq i \leq m\} \\
+\ & \sum\{\tau.(B_1 \mid ... \mid B_i'\{\tilde{E}/\tilde{x}\} \mid ... \mid B_j' \mid ... \mid B_m)\ ; \\
& \qquad \alpha\tilde{x}.B_i' \text{ de } B_i,\ \overline{\alpha}\tilde{E}.B_j' \text{ de } B_j,\ i,j \in \{1..m\},\ i \neq j\}
\end{aligned}
$$

y sea C el lado derecho de la equivalencia .

Aplicando el resultado para dos comportamientos se obtiene

$(B_1 \mid .. \mid B_m) \mid B_{m+1} \sim C \mid B_{m+1}$ ya que C y B_{m+1} corresponden a sumas de guardias.

$$
\begin{aligned}
C \mid B_{m+1} \sim \Big(& \sum\{g.(B_1 \mid ... \mid B_i' \mid ... \mid B_m) ;\ g.B_i' \text{ de } B_i ,\ 1 \le i \le m\} \\
& + \sum\{\tau.(B_1 \mid ... \mid B_i'\{\tilde{E}/\tilde{x}\} \mid ... \mid B_j' \mid ... \mid B_m) ; \\
& \qquad \alpha\tilde{x}.B_i' \text{ de } B_i,\ \bar{\alpha}\tilde{E}.B_j' \text{ de } B_j,\ i,j \in \{1..m\},\ i \ne j\} \Big) \ \Big|\ \sum g.B_{m+1}' \\
\sim \ & \sum\{g.(B_1 \mid ... \mid B_i' \mid ... \mid B_{m+1}) ;\ g.B_i' \text{ de } B_i ,\ 1 \le i \le m\} \\
& + \sum\{\tau.(B_1 \mid ... \mid B_i'\{\tilde{E}/\tilde{x}\} \mid ... \mid B_j' \mid ... \mid B_{m+1}) ; \\
& \qquad \alpha\tilde{x}.B_i' \text{ de } B_i,\ \bar{\alpha}\tilde{E}.B_j' \text{ de } B_j,\ i,j \in \{1..m\},\ i \ne j\} \\
& + \sum\{g.(C \mid B_{m+1}');\ g.B_{m+1}' \text{ de } B_{m+1}\} \\
& + \sum\{\tau.(B_1 \mid ... \mid B_i'\{\tilde{E}/\tilde{x}\} \mid ... \mid B_{m+1}') ; \\
& \qquad \alpha\tilde{x}.B_i' \text{ de } B_i,\ 1 \le i \le m,\ \bar{\alpha}\tilde{E}.B_{m+1}' \text{ de } B_{m+1}\} \\
& + \sum\{\tau.(B_1 \mid ... \mid B_i' \mid ... \mid B_m \mid B_{m+1}'\{\tilde{E}/\tilde{x}\}) ; \\
& \qquad \bar{\alpha}\tilde{E}.B_i' \text{ de } B_i,\ 1 \le i \le m,\ \alpha\tilde{x}.B_{m+1}' \text{ de } B_{m+1}\ \}
\end{aligned}
$$

Como $C \sim B_1 \mid .. \mid B_m$ por hipótesis inductiva y $\sim$ es una congruencia se obtiene que

$$
\begin{aligned}
\sim \ & \sum\{g.(B_1 \mid ... \mid B_i' \mid ... \mid B_{m+1}) ;\ g.B_i' \text{ de } B_i,\ 1 \le i \le m+1\} \\
& + \sum\{\tau.(B_1 \mid ... \mid B_i'\{\tilde{E}/\tilde{x}\} \mid ... \mid B_j' \mid ... \mid B_{m+1}) ; \\
& \qquad \alpha\tilde{x}.B_i' \text{ de } B_i,\ \bar{\alpha}\tilde{E}.B_j' \text{ de } B_j,\ i,j \in \{1..m+1\},\ i \ne j\}
\end{aligned}
$$

El resultado finalmente se obtiene utilizando propiedades generales de la operación \A.

Nota.

El hecho de que las variables libres no son acotadas por los guardias se ha utilizado en la aplicación de la expansión de dos comportamientos.

7. Equivalencia observacional y sus propiedades.

Esta sección está dedicada al estudio de la equivalencia observacional y de la congruencia observacional.

Notación.

i. $B \stackrel{\varepsilon}{\Longrightarrow} B'$ denota $B \stackrel{\tau^m}{\Longrightarrow} B'$, $m \geq 0$.

ii. $B \stackrel{\tau}{\Longrightarrow} B'$ denota $B \stackrel{\tau^m}{\Longrightarrow} B'$, $m > 0$.

iii. $B \stackrel{\mu v}{\Longrightarrow} B'$ denota $B \stackrel{\tau^m.\mu v.\tau^n}{\Longrightarrow} B'$, $m,n \geq 0$.

iv. $B \stackrel{\mu_1 v_1 \ldots \mu_k v_k}{\Longrightarrow} B'$ denota $B \stackrel{\tau^{m_0}.\mu_1 v_1.\tau^{m_1}\ldots\mu_k v_k.\tau^{m_k}}{\Longrightarrow} B'$, $k, m_0, \ldots, m_k \geq 0$.

La definición de la equivalencia observacional ($\approx$) para programas está dada en términos de una secuencia decreciente de equivalencias .

Definición.

Sean B, C dos programas en CCS.

i. $B \approx_0 C$ siempre es cierto.

ii. $B \approx_{k+1} C$ sii $\forall s \in (\Lambda \times V)^*$,

a) Si $B \stackrel{s}{\Longrightarrow} B'$ entonces $\exists C'$ tal que $C \stackrel{s}{\Longrightarrow} C'$ y $B' \approx_k C'$.

b) Si $C \stackrel{s}{\Longrightarrow} C'$ entonces $\exists B'$ tal que $B \stackrel{s}{\Longrightarrow} B'$ y $B' \approx_k C'$.

iii. $B \approx C$ sii $\forall k \geq 0$ $B \approx_k C$.

Si $B \overset{s}{\Longrightarrow} B_n$ mediante la derivación $B \xrightarrow{\mu_1 v_1} B_1 \rightarrow \;...\; \xrightarrow{\mu_n v_n} B_n$ (en donde algunos $\mu_i v_i$ pueden ser τ) entonces existen $C_1, C_2, ..., C_n$ tales que

$C \xrightarrow{\mu_1 v_1} C_1 \rightarrow \;...\; \xrightarrow{\mu_n v_n} C_n$ $\quad (C \overset{s}{\Longrightarrow} C_n)$ con $C_i \sim B_i$, $1 \leq i \leq n$.

En particular $B_n \sim C_n$ y por la hipótesis inductiva $B_n \approx_k C_n$.

En el otro sentido la prueba es simétrica .

Teorema .

La equivalencia observacional es una relación de congruencia para todas las operaciones de comportamiento excepto para la adición (+). Más precisamente si $B \approx_k C$ entonces

i. $\bar{\alpha}\tilde{v}.B \approx_k \bar{\alpha}\tilde{v}.C$

ii. $\tau.B \approx_k \tau.C$

iii. $B \mid D \approx_k C \mid D$

iv. $B \setminus \alpha \approx_k C \setminus \alpha$

v. $B[S] \approx_k C[S]$

vi. $B\{\tilde{v}/\tilde{x}\} \approx_k C\{\tilde{v}/\tilde{x}\}$ $\forall \tilde{v}$ implica $\alpha\tilde{x}.B \approx_k \alpha\tilde{x}.C$

Nota.

Como corolario de este teorema se obtiene que las mismas relaciones son ciertas suprimiendo los subíndices k .

Demostración .

i. $\bar{\alpha}\tilde{v}.B \approx_k \bar{\alpha}\tilde{v}.C$

Se demuestra por inducción sobre k que $B \approx_k C$ implica $\bar{\alpha}\tilde{v}.B \approx_k \bar{\alpha}\tilde{v}.C$.

Para $k = 0$ es inmediato . Suponga $B \approx_{k+1} C$ y $\bar{\alpha}\tilde{v}.B \overset{s}{\Longrightarrow} B'$.

Caso 1. $s = \varepsilon$ y $\widetilde{\alpha v}.B \stackrel{s}{\Longrightarrow} \widetilde{\alpha v}.B$.

Se sigue que $\widetilde{\alpha v}.C \stackrel{s}{\Longrightarrow} \widetilde{\alpha v}.C$ de donde $\widetilde{\alpha v}.B \approx_k \widetilde{\alpha v}.C$ aplicando la hipótesis inductiva y el hecho de que $B \approx_{k+1} C$.

Caso 2. $s = \widetilde{\alpha v}.s'$ y $\widetilde{\alpha v}.B \xrightarrow{\widetilde{\alpha}} B \stackrel{s'}{\Longrightarrow} B'$.

Como $B \approx_{k+1} C$ se sabe que $\exists C'$ tal que $C \stackrel{s'}{\Longrightarrow} C'$ con $C' \approx_k B'$.

Finalmente $\widetilde{\alpha v}.C \stackrel{s}{\Longrightarrow} C' \approx_k B'$.

ii. $\tau.B \approx_k \tau.C$

Se demuestra por inducción sobre k que $B \approx_k C$ implica $\tau.B \approx_k \tau.C$.

Para $k = 0$ es inmediato. Suponga $B \approx_{k+1} C$ y $\tau.B \stackrel{s}{\Longrightarrow} B'$, se tiene que $B \stackrel{s}{\Longrightarrow} B'$ y por lo tanto $\exists C'$ tal que $C \stackrel{s}{\Longrightarrow} C'$ con $B' \approx_k C'$.

Finalmente $\tau.C \stackrel{s}{\Longrightarrow} C' \approx_k B'$.

iii. $B \mid D \approx_k C \mid D$

Se demuestra por inducción sobre k que $\forall B, C, D$, $B \approx_k C$ implica $B \mid D \approx_k C \mid D$. Para $k = 0$ es inmediato. Suponga $B \approx_{k+1} C$ y $B \mid D \stackrel{s}{\Longrightarrow} E$, entonces $E = B' \mid D'$ con $B \stackrel{q}{\Longrightarrow} B'$, $D \stackrel{r}{\Longrightarrow} D'$ (en donde s es el resultado de un entrelazamiento de los eventos en q y r posiblemente considerando τ como resultado de operaciones sobre etiquetas complementarias).

Como $B \approx_{k+1} C$ entonces $\exists\, C'$ tal que $C \overset{q}{\Longrightarrow} C'$ y $B' \approx_k C'$.

Se sigue que $C \mid D \overset{s}{\Longrightarrow} C' \mid D' \approx_k B' \mid D' = E$ aplicando la hipótesis inductiva. En el otro sentido la demostración es simétrica.

iv. $B \setminus \alpha \approx_k C \setminus \alpha$

Se prueba por inducción sobre k que $B \approx_k C$ implica $B \setminus \alpha \approx_k C \setminus \alpha$.

Para k = 0 es inmediato. Suponga $B \approx_{k+1} C$ y $B \setminus \alpha \overset{s}{\Longrightarrow} B' = B'' \setminus \alpha$,

se sigue que $B \overset{s}{\Longrightarrow} B''$ y por lo tanto $\exists\, C''$ tal que $C \overset{s}{\Longrightarrow} C''$ y $C'' \approx_k B''$.

Finalmente $C \setminus \alpha \overset{s}{\Longrightarrow} C'' \setminus \alpha \approx_k B'' \setminus \alpha = B'$ por la hipótesis inductiva.

v. $B[S] \approx_k C[S]$

Se prueba por inducción sobre k que $B \approx_k C$ implica $B[S] \approx_k C[S]$.

Para k = 0 es inmediato. Suponga $B \approx_{k+1} C$ y $B[S] \overset{s}{\Longrightarrow} B' = B''[S]$,

se sigue que $B \overset{S^{-1}(s)}{\Longrightarrow} B''$, $(s = \mu_1 v_1 .. \mu_n v_n$, $S^{-1}(s) = S^{-1}(\mu_1) v_1 ... S^{-1}(\mu_n) v_n)$

y por lo tanto $\exists\, C''$ tal que $C \overset{S^{-1}(s)}{\Longrightarrow} C''$ y $B'' \approx_k C''$.

Finalmente $C[S] \overset{s}{\Longrightarrow} C''[S] \approx_k B''[S] = B'$ por la hipótesis inductiva.

vi. $B\{\tilde{v} / \tilde{x}\} \approx_k C\{\tilde{v} / \tilde{x}\}$ $\forall \tilde{v}$ implica $\alpha\tilde{x}.B \approx_k \alpha\tilde{x}.C$

Se prueba por inducción sobre k que $B\{\tilde{v} / \tilde{x}\} \approx_k C\{\tilde{v} / \tilde{x}\}$ $\forall \tilde{v}$ implica $\alpha\tilde{x}.B \approx_k \alpha\tilde{x}.C$. Para k = 0 es inmediato. Suponga

$B\{\tilde{v} / \tilde{x}\} \approx_{k+1} C\{\tilde{v} / \tilde{x}\}$ $\forall \tilde{v}$ y $\alpha\tilde{x}.B \overset{s}{\Longrightarrow} B'$. Se presentan dos casos :

Caso 1. $s = \varepsilon$ y $\alpha\tilde{x}.B \stackrel{s}{\Longrightarrow} \alpha\tilde{x}.B$.

En este caso se obtiene $\alpha\tilde{x}.C \stackrel{s}{\Longrightarrow} \alpha\tilde{x}.C \approx_k \alpha\tilde{x}$.

Caso 2. $s = \alpha\tilde{v}.s'$ y $\alpha\tilde{x}.B \xrightarrow{\alpha\tilde{v}} B\{\tilde{v} / \tilde{x}\} \stackrel{s'}{\Longrightarrow} B' = B'_{\tilde{v}}$.

Como $B\{\tilde{v} / \tilde{x}\} \approx_{k+1} C\{\tilde{v} / \tilde{x}\}$ se sabe que $\exists\ C'_{\tilde{v}}$ tal que

$C\{\tilde{v} / \tilde{x}\} \stackrel{s'}{\Longrightarrow} C'_{\tilde{v}}$ con $C'_{\tilde{v}} \approx_k B'_{\tilde{v}}$. Finalmente

$\alpha\tilde{x}.C \xrightarrow{\alpha\tilde{v}} C\{\tilde{v} / \tilde{x}\} \stackrel{s'}{\Longrightarrow} C'_{\tilde{v}} \approx_k B'_{\tilde{v}}$.

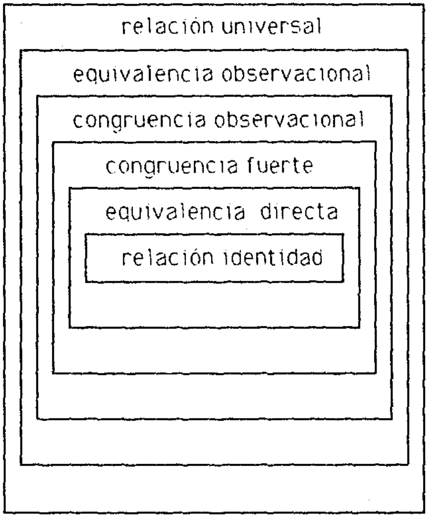
Nota.

De los resultados anteriores se deduce que en el proceso de demostración de equivalencia observacional , se puede hacer uso de todas las reglas probadas incluyendo la cancelación de τ. No se pueden realizar inferencias utilizando la congruencia aditiva .

7.1. Congruencia observacional.

Con el objetivo de obtener una congruencia observacional y conseguir la libertad completa de poder realizar substituciones de expresiones equivalentes en cualquier contexto, se procede a introducir algunos nuevos conceptos.

Recapitulando lo revisado hasta este punto, se han presentado varias equivalencias cuyas dependencias se muestran en la siguiente figura :



Definición . (Congruencia observacional)

Sean B,C términos de CCS, $B \overset{c}{\approx} C$ sii en cualquier expresión contextual $\Im[\ ]$, $\Im[B] \approx \Im[C]$.

Nota.

Esta definición está dada en términos de la equivalencia observacional .

Teorema.

i. $\overset{c}{\approx}$ es una congruencia.

ii. Si θ es una congruencia, $B\theta C$ implica $B \approx C$ entonces

$$B\theta C \text{ implica } B \overset{c}{\approx} C.$$

Demostración.

i. Es un resultado inmediato ya que cualquier contexto involucrando alguna operación sobre B y / o C también constituye un contexto involucrando a B y / o C.

ii. θ una congruencia implica que $\forall B, C$, $\Im[B] \overset{c}{\approx} \Im[C]$ $(\forall \Im[\ ])$ y por lo tanto $\Im[B] \approx \Im[C]$ de donde $B \overset{c}{\approx} C$.

Nota.

El resultado anterior establece que la relación $\overset{c}{\approx}$ es la congruencia más débil entre las congruencias que son más fuertes que $\approx$.

Nota.

$\overset{c}{\approx}$ es una congruencia,

$$B \overset{c}{\approx} C \text{ implica } \begin{cases} \overline{\alpha v}.B \overset{c}{\approx} \overline{\alpha v}.C & \tau.B \overset{c}{\approx} \tau.C \\ B + D \overset{c}{\approx} C + D & D + B \overset{c}{\approx} D + C \\ B \mid D \overset{c}{\approx} C \mid D & D \mid B \overset{c}{\approx} D \mid C \\ B \setminus \alpha \overset{c}{\approx} C \setminus \alpha & B[S] \overset{c}{\approx} C[S] \end{cases}$$

$$B\{\tilde{v} / \tilde{x}\} \overset{c}{\approx} C\{\tilde{v} / \tilde{x}\} \quad \forall \tilde{v} \quad \text{implica} \quad \alpha\tilde{x}.B \overset{c}{\approx} \alpha\tilde{x}.C$$

Corolario.

$\forall$ B,C términos de CCS, $B \sim C$ implica $B \overset{c}{\approx} C$ implica $B \approx C$.

Demostración.

$B \sim C$ implica $B \overset{c}{\approx} C$ se obtiene porque $\sim$ es una congruencia y $B \sim C$ implica $B \approx C$.

$B \overset{c}{\approx} C$ implica $B \approx C$ se sigue considerando el contexto particular $\Im[B] = B$.

Existe otra forma de introducir a la congruencia observacional, considerando que la adición es la única operación de la cual adolece la equivalencia $\approx$ para poder ser una congruencia.

Definición.

Sean B,C términos en CCS, $B \overset{+}{\approx} C$ sii $\forall D$, $B + D \approx C + D$.

Nota.

Esta definición ajusta precisamente el punto del cual adolece la equivalencia observacional, a saber, la congruencia aditiva.

Algunos resultados adicionales que son de interés son los siguientes.

Lema.

Si $B \overset{+}{\approx} C$ y $B \xrightarrow{\tau} B'$ entonces para cada $k > 0$ existe C' tal que

$C \overset{\tau}{\Longrightarrow} C'$ y $B' \approx_k C'$.

Demostración.

Suponga que C' no existe para alguna k.

Sea $D = \lambda_0.NIL$ con $\lambda_0 \notin$ género(B) $\cup$ género(C), como $B \xrightarrow{\tau} B'$

se sigue que $B + D \overset{\varepsilon}{\Longrightarrow} B'$. Sea $C + D \overset{\varepsilon}{\Longrightarrow} E$, se presentan tres posibilidades :

i. E es $C + D \not\approx B'$ ya que $C + D \xrightarrow{\lambda_0} NIL$ pero B' no tiene algún evento λ_0.

ii. $C \overset{\tau}{\Longrightarrow} E \not\approx B'_k$ por la suposición hecha,

iii. $D \overset{\tau}{\Longrightarrow} E$ es imposible,

por lo tanto $B + D \not\approx C + D$ lo cual contradice el hecho de que $B \overset{+}{\approx} C$.

Teorema.

La relación $\overset{+}{\approx}$ es una congruencia, es decir $B \overset{+}{\approx} C$ implica

i. $g.B \overset{+}{\approx} g.C \qquad (g \neq \tau)$

ii. $\tau.B \overset{+}{\approx} \tau.C$

iii. $B + D \overset{+}{\approx} C + D$

iv. $B \setminus \alpha \overset{+}{\approx} C \setminus \alpha$

v. $B[S] \overset{+}{\approx} C[S]$

vi. $B \mid D \overset{+}{\approx} C \mid D$.

Demostración.

i. $B \overset{+}{\approx} C$ implica $g.B \overset{+}{\approx} g.C \qquad (g \neq \tau)$

Se prueba por inducción sobre k que $g.B + D \approx_k g.C + D \quad \forall D$.

Suponga $g.B + D \overset{s}{\Longrightarrow} D'$, se presentan dos casos:

Caso 1. $s = \varepsilon$.

Si $D' = g.B + D$ entonces $g.C + D \overset{\varepsilon}{\Longrightarrow} g.C + D \approx_k D'$ por hipótesis, en otro caso $D \overset{\tau}{\Longrightarrow} D'$ y por lo tanto $g.C + D \overset{\tau}{\Longrightarrow} D'$.

Caso 2. $s \neq \varepsilon$.

Si $g.B \overset{s}{\Longrightarrow} D'$ entonces $\exists\, E'$ tal que $g.C \overset{s}{\Longrightarrow} E' \approx_k D'$ ya que

$g.B \approx g.C$, en otro caso $D \overset{s}{\Longrightarrow} D'$ y por lo tanto $g.C + D \overset{s}{\Longrightarrow} D'$.

ii. $B \overset{+}{\approx} C$ implica $\tau.B \overset{+}{\approx} \tau.C$

Se prueba por inducción sobre k que $\tau.B + D \approx_k \tau.C + D \quad \forall D$.

Suponga $\tau.B + D \overset{s}{\Longrightarrow} D'$, se presentan dos casos :

Caso 1. $s = \varepsilon$.

Si $D' = \tau.B + D$ entonces $\tau.C + D \overset{s}{\Longrightarrow} \tau.C + D \approx_k D'$ por hipótesis,

en otro caso $D \overset{\tau}{\Longrightarrow} D'$ y por lo tanto $\tau.C + D \overset{\tau}{\Longrightarrow} D'$, en el último caso

$\tau.B \overset{\tau}{\Longrightarrow} B \overset{\varepsilon}{\Longrightarrow} D'$ y por lo tanto $\exists\, E'$ tal que $C \overset{\varepsilon}{\Longrightarrow} E' \approx_k D'$ de donde

$\tau.C \overset{\tau}{\Longrightarrow} C \overset{\varepsilon}{\Longrightarrow} E' \approx_k D'$.

Caso 2. $s \neq \varepsilon$.

Si $D \overset{s}{\Longrightarrow} D'$ entonces $g.C + D \overset{s}{\Longrightarrow} D'$, en otro caso $B \overset{s}{\Longrightarrow} D'$ y por lo

tanto $\exists\, E'$ tal que $C \overset{s}{\Longrightarrow} E' \approx_k D'$ de donde $\tau.C + D \overset{s}{\Longrightarrow} E' \approx_k D'$.

iii. $B \overset{+}{\approx} C$ implica $B + D \overset{+}{\approx} C + D$

Se prueba que $(B + D) + E \approx (C + D) + E \quad \forall E$.

$(B + D) + E \approx B + (D + E) \approx C + (D + E)$ (ya que $B \overset{+}{\approx} C$)

$\approx (C + D) + E$.

iv. $B \stackrel{+}{\approx} C$ implica $B \setminus \alpha \stackrel{+}{\approx} C \setminus \alpha$

Se prueba por inducción que $B \setminus \alpha + D \approx_k C \setminus \alpha + D \quad \forall D$.

Suponga $B \setminus \alpha + D \stackrel{s}{\Longrightarrow} D'$, se tienen dos casos :

Caso 1. $s = \varepsilon$.

Si $D' = B \setminus \alpha + D$ entonces $C \setminus \alpha + D \stackrel{\varepsilon}{\Longrightarrow} C \setminus \alpha + D \approx_k D'$ por hipótesis,

en otro caso $D \stackrel{\tau}{\Longrightarrow} D'$ implica $C \setminus \alpha + D \stackrel{\tau}{\Longrightarrow} D'$, en el último caso

$B \setminus \alpha \stackrel{\tau}{\Longrightarrow} D' = D'' \setminus \alpha$ de donde $B \stackrel{\tau}{\Longrightarrow} D''$ y por lo tanto $C \stackrel{\tau}{\Longrightarrow} E' \approx_k D''$

(lema), finalmente $C \setminus \alpha + D \stackrel{s}{\Longrightarrow} E' \setminus \alpha = D'' \setminus \alpha = D'$.

Caso 2. $s \neq \varepsilon$.

Si $B \setminus \alpha \stackrel{s}{\Longrightarrow} D'$ entonces $\exists$ E' tal que $C \setminus \alpha \stackrel{s}{\Longrightarrow} E' \approx_k D'$ ya que

$B \setminus \alpha \approx C \setminus \alpha$, en otro caso $D \stackrel{s}{\Longrightarrow} D'$ y por lo tanto $C \setminus \alpha + D \stackrel{s}{\Longrightarrow} D'$.

v. $B \stackrel{+}{\approx} C$ implica $B[S] \stackrel{+}{\approx} C[S]$

Se prueba por inducción sobre k que $B[S] + D \approx_k C[S] + D \quad \forall D$.

Suponga $B[S] + D \stackrel{s}{\Longrightarrow} D'$, se obtienen dos casos :

Caso 1. $s = \varepsilon$

Si $D' = B[S] + D$ entonces $C[S] + D \stackrel{s}{\Longrightarrow} C[S] + D \approx_k D'$ por hipótesis, en otro

caso $D \stackrel{\tau}{\Longrightarrow} D'$ y por lo tanto $C[S] + D \stackrel{s}{\Longrightarrow} D'$, en el último caso

$B[S] \stackrel{\tau}{\Longrightarrow} D'$ implica $B \stackrel{\tau}{\Longrightarrow} D'[S^{-1}]$ y por lo tanto $C \stackrel{\tau}{\Longrightarrow} E'[S^{-1}] \approx_k D'[S^{-1}]$

(lema) de donde $C[S] + D \stackrel{\tau}{\Longrightarrow} E' \approx_k D'$.

Caso 2. $s \neq \varepsilon$.

Si $B[S] \overset{s}{\Longrightarrow} D'$ entonces $\exists\, E'$ tal que $C[S] \overset{s}{\Longrightarrow} E' \approx_k D'$ ya que $B[S] \approx C[S]$,

en otro caso $D \overset{s}{\Longrightarrow} D'$ y por lo tanto $C[S]+D \overset{s}{\Longrightarrow} D'$.

vi. $B \overset{+}{\approx} C$ implica $B|D \overset{+}{\approx} C|D \quad \forall D$.

Se prueba por inducción sobre k que $B|D+E \approx_k C|D+E \quad \forall E$.

Suponga $B|D+E \overset{s}{\Longrightarrow} E'$, se presentan varios casos :

Caso 1. $s \neq \varepsilon$.

Si $E \overset{s}{\Longrightarrow} E'$ entonces $C|D+E \overset{s}{\Longrightarrow} E'$, en otro caso $B|D \overset{s}{\Longrightarrow} E'$ y por lo tanto

$\exists\, F$ tal que $C|D \overset{s}{\Longrightarrow} F$ ya que $B|D \approx C|D$, finalmente $C|D+D \overset{s}{\Longrightarrow} F \approx_k E'$.

Caso 2. $s = \varepsilon$.

Si $E' = B|D+E$ entonces $C|D+E \overset{\varepsilon}{\Longrightarrow} C|D+E \approx_k B|D+E$ por la

hipótesis, en otro caso $E \overset{\tau}{\Longrightarrow} E'$ y por lo tanto $C|D+E \overset{\tau}{\Longrightarrow} E'$, en el caso de que

$B|D \xrightarrow{\tau} B'|D' \overset{\varepsilon}{\Longrightarrow} E'$ se presentan 3 subcasos :

a) $B'=B$ y $D \xrightarrow{\tau} D'$.

Se tiene que $C|D \xrightarrow{\tau} C|D'$ y $B|D' \approx C|D'$, por lo tanto $\exists\, F'$ tal que

$C|D' \overset{\varepsilon}{\Longrightarrow} F' \approx_k E'$ de donde $C|D+E \overset{\varepsilon}{\Longrightarrow} F' \approx_k E'$.

b) $D'=D$ y $B \xrightarrow{\tau} B'$.

Por el lema se obtiene que $\exists\, C'$ tal que $C \overset{\tau}{\Longrightarrow} C' \approx_{k+1} B'$ de donde

$B'|D \approx_{k+1} C'|D$, ya que $B'|D \overset{\varepsilon}{\Longrightarrow} E'$ entonces $\exists\, F'$ tal que $C'|D \overset{\varepsilon}{\Longrightarrow} F' \approx_k E'$,

finalmente $C|D+E \overset{\tau}{\Longrightarrow} C'|D \overset{\varepsilon}{\Longrightarrow} F' \approx_k E'$.

c) $B \xrightarrow{\lambda v} B'$ y $D \xrightarrow{\bar{\lambda} v} D'$.

Se sigue que $\exists C'$ tal que $C \overset{\lambda v}{\Longrightarrow} C' \approx_{k+1} B'$ de donde

$C|D \overset{\tau}{\Longrightarrow} C'|D' \approx_{k+1} B'|D'$, por lo tanto $C'|D' \overset{r}{\Longrightarrow} F' \approx_k E'$ para algún

F' y finalmente $C|D + E \overset{r}{\Longrightarrow} F' \approx_k E'$.

A partir de este resultado se obtiene el siguiente teorema .

Teorema .

Las relaciones $\overset{c}{\approx}$ y $\overset{+}{\approx}$ corresponden a la misma congruencia .

Demostración .

$B \overset{c}{\approx} C$ implica $B \overset{+}{\approx} C$ ya que $\mathfrak{C}[.] = . + D$ es un contexto particular .

$B \overset{+}{\approx} C$ implica $B \overset{c}{\approx} C$ ya que $\overset{+}{\approx}$ es una congruencia y $B \overset{+}{\approx} C$ implica $B \approx C$.

Se sabe que la equivalencia $\approx$ no preserva la substitución aditiva , sin embargo, afortunadamente existe una clase importante de términos B y C para los cuales se puede inferir que $B \overset{c}{\approx} C$ a partir de $B \approx C$, y por lo tanto $B + D \overset{c}{\approx} C + D$ $\forall D$.

Definición .

Sea B un término de CCS, B es estable si y sólo si $B \xrightarrow{\tau} B'$ es imposible $\forall B'$.

Un comportamiento estable es aquel en el cual el sistema no puede realizar cambios a menos que sean registrados por el observador externo .

Las siguientes dos proposiciones son relativas a la propiedad de estabilidad de un sistema .

Proposición.

Si $B \overset{c}{\approx} C$ entonces B y C son ambos estables o ambos inestables.

Demostración.

Es un resultado directo del lema ya probado.

Proposición.

Si B y C son estables y $B \approx C$ entonces $B \overset{c}{\approx} C$.

Demostración.

Se prueba por inducción que $B + D \approx_k C + D \quad \forall D$.

Suponga que $B + D \overset{s}{\Longrightarrow} E$, se obtienen dos casos:

Caso 1. $s = \varepsilon$.

Si $E = B + D$ entonces $C + D \overset{\varepsilon}{\Longrightarrow} C + D \approx_k B + D$ por la hipótesis inductiva, en otro caso $D \overset{\tau}{\Longrightarrow} E$ de donde $C + D \overset{\varepsilon}{\Longrightarrow} E$. ($B \overset{\tau}{\Longrightarrow} E$ es imposible por hipótesis).

Caso 2. $s \neq \varepsilon$.

Si $D \overset{s}{\Longrightarrow} E$, entonces $C + D \overset{s}{\Longrightarrow} E$, en otro caso $B \overset{s}{\Longrightarrow} E$, de donde $C \overset{s}{\Longrightarrow} F \approx_k E$ ya que $B \approx C$, finalmente $C + D \overset{s}{\Longrightarrow} F \approx_k E$.

Proposición.

Para todo guardia g, $B \approx C$ implica $g.B \overset{c}{\approx} g.C$.

Demostración.

Caso 1. $g \neq \tau$.

Se obtiene que $g.B \approx g.C$ y por lo tanto $g.B \overset{c}{\approx} g.C$ ya que ambos son estables.

Caso 2. $g = \tau$.

Se prueba por inducción que $\tau.B + D \approx_k \tau.C + D \quad \forall D$.

Suponga $\tau.B + D \stackrel{s}{\Longrightarrow} E$, se presentan dos casos :

a) $s = \varepsilon$.

Si $E = \tau.B + D$ entonces $\tau.C + D \stackrel{\varepsilon}{\Longrightarrow} \tau.C + D \approx_k E$ por hipótesis, en otro caso $D \stackrel{\tau}{\Longrightarrow} E$ y por lo tanto $\tau.C + D \stackrel{\varepsilon}{\Longrightarrow} E$, en el último caso $\tau.B \stackrel{\tau}{\Longrightarrow} B \stackrel{\varepsilon}{\Longrightarrow} E$ de donde $C \stackrel{\varepsilon}{\Longrightarrow} F \approx_k E$ para algún F ya que $B \approx C$, finalmente $\tau.C + D \stackrel{\varepsilon}{\Longrightarrow} F \approx_k E$.

b) $s \neq \varepsilon$.

Si $D \stackrel{s}{\Longrightarrow} E$ entonces $C + D \stackrel{s}{\Longrightarrow} E$, en otro caso $B \stackrel{s}{\Longrightarrow} E$ de donde $C \stackrel{s}{\Longrightarrow} F \approx_k E$ ya que $B \approx C$, finalmente $\tau.C + D \stackrel{s}{\Longrightarrow} F \approx_k E$.

Las siguientes equivalencias de la congruencia observacional involucran a la acción no observable τ. Estas equivalencias pueden justificarse intuitivamente pensando a cada comportamiento B como una colección de capacidades de acción (las ramas del árbol que lo identifica) que incluyen acciones τ las cuales son capaces de rechazar a las otras acciones .

Teorema. (leyes τ)

Las siguientes equivalencias son ciertas :

i. $g.\tau.B \stackrel{c}{\approx} g.B$

ii. $B + \tau.B \stackrel{c}{\approx} \tau.B$

iii. $g.(B + \tau.C) + g.C \stackrel{c}{\approx} g.(B + \tau.C)$

Demostración .

i. Como $B \approx \tau.B$ entonces $g.B \overset{c}{\approx} g.\tau.B$ por la última proposición .

ii. Se prueba por inducción sobre k que $B + \tau.B + D \approx_k \tau.B + D \quad \forall D$.

Suponga $B + \tau.B + D \overset{s}{\Longrightarrow} E$, se presentan dos casos :

Caso 1. $s = \varepsilon$.

Si $E = B + \tau.B + D$ entonces $\tau.B + D \overset{\varepsilon}{\Longrightarrow} \tau.B + D \approx_k E$ por la hipótesis inductiva , en otro caso $D \overset{\tau}{\Longrightarrow} E$ implica $\tau.B + D \overset{\tau}{\Longrightarrow} E$, en otro caso si $\tau.B \overset{\tau}{\Longrightarrow} E$ entonces $\tau.B \overset{\varepsilon}{\Longrightarrow} E$, finalmente si $B \overset{\tau}{\Longrightarrow} E$ entonces $\tau.B \xrightarrow{\tau} B \xrightarrow{\tau} E$ de donde $\tau.B + D \overset{\varepsilon}{\Longrightarrow} E$.

Caso 2. $s \neq \varepsilon$.

Si $D \overset{s}{\Longrightarrow} E$ entonces $\tau.B + D \overset{s}{\Longrightarrow} E$, en otro caso si $B \overset{s}{\Longrightarrow} E$ o $\tau.B \overset{s}{\Longrightarrow} E$ se sigue $\tau.B \overset{s}{\Longrightarrow} E$, los casos inversos son similares.

iii. Se prueba por inducción sobre k que $g.(B + \tau.C) + g.C + D \approx_k g.(B + \tau.C) + D \quad \forall D$.

Suponga $g.(B + \tau.C) + g.C + D \overset{s}{\Longrightarrow} E$, se presentan dos casos :

Caso 1. $s = \varepsilon$.

Si $E = g.(B + \tau.C) + g.C + D$ entonces $g.(B + \tau.C) + D \overset{\varepsilon}{\Longrightarrow} g.(B + \tau.C) + D \approx_k E$ por hipótesis, en otro caso $D \overset{\tau}{\Longrightarrow} E$ implica $g.(B + \tau.C) + D \overset{\tau}{\Longrightarrow} E$, en otro caso si $g.(B + \tau.C) \overset{\tau}{\Longrightarrow} E$ entonces $g.(B + \tau.C) \overset{\tau}{\Longrightarrow} E$, en el último caso $g.C \overset{\tau}{\Longrightarrow} E$ se sigue $g.(B + \tau.C) \xrightarrow{\tau} B + \tau.C \xrightarrow{\tau} E$ de donde $g.(B + \tau.C) \overset{\varepsilon}{\Longrightarrow} E$.

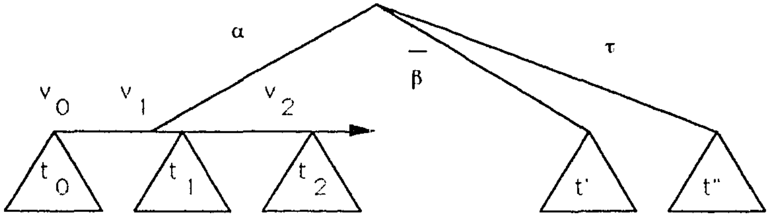
Caso 2. $s \neq \tau$.

Si $D \overset{s}{\Longrightarrow} E$ o $g.(B + \tau.C) \overset{s}{\Longrightarrow} E$ entonces $g.(B + \tau.C) + D \overset{s}{\Longrightarrow} E$, en otro caso $g.C \overset{s}{\Longrightarrow} E$ y se sigue que $g.(B + \tau.C) + D \xrightarrow{g} B + \tau.C \xrightarrow{\tau} \tau.C \overset{s'}{\Longrightarrow} E$ con $s = gs'$. En forma similar se realizan los casos inversos .

8. Un modelo de CCS : Arboles de comunicación.

Se ha dicho que un árbol de sincronización de género $L \subsetneq \Lambda$ es un árbol no ordenado finitamente ramificado cuyos arcos están etiquetados por elementos de $L \cup \{\tau\}$. Las etiquetas positivas se utilizan para asociar valores a variables (entrada) y las etiquetas negativas para calificar expresiones de valor (salida).

Si $\{v_0, v_1, ...\}$ son valores del tipo apropiado a α y v es un valor del tipo asociado a $\bar{\beta}$, un árbol de comunicación es representado como se muestra a continuación :



Una vez realizado el evento α, el valor v_i recibido selecciona el correspondiente árbol t_i como el comportamiento posterior. En el caso de realizarse el evento $\bar{\beta}$, el valor v es transmitido.

Este árbol de comunicación es una interpretación de un programa con comportamiento

$$\alpha x.B + \bar{\beta} v.B' + \tau.B''$$

en donde el programa $B\{v_i / x\}$ corresponde a t_i, el programa B' corresponde a t' y el programa B" corresponde a t".

Definición .

Un CT (árbol de comunicación) de género L es una colección finita de parejas de la forma

$< \alpha, f >$ ($\alpha \in L$) en donde f es una familia de CT 's de género L indizada por el conjunto de valores del tipo apropiado a α,

$< \bar{\beta}, < v,t >>$ $\bar{\beta} \in L$, v un valor del tipo asociado a $\bar{\beta}$, t un CT de género L,

$< \tau, t >$ t un CT de género L.

Notación.

CT_L denota el conjunto de CT 's de género L, v_α denota el conjunto de valores asociados a α.

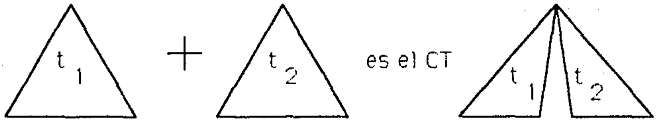
Las operaciones dinámicas operan sobre los árboles de comunicación de la siguiente forma :

i) NIL (operación cero - aria)

NIL es el CT .

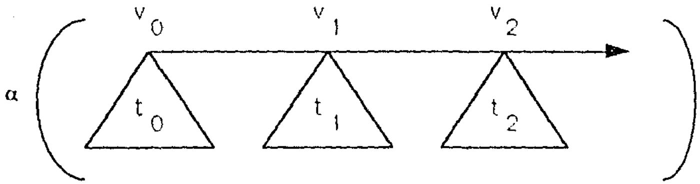
NIL $\in CT_\Phi$.

ii) Adición + (operación binaria)

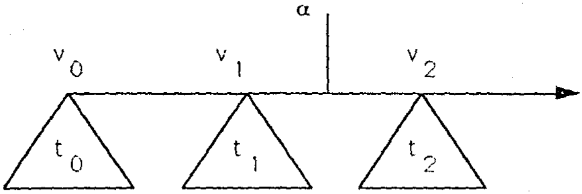


$+ \in CT_L \times CT_M \rightarrow CT_{L \cup M}$.

iii) Acción prefijo α (operación v_α - aria)



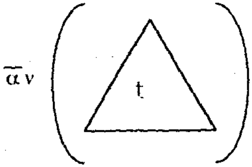
es el CT



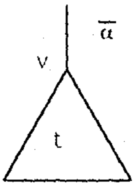
Si $f : v_\alpha \rightarrow CT_L$ es una familia de CT 's indizada por v_α entonces $\alpha \in (v_\alpha \rightarrow CT_L) \rightarrow CT_{L\cup\{\alpha\}}$.

Acción prefijo $\bar{\alpha}$ (operación unaria)

Para cada $v \in v_\alpha$,



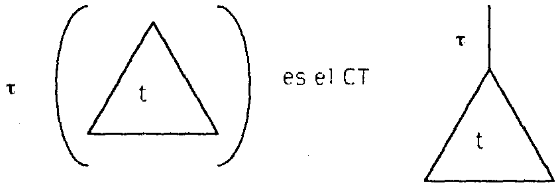
es el CT



Para cada $v \in v_{\alpha}$, $\bar{\alpha}v \in CT_L \to CT_{L\cup\{\bar{\alpha}\}}$ o

$\alpha \in v_{\alpha} \to (CT_L \to CT_{L\cup\{\bar{\alpha}\}})$.

Acción prefijo τ (operación unaria)



$\tau \in CT_L \to CT_L$.

Existe una correspondencia natural entre programas de CCS y árboles de comunicación ,
por ejemplo

NIL	NIL
$\bar{\alpha}v.B$	$\bar{\alpha}vt$
$\tau.B$	τt
$B + B'$	$t + t'$
$\alpha x.B$	αf

Notación.

Si $v \mapsto t(v)$ denota una familia de CT 's se utiliza $\alpha(v \mapsto t(v))$ para denotar αf.

Notación.

$[[B]] \in CT$ denota la interpretación de un programa B de CCS .

Definición .

$$||NIL|| = NIL$$
$$||\alpha x.B|| = a(v \mapsto ||B\{v / x\}||)$$
$$||\bar{\alpha}v.B|| = \bar{\alpha}v||B||$$
$$||\tau.B|| = \tau||B||$$
$$||B + B'|| = ||B|| + ||B'||.$$

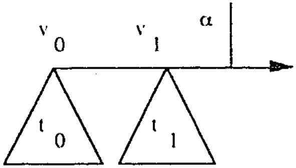
A continuación se presentan las operaciones estáticas $|$, $\backslash \alpha$ y $[S]$ definidas sobre árboles de comunicación .

iv) Composición $|$ (operación binaria)

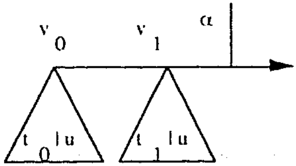
$$| \in CT_L \times CT_M \rightarrow CT_{L \cup M}$$

Sean $t \in CT_L$, $u \in CT_M$, entonces $t|u$ tiene las siguientes ramas

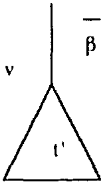
a) Para cada rama



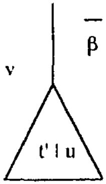
de t, una rama



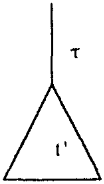
b) Para cada rama



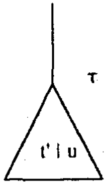
de t, una rama



c) Para cada rama

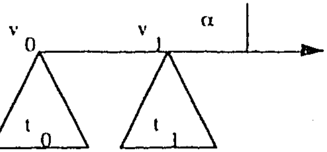


de t, una rama

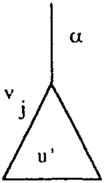


Para las ramas de u se opera en forma similar.

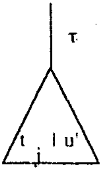
d) Para cada par de ramas



de t y



de u, una rama



y similarmente para ramas $<\bar{\alpha},<v_j,t'>>$ de t y $<\alpha,v_i \mapsto u_i>$ de u.

v) Restricción $\backslash\alpha$ (operación unaria)

$$\backslash\alpha \in CT_L \rightarrow CT_{L-\{\alpha,\bar{\alpha}\}}$$

El resultado se obtiene podando todas las ramas etiquetadas con $\bar{\alpha}$ o α.

vi) Re - etiquetamiento $[S]$ (operación unaria)

$$[S] \in CT_L \rightarrow CT_M \qquad S : L \rightarrow M .$$

El resultado se obtiene reemplazando cada $\lambda \in L$ por $S\lambda \in M$ en todo el árbol .
Para continuar con la interpretación de CCS en términos de árboles de sincronización se ofrece la siguiente definición .

Defínción .

$$[[B|B']] = [[B]] \mid [[B']]$$
$$[[B\backslash\alpha]] = [[B]]\backslash\alpha$$
$$[[\ B[S]\ ]] = [[B]]\,[S]$$
$$[[\text{if true then } B \text{ else } B']] = [[B]]$$
$$[[\text{if false then } B \text{ else } B']] = [[B']].$$

Nota.

Excepto por la regla de absorción ($B + B = B$), se puede interpretar CCS y la congruencia fuerte $\sim$ mediante los CT 's que los representan.

Definición .

Si $b(x_1, \ldots, x_{n(b)}) \Leftarrow B_b$ entonces $[|b(\tilde{v})|] = [|B_b\{\tilde{v} / \tilde{x}\}|]$.

La familia de relaciones $\xrightarrow{\mu v}$ de CCS se pueden interpretar en este modelo de CT 's.

La siguiente definición de las relaciones $\xrightarrow{\mu v}$ sobre CT's es independiente de la definición correspondiente en CCS aunque se ha utilizado la misma notación .

Definición .

Sea t un CT definido mediante una colección de pares de la forma $<\alpha, f>$, $<\bar{\beta}, <v,t>>$, $<\tau, t>$. Entonces t tiene las siguientes acciones atómicas :

i) $t \xrightarrow{\alpha v} f(v)$ para cada $<\alpha, f>$ y para cada $v \in v_\alpha$ de t,

ii) $t \xrightarrow{\bar{\beta} v} t'$ para cada $<\beta', <v, t'>>$ de t,

iii) $t \xrightarrow{\tau} t'$ para cada $<\tau, t'>$ de t.

Nota.

Las acciones atómicas de un comportamiento B y su interpretación $[|B|]$ son exactamente las mismas . Si $B \xrightarrow{\mu v} B'$ entonces $[|B|] \xrightarrow{\mu v} [|B'|]$ y si $[|B|] \xrightarrow{\mu v} t'$ entonces existe B' tal que $B \xrightarrow{\mu v} B'$ y $[|B'|] = t'$.

9. Prueba del despachador.

El ejemplo del despachador presentado en 3.4 está sujeto a las siguientes restricciones:

i) la eliminación de todas las ocurrencias β_i $(1 \leq i \leq n)$ produce la secuencia $(\alpha_1\alpha_2...\alpha_n)^{(\omega)}$,
ii) para cada i, la eliminación de todas las ocurrencias α_j,β_j $(j \neq i)$ produce la secuencia

$(\alpha_i\beta_i)^{(\omega)}$.

Las anteriores restricciones pueden ser descritas en CCS de la siguiente forma:

i) $Sch \parallel (\beta_1^{(\omega)} \mid ... \mid \beta_n^{(\omega)}) \approx (\bar{\alpha}_1 ... \bar{\alpha}_n)^{(\omega)}$

ii) $Sch \parallel (\prod_{j \neq i} \alpha_j^{(\omega)} \mid \prod_{j \neq i} \beta_j^{(\omega)}) \approx (\bar{\alpha}_i\bar{\beta}_i)^{(\omega)}$

Notación.

$\prod_{i \in I} q_i$ denota la composición múltiple de $\{q_i : i \in I\}$.

Nota.

El despachador está definido por $Sch = (s \mid c_1 \mid c_2 \mid ... \mid c_n) \setminus \gamma_1 .. \setminus \gamma_n$

Prueba de i).

Sea $Sch' = Sch \parallel (\beta_1^{(\omega)} \mid ... \mid \beta_n^{(\omega)})$, se probará que $Sch' \approx (\bar{\alpha}_1 ... \bar{\alpha}_n)\, Sch'$.
Se puede reescribir $Sch' = (s \mid c_1' \mid c_2' \mid ... \mid c_n') \setminus \gamma_1 .. \setminus \gamma_n$ con

$c_i' = (c_i \mid \beta_i^{(\omega)}) \setminus \beta_i$ utilizando propiedades de los operadores $\mid$ y $\setminus \alpha$.

Se tiene que

$$c_i' \overset{c}{\approx} \gamma_i \bar{\alpha}_i \bar{\gamma}_{i+1} c_i' \quad \text{ya que}$$

$$\begin{aligned} c_i' &\approx (\gamma_i \bar{\alpha}_i (\bar{\beta}_i\, \bar{\gamma}_{i+1} c_i + \bar{\gamma}_{i+1} \bar{\beta}_i c_i) \mid \beta_i^{(\omega)}) \setminus \beta_i \\ &\approx \gamma_i \bar{\alpha}_i (\tau\, \bar{\gamma}_{i+1} c_i' + \bar{\gamma}_{i+1} \tau c_i') \\ &\approx \gamma_i \bar{\alpha}_i (\tau\, \bar{\gamma}_{i+1} c_i' + \bar{\gamma}_{i+1} c_i') \\ &\approx \gamma_i \bar{\alpha}_i\, \bar{\gamma}_{i+1} c_i' \end{aligned}$$

utilizando la regla $B + \tau.B \overset{c}{\approx} \tau.B.$

Haciendo uso de lo anterior se prosigue como sigue :

$$
\begin{aligned}
Sch' &\overset{c}{\approx} (\bar{\gamma}_1 NIL \mid \gamma_1 \bar{\alpha}_1 \bar{\gamma}_2 c_1' \mid \dots \mid \gamma_n \bar{\alpha}_n \bar{\gamma}_1 c_n') \setminus \gamma_1 \dots \setminus \gamma_n \\
&\overset{c}{\approx} \tau(NIL \mid \bar{\alpha}_1 \bar{\gamma}_2 c_1' \mid \gamma_2 \bar{\alpha}_2 \bar{\gamma}_3 c_2' \mid \dots \mid \gamma_n \bar{\alpha}_n \bar{\gamma}_1 c_n') \setminus \gamma_1 \dots \setminus \gamma_n \\
&\overset{c}{\approx} \tau \bar{\alpha}_1 \tau \bar{\alpha}_2 \dots \tau \bar{\alpha}_n (NIL \mid c_1' \mid c_2' \mid \dots \mid \bar{\gamma}_1 c_n') \setminus \gamma_1 \dots \setminus \gamma_n \\
&\overset{c}{\approx} \tau \bar{\alpha}_1 \tau \bar{\alpha}_2 \dots \tau \bar{\alpha}_n \tau(NIL \mid \bar{\alpha}_1 \bar{\gamma}_2 c_1' \mid c_2' \mid \dots \mid c_n') \setminus \gamma_1 \dots \setminus \gamma_n \\
&\approx \bar{\alpha}_1 \bar{\alpha}_2 \dots \bar{\alpha}_n \, Sch'.
\end{aligned}
$$

Prueba de ii).

Sea $Sch' = Sch \parallel (\prod_{j \neq i} \alpha_j^{\omega} \mid \prod_{j \neq i} \beta_j^{\omega})$, se probará que $Sch' \approx \bar{\alpha}_i \bar{\beta}_i \, Sch'$.

Reescribiendo ,

$$Sch' = (s \mid c_1' \mid c_2' \mid \dots \mid c_{i-1}' \mid c_i \mid c_{i+1}' \mid \dots \mid c_n') \setminus \gamma_1 \dots \setminus \gamma_n$$

con $c_j' = (c_j \mid \alpha_j^{\omega} \mid \beta_j^{\omega}) \setminus \beta_j \setminus \alpha_j \;\; j \neq i,$

y $c_i = \gamma_i \bar{\alpha}_i (\bar{\beta}_i \bar{\gamma}_{i+1} c_i + \bar{\gamma}_{i+1} \bar{\beta}_i \; c_i).$

Se tiene que $c_j' \overset{c}{\approx} \gamma_j \bar{\gamma}_{j+1} c_j'$ ya que

$$
\begin{aligned}
c_j' &\approx (c_j \mid \alpha_j^{\omega} \mid \beta_j^{\omega}) \setminus \alpha_j \setminus \beta_j \\
&\approx (\gamma_j \bar{\alpha}_j (\bar{\beta}_j \bar{\gamma}_{j+1} c_j + \bar{\gamma}_{j+1} \bar{\beta}_j \; c_j) \mid \alpha_j^{\omega} \mid \beta_j^{\omega}) \setminus \alpha_j \setminus \beta_j \\
&\approx \gamma_j \tau (\tau \bar{\gamma}_{j+1} c_j' + \bar{\gamma}_{j+1} \tau c_j') \\
&\approx \gamma_j (\bar{\gamma}_{j+1} c_j' + \bar{\gamma}_{j+1} c_j') \\
&\approx \gamma_j \; \bar{\gamma}_{j+1} c_j'.
\end{aligned}
$$

De lo anterior , se procede como sigue :

$$\begin{aligned}
Sch' &\overset{c}{\approx} (\bar{\gamma}_1\ NIL \mid \gamma_1\bar{\gamma}_2 c_1' \mid \ldots \mid \gamma_{i-1}\bar{\gamma}_i c_{i-1}' \mid \gamma_i\,\bar{\alpha}_i(\bar{\beta}_i\,\bar{\gamma}_{i+1}c_i + \bar{\gamma}_{i+1}\bar{\beta}_i c_i) \mid \\
&\qquad \gamma_{i+1}\bar{\gamma}_{i+2}c_{i+1}' \mid \ldots \mid \gamma_n\bar{\gamma}_1 c_n')\setminus\gamma_1\ldots\setminus\gamma_n. \\
&\overset{c}{\approx} \tau(NIL \mid \bar{\gamma}_2 c_1' \mid \ldots \mid \gamma_{i-1}\bar{\gamma}_i c_{i-1}' \mid \gamma_i\bar{\alpha}_i(\bar{\beta}_i\bar{\gamma}_{i+1}c_i + \bar{\gamma}_{i+1}\bar{\beta}_i c_i) \mid \\
&\qquad \gamma_{i+1}\bar{\gamma}_{i+2}c_{i+1}' \mid \ldots \mid \gamma_n\bar{\gamma}_1 c_n')\setminus\gamma_1\ldots\setminus\gamma_n. \\
&\overset{c}{\approx} \tau^i\bar{\alpha}_i(NIL \mid c_1' \mid \ldots \mid c_{i-1}' \mid \bar{\beta}_i\,\bar{\gamma}_{i+1}c_i + \bar{\gamma}_{i+1}\bar{\beta}_i c_i \mid \\
&\qquad \gamma_{i+1}\bar{\gamma}_{i+2}c_{i+1}' \mid \ldots \mid \gamma_n\bar{\gamma}_1 c_n')\setminus\gamma_1\ldots\setminus\gamma_n. \\
&\overset{c}{\approx} \tau^i\bar{\alpha}_i(\bar{\beta}_i(NIL \mid c_1' \mid \ldots \mid c_{i-1}' \mid \bar{\gamma}_{i+1}c_i \mid \gamma_{i+1}\bar{\gamma}_{i+2}c_{i+1}' \mid \ldots \mid \gamma_n\bar{\gamma}_1 c_n') \\
&\qquad + \bar{\gamma}_{i+1}(NIL \mid c_1' \mid \ldots \mid c_{i-1}' \mid \bar{\beta}_i c_i \mid \gamma_{i+1}\bar{\gamma}_{i+2}c_{i+1}' \mid \ldots \mid \gamma_n\bar{\gamma}_1 c_n'))\setminus\gamma_1\ldots\setminus\gamma_n. \\
&\overset{c}{\approx} \tau^i\bar{\alpha}_i(\bar{\beta}_i\ \tau^{n-i}(NIL \mid c_1' \mid \ldots \mid c_{i-1}' \mid c_i \mid c_{i+1}' \mid \ldots \mid \bar{\gamma}_1 c_n') \\
&\qquad + \bar{\gamma}_{i+1}\bar{\beta}_i\gamma_{i+1}\tau^{n-i+1}(NIL \mid c_1' \mid \ldots \mid c_{i-1}' \mid c_i \mid c_{i+1}' \mid \ldots \mid \bar{\gamma}_1 c_n')\setminus\gamma_1\ldots\setminus\gamma_n. \\
&\overset{c}{\approx} \tau^i\bar{\alpha}_i(\bar{\beta}_i\ \tau^{n-i+1}\ Sch' + \bar{\beta}_i\ \tau^{n-i+2}\ Sch') \\
&\overset{c}{\approx} \bar{\alpha}_i(\bar{\beta}_i\ Sch' + \bar{\beta}_i\ Sch') \\
&\approx \bar{\alpha}_i\bar{\beta}_i\ Sch'.
\end{aligned}$$

Conclusiones.

El cálculo de sistemas comunicantes constituye un caso de estudio especial dentro de un área más amplia conocida como teoría algebraica de procesos.

Mediante el estudio y aplicación del CCS se pueden enfrentar los problemas de diseño análisis y razonamiento sobre procesos concurrentes y paralelos en una forma conveniente.

La presentación realizada sobre el cálculo de sistemas comunicantes CCS ha mostrado métodos para definir nuevos procesos a partir de otros ya existentes mediante la aplicación de operadores específicos y también se han ilustrado las facilidades que ofrece para establecer definiciones recursivas, todo esto a través de la presentación y solución de problemas concretos sobre el paralelismo y la concurrencia de procesos.

En forma general , cualquier sistema consistente de un conjunto de procesos y de una colección de puertos mediante los cuáles se establece la comunicación entre los procesos es susceptible de ser analizado, descrito y razonado a través de CCS.

Si se considera la totalidad de descripciones sintácticas obtenidas mediante la aplicación sobre procesos de los operadores que conforman a CCS (álgebra de términos) como un modelo denotacional del mismo, es posible apreciar que algunos términos están relacionados en el sentido de que describen al mismo sistema y que difieren por el grado de detalle de la descripción, algunos siendo más descriptivos que otros, pero esencialmente denotando al mismo sistema. De hecho, esta identificación de procesos exhibe a un término como una implementación de otro.

En cualquier teoría algebraica para el tratamiento de procesos, la idea más importante que debe de justificarse es el razonamiento que se sigue para decidir si dos términos deben identificarse como descripciones distintas de un mismo sistema. La decisión anterior está fuertemente apoyada en la noción del no determinismo el cual es un concepto básico para explicar las posibles interacciones entre procesos como lo son la sincronización y la comunicación de los mismos.

El cálculo de sistemas comunicantes está equipado con varias nociones de equivalencia basadas en principios extraídos del estudio de esquemas de transición etiquetados de los cuáles se obtiene información sobre el comportamiento de los procesos, de la forma en que pueden ser utilizados y de la manera en la cual interactuan como parte de sistemas más complejos, con lo cual es posible proporcionar criterios de distinción de procesos para poder separar a los términos diferentes utilizando la observación como mecanismo fundamental.

En particular, las pruebas a las cuáles son sometidos los procesos para poder distinguirlos tienen que ver con la habilidad o capacidad para que éstos reaccionen o participen en un evento bajo determinadas condiciones.

En forma conjunta se ha presentado la semántica de CCS mediante otro modelo formal denotacional, a saber , los árboles de comunicación, en donde dos descripciones son consideradas equivalentes si la función de significado que los mapea a un árbol les asigna la misma imagen. Una de las propiedades principales de la función de significado en el modelo CT es que preserva la estructura de los términos, es decir , la semántica de un término complejo se puede obtener conociendo la semántica de los términos más sencillos que lo constituyen.

En general puede observarse que la utilización de modelos denotacionales facilita el desarrollo de sistemas deductivos de pruebas para demostrar la equivalencia de descripciones, en nuestro caso se han obtenido ecuaciones que son ciertas tanto en el modelo semántico CT como en distintas equivalencias de comportamiento.

El papel que representa CCS como mecanismo descriptivo para analizar sistemas conformados mediante procesos comunicantes es central en el área algebraica ya que ha servido como motivación para el desarrollo de distintos lenguajes orientados al tratamiento algebraico de procesos, los cuales incluyen variadas equivalencias de comportamiento y cálculos basados en sistemas ecuacionales que permiten establecer razonamientos precisos sobre diversos sistemas paralelos .

La influencia de CCS en investigaciones recientes se manifiesta en el lenguaje LOTOS (Language Of Temporal Ordering Specification) desarrollado bajo el programa ESPRIT de la Comunidad Europea con el objetivo de crear formalismos descriptivos para el diseño y la especificación de sistemas distribuidos abiertos. El lenguaje LOTOS se ha utilizado para la especificación de la jerarquía de protocolos OSI en donde existe una necesidad evidente de separar conceptos como lo son servicio y protocolo en una forma clara, concisa y no ambigua.

Bibliografía.

Ahuja V.,1982

Design and Analysis of Computer Communications Networks

Mc-Graw Hill.

Andrews G.R.,1989

Concepts and Notations for Concurrent Programming

Computing Surveys,15(1),3-43.

Ben-Ari M.,1982

Principles of Concurrent Programming

Prentice Hall International.

Bertsekas D.,Gallager R.,1987

Data Networks

Prentice Hall Inc.

Brinch Hansen P.,1978

Distributed Processes:A Concurrent Programming Concept

CACM 21(11),934-941.

Brinch Hansen P.,1978

Specification and Implementation of Mutual Exclusion

IEEE Transactions on Software Engineering Vol. SE-4,No.5,365-370.

Gehani N.,McGettrick A.,1988

Concurrent Programming

Addison Wesley.

Habermann A.,1972

Synchronization of Communicating Processes

CACM 15(3),171-176.

Halsall F.,1988

Data Communications,Computer Networks and OSI, 2nd. ed.

Addison Wesley.

Hennessy M.,1988

Algebraic Theory of Processes

MIT Press.

Hoare C.A.R.,1985
Communicating Sequential Processes
Prentice Hall International.

Hoare C.A.R.,1974
Monitors:An Operating System Structuring Concept
CACM 17(10),549-557.

Krishnamurthy E.,1989
Parallel Processing,Principles and Practice
Addison Wesley.

Lamport L.,1974
A New Solution of Dijkstra's Concurrent Programming Problem
CACM 17(8),453-455.

Milner R.,1980
A Calculus of Communicating Systems
Lecture Notes in Computer Science,92
Springer Verlag.

Milner R.,1989
Communication and Concurrency
Prentice Hall.

Oktaba H.,1982
Programación Concurrente
Comunicaciones Técnicas, Serie Azul,86
IIMAS,UNAM.

Owicki S.,Gries D.,1976
Verifying Properties of Parallel Programs: An Axiomatic Approach
CACM 19(5),279-285.

Raynal M.,1982
Algorithmes Distribués et Protocoles
Eyrolles,París.

Raynal M.,1987
Networks and Distributed Computation
Concepts,Tools and Algorithms
North Oxford Academic.

Tanenbaum A.,1988

Computer Networks,2nd. ed.

Prentice Hall Inc.

Van Eijk P.,Vissers C.,Díaz M.,(Editors),1989

The Formal Description Technique Lotos

Results of the Esprit/Sedos Project

North Holland.