

# UN LENGUAJE FORMAL DE ESPECIFICACION DE PROCESOS COMUNICANTES PARA EL ANALISIS DE REDES.

Ph. JORRAND\*, J. M. PEREIRA\*\*

## RESUMEN

Un lenguaje formal para la especificación de Procesos Comunicantes es presentado. Dos modelos operacionales equivalentes le son asociados: un sistema de transición de estados y un sistema de reescritura de términos con eventos.

Sobre este lenguaje se introduce un cálculo para la construcción de redes de procesos. Este cálculo es una manipulación sintáctica de símbolos del lenguaje formal que nos permite reducir el análisis de una red al análisis de un proceso simple.

## ABSTRACT

A formal language for the specification of Communicating Processes is presented. Two equivalent operational models are associated with it: a state transition system and a term rewriting system with events.

On top of this language we introduce a calculus for the construction of networks of processes. This calculus is syntactic manipulation of symbolic of the formal language which allows us to reduce the analysis of a network to the analysis of a single process.

\* Director de Investigación en el CNRS. Teoría de la Programación, Informática Fundamental, Especificación, Procesos Comunicantes, IMAG-BP 68-38402  
St. Martin d'Hères cedex - FRANCE.

\*\* Ingeniero en Computación. Magister en Ciencias de la Computación. Docteur "Informatique"; Teoría de la Programación-Teoría de Tipos-Semántica de Lenguajes-Sistemas de Procesos; Universidad Simón Bolívar-Dpto. de Matemáticas-AP 80659-CARACAS-VENEZUELA.

\*\* Este trabajo ha sido realizado bajo el patrocinio financiero del CONICIT (VENEZUELA), el Servicio Cultural de la Embajada de Francia en Venezuela y el Ministerio de Relaciones Exteriores de Francia.

Un "*proceso*" puede ser considerado como una caja (máquina, agente) capaz de hacer cálculos y de enviar y recibir mensajes a través ciertos puntos de conexión ligados a la caja. Estos puntos de conexión son llamados "*puertas*" de entrada y/o salida del proceso. Ellos constituyen los únicos medios por los cuales el proceso se puede comunicar con su medio ambiente.

Esta capacidad de transmitir y recibir información hacia o por el medio ambiente permite construir *Sistemas de Procesos* que se comunican (SPC). En estos sistemas los procesos intercambian mensajes entre ellos a través de sus puertas. Así una "*comunicación*" es considerada como la transmisión instantánea de un mensaje desde una puerta de entrada hacia una puerta de salida.

Los Sistemas de Procesos que deseamos modelar son tales que:

- los procesos que conforman el sistema evolucionan de forma independiente, salvo en los momentos en los cuales ellos se comunican. Esto introduce todas las posibilidades del *paralelismo*.
- para que una comunicación tenga lugar ella debe ser sincronizada.  
Esto quiere decir que la comunicación se realiza solamente en el caso en el proceso que transmite y el proceso que recibe estén listos para intercambiar el mensaje. Esta hipótesis sobre las comunicaciones ha sido introducida por C.A.R. Hoare y se la conoce bajo el nombre de "*rendez-vous*".
- varias comunicaciones pueden ser realizadas simultáneamente. El modelo describe entonces secuencias de conjuntos de comunicaciones simultáneos.  
Estos conjuntos serán denominados "*eventos*".
- la construcción de un sistema de procesos será un caso particular de la especificación de un proceso simple. Es decir un SPC es una frase de la notación sintáctica utilizada para describir un proceso simple.
- podamos estudiar el comportamiento del proceso independiente de su medio ambiente. Para ello, un proceso es considerado como una "*unidad aislada*".  
Para asirla, nosotros asociamos, a cada puerta, una "*puerta ficticia*" llamada  $\delta$  que representa cualquier puerta de cualquier medio ambiente.  
Debido a este aislamiento no se puede hacer ninguna suposición sobre el orden en el cual un medio ambiente se va a comunicar con el proceso. Luego la descripción de un proceso debe considerar todas las secuencias posibles de eventos que él puede realizar i.e. todos los medios ambientes en los cuales él puede ser utilizado.

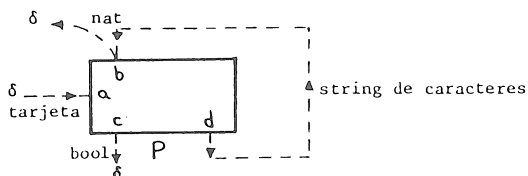
En la sección 2 se introduce el lenguaje formal para describir procesos y dos modelos semánticos para dicho lenguaje formal. Ambos modelos son operacionales y equivalentes.

En la sección 3 se introduce un cálculo formal que nos permite construir los SPC. Este calculo es una manipulación sintáctica de frases del lenguaje formal definido en la sección 2 que permite reducir un SPC a un proceso simple.

Finalmente en la sección 4 se discuten las posibilidades de analizar el comportamiento de un SPC bajo el modelo propuesto y la potencia de este lenguaje.

## 2.SINTAXIS Y SEMANTICA FORMAL DE UN PROCESO.

La estructura de un proceso puede ser considerada como un diagrama del tipo:



Así, un proceso posee:

- un nombre
  - un conjunto Puertas formado por los identificadores de sus puertas
  - un conjunto de conexiones entre puertas definidas como un subconjunto de
- $$K = K_{IN} \cup K_{EX}$$

|                          |                                    |                       |
|--------------------------|------------------------------------|-----------------------|
| donde $K_{IN} \subseteq$ | Puertas x Puertas                  | conexiones internas   |
| $K_{EX} \subseteq$       | $K_{en} \cup K_{sa}$               | conexiones externas   |
| $K_{en} \subseteq$       | $\{\delta\} \times \text{Puertas}$ | conexiones de entrada |
| $K_{sa} \subseteq$       | $\text{Puertas} \times \{\delta\}$ | conexiones de salida  |

Los elementos  $\langle x, y \rangle \in K$  son denotados por  $x.y$  y ellos representan la conexión entre la puerta de salida  $x$  y la puerta de entrada  $y$ .

Cada conexión en el proceso tiene asociado un identificador de dominio que denota el tipo de mensaje que puede circular por la conexión.

El comportamiento de un proceso expresa las diferentes secuencias de eventos permitidas y las diferentes funciones calculadas. Al nivel del diagrama, esto expresa cuales son las diferentes secuencias en que las conexiones se disparan y cual es el mensaje que ese momento se está transmitiendo.

### 2.1 LA SINTAXIS FORMAL DE UN PROCESO.

Un lenguaje  $L$  para describir procesos es un par  $L = \langle D, L \rangle$  donde  $D$  es un conjunto de identificadores de dominios

$L$  consiste en tres firmas:

- un conjunto  $F$  de símbolos de operación con una función de aridad

$\rho_F : F \rightarrow D^* \times D$  donde  $D^*$  es el conjunto de secuencias finitas de elementos de  $D$ , incluyendo la secuencia vacía  $\lambda$ .

-un conjunto  $K$  de símbolos de conexión con una función de aridad  $\rho_K : K \rightarrow D$ .

-un conjunto  $E$  de símbolos de relación con una función de aridad  $\rho_E : E \rightarrow D^*$

Sea  $V(L) = \{V_d \mid d \in D\}$  una familia de conjuntos no-vacios indexada por  $D$  denominada variables de  $L$ .

Sea  $D' = D \cup \{\text{estado}, \text{evento}\}$  t.q. estado, evento  $\notin D$

Los términos de  $L$ ,  $T(L) = \{T_d \mid d' \in D'\}$  es la mas pequeña familia de conjuntos indexada por  $D'$  t.q.

i.  $V_d \subseteq T_d$  para todo  $d \in D$

ii. si  $f \in F$  y  $\rho_F(f) = \langle \lambda, d \rangle$  entonces  $f \in T_d$  para todo  $d \in D$

iii. si  $f \in F$  y  $\rho_F(f) = \langle d_1, \dots, d_n, d \rangle$  y  $t_i \in T_{d_i}$  para  $1 \leq i \leq n$  entonces  $f(t_1, \dots, t_n) \in T_d$

iv. si  $\{k_1, \dots, k_n\} \subseteq K$  y  $\{k_1, \dots, k_n\} \neq \emptyset$  y  $\rho_K(k_i) = d_i$  y  $t_i \in T_{d_i}$  para  $1 \leq i \leq n$  entonces  $\{k_1(t_1), \dots, k_n(t_n)\} \in T_{\text{evento}}$

v.  $\tau \in T_{\text{evento}}$  ( $\tau$  denota el evento vacío i.e.  $\emptyset$ )

vi. si  $e \in E$  y  $\rho_E(e) = d_1 \dots d_n$  y  $t_i \in T_{d_i}$  para  $1 \leq i \leq n$  entonces  $e(t_1, \dots, t_n) \in T_{\text{estado}}$ .

Los conjuntos  $T_d$  engendrados por las reglas i)ii)iii) son denominados términos de operación y denotados por  $T_{\text{oper}}$ .

Para cada  $t \in T(L)$  la familia de variables que aparecen en  $t$  es denotada por  $\text{var}(t)$ .

En el caso que  $t \in T_{\text{evento}}$ ,  $\text{var}(t)$  esta dividida en dos

$$\text{var}_{\text{en}}(t) = \bigcup_{t_i \in E_n(t)} \text{var}(t_i) \quad \text{donde } E_n(t) = \{t' \mid k(t') \in t \wedge k \in K_{\text{en}}\}$$

$$\text{var}_{\text{sa}}(t) = \bigcup_{t_i \in S_a(t)} \text{var}(t_i) \quad \text{donde } S_a(t) = \{t' \mid k(t') \in t \wedge k \in K_{\text{so}}\} \cup K_{\text{IN}}$$

Las reglas de  $L$  es el conjunto definido por

$$R(L) = \{t_1 \xrightarrow{t_2} t_3 \mid t_1, t_3 \in T_{\text{estado}} \wedge t_2 \in T_{\text{evento}} \\ \wedge \text{var}_{\text{sa}}(t_2), \text{var}(t_3) \subseteq \text{var}(t_1) \cup \text{var}_{\text{en}}(t_2)\}$$

Luego un proceso es descrito por una tripleta

$$p = \langle L, I, R \rangle$$

donde

$L$  es un lenguaje para describir procesos

$I \subseteq T_{\text{estado en } L} \text{ t.q. } I \neq \emptyset \wedge \forall t \in I . \text{var}(t) = \emptyset \wedge I \text{ finito}$

$R \subseteq R(L) \text{ t.q. } R \text{ es finito}$

Por comodidad en los ejemplos se utilizan la siguiente notación para la descripción de un proceso

$p = \text{proceso}$

operaciones  $f_{1,1}, \dots, f_{1,n} : d_{1,1} \dots d_{1,i} \rightarrow d_1$   
 $\vdots$   
 conexiones  $k_{1,1}, \dots, k_{1,r} : d_1$   
 $\vdots$   
 estados  $e_{1,1}, \dots, e_{1,p} : d_{1,1} \dots d_{1,j}$   
 $\vdots$   
 inidales  $t_1, \dots, t_k$   
 reglas  $t_{1,1} \xrightarrow{t_{2,1}} t_{3,1}$   
 $\vdots$

Ejemplo:

Sea  $\text{Nat}$  el identificador del dominio de los naturales

$\text{Sec}(\text{Nat})$  el identificador de dominios de las secuencias de los naturales

El proceso Pila recibe naturales por su puerta de entrada  $i$  y los envia por su puerta de salida  $o$  en orden LIFO (último en entrar primero en salir)

$\text{Pila} = \text{proceso}$

operaciones  $\text{cons} : \text{Sec}(\text{Nat}), \text{Nat} \rightarrow \text{Sec}(\text{Nat.})$

$\wedge : \rightarrow \text{Sec}(\text{Nat})$

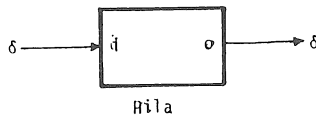
conexiones  $\delta \cdot 1, o \cdot \delta : \text{Nat}$

estados  $P : \text{Sec}(\text{Nat})$

inicial  $P(\wedge)$

reglas  $P(w) \xrightarrow{\{\delta \cdot i(n)\}} P(\text{cons}(w, n))$

$P(\text{cons}(w, n)) \xrightarrow{\{o \cdot \delta(n)\}} P(w)$



Se proponen en esta sección dos modelos semánticos para las descripciones de procesos y se muestra su equivalencia.

El primer modelo interpreta la descripción de un proceso  $p = \langle L, I, R \rangle$  como un sistema de transición de estados, i.e. un automata, en general no-determinístico e infinito.

El segundo modelo propuesto también operacional interpreta la descripción  $p = \langle L, I, R \rangle$  como un sistema de transición de términos con Eventos, i.e. una especie de sistema de Reglas de Reescritura con Eventos incorporadas a las reglas.

Otros modelos semánticos han sido asociados a la descripción de procesos [Per 84]: la presentación de una variedad de álgebras y un árbol en el espacio ultramétrico de los árboles infinitos. Estos modelos permiten estudiar ciertas propiedades del comportamiento de un SPC que serán objeto de futuras publicaciones.

### 2.2.1 LA DESCRIPCION DE UN PROCESO: UN SISTEMA DE TRANSICION DE ESTADOS.

Un sistema de transición de estados está definido como una tupla  $S = (Q, T, Rel)$  donde:

- $Q$  es un conjunto (no necesariamente finito) de estados
- $T$  es un conjunto (no necesariamente finito) de transiciones, en nuestro caso de eventos.
- $Rel$  es un conjunto de  $Q \times T \times Q$

Nos proponemos construir un sistema de transición de estados para cada descripción de procesos.

Sean  $p = \langle L, I, R \rangle$  (donde  $L = \langle D, L \rangle$  con  $L$  conteniendo  $F, K, E$ )

$H_F(p) = \{c / \text{var}(c) = \emptyset \wedge c \in T_d \in T(L) \text{ para cada } d \in D\}$  el universo de

Herbrand de las operaciones en p

La  saturación  en  $H_F(p)$   de una regla   $t_1 \xrightarrow{t_2} t_3 \in R$  es el conjunto

$$\text{Sat}_{H_F(p)}(t_1 \xrightarrow{t_2} t_3) = \{e_1 \xrightarrow{e_2} e_3 / e_1 = \emptyset t_1 \wedge e_2 = \emptyset t_2 \wedge e_3 = \emptyset t_3 \\ \wedge \emptyset : \text{var}(t_1) \cup \text{var}(t_2) \rightarrow H_F(p)\}$$

Sean  $\text{Universo}(R) = \bigcup_{\zeta \in R} \text{Sat}_{H_F(p)}(\zeta)$

Estados  $(R) = \{e_1, e_3 / e_1 \xrightarrow{e_2} e_3 \in \text{Universo}(R)\} \cup I$

Eventos  $(R) = \{e_2 / e_1 \xrightarrow{e_2} e_3 \in \text{Universo}(R)\}$

La evolución de la descripción  $p$  es una relación

$$\rightsquigarrow_n \subseteq \text{Estados}(R) \times \text{Eventos}(R) \times \text{Estados}(R)$$

definida por (se denota a  $\rightsquigarrow_n^b c$  para  $\langle a, b, c \rangle \in \rightsquigarrow_n$ ):

$$i. \quad e_1 \rightsquigarrow_n^{e_2} e_3 \text{ ssi } e_1 \xrightarrow{e_2} e_3 \in \text{Universo}(R) \wedge e_1 \in I$$

$$\text{ii. } e_1 \xrightarrow[n]{e_2} e_3 \text{ ssi } e_1 \xrightarrow{e_2} e_3 \in \text{Universo}(R) \wedge \exists e_1' \xrightarrow[n-1]{e_2'} e_1 \quad 532$$

Entonces, el sistema de transición de estados de una descripción de procesos  $p$  es el siguiente

$$S(R) = (\text{Estados}(R), \text{Eventos}(R), \rightsquigarrow)$$

donde  $\rightsquigarrow = \bigcup_{n \in \mathbb{N}} \rightsquigarrow_n$

## 2.2.2 LA DESCRIPCION DE UN PROCESO: UN SISTEMA DE TRANSICION DE TERMINOS.

El estudio de ciertas propiedades sobre los procesos es mas facilmente realizable si se posee una interpretación mas "finita" de su comportamiento.

Así por ejemplo las Redes de Petri o los Programas Paralelos [Kel76] son formalismos operacionales para tener una representación finita del comportamiento infinito. Por ello nosotros proponemos un segundo modelo operacional que interpreta a  $p = \langle L, I, R \rangle$  como un Sistema de Transición de Términos con Eventos. Se pudiera decir que el primer modelo es el conceptual mientras que el segundo es su interpretación finita.

Sea  $H_F(p)$  el universo de Herbrand de las operaciones en  $p = \langle L, I, R \rangle$

$$\forall t \in T(L) \text{ se define } \text{Sat}_{H_F(p)}(t) = \{t' / \exists \theta = t' \wedge \theta: \text{var}(t) \rightarrow H_F(p)\}$$

En el caso que  $t \in T_{\text{estado}}$  entonces  $\text{Sat}_{H_F(p)}(t)$  contiene todos los estados de notados por el término  $t$ .

$$\text{Estados}(p) = \bigcup_{t \in T_{\text{estados}}} \text{Sat}_{H_F(p)}(t)$$

Una regla  $t_1 \xrightarrow{t_2} t_3 \in R$  se dice aplicable a partir  $e \in \text{Estados}(p)$  si existe  $\theta: \text{var}(t_1) \cup \text{var}(t_2) \rightarrow H_F(p)$  t.q.  $\theta t_1 = e$ .

Una vez que esta regla es aplicada el evento que se produce es  $\theta t_2$  y el estado resultante es  $\theta t_3$ .

$$\text{Finalmente, si denotamos por } \text{Eventos}(p) = \bigcup_{t \in T_{\text{evento}}} \text{Sat}_{H_F(p)}(t)$$

La relación calculada por la descripción de un proceso  $p = \langle L, I, R \rangle$  es

$$\vdash = \bigcup_{n \in \mathbb{N}} \vdash_n \subseteq \text{Estados}(p) \times \text{Eventos}(p) \times \text{Estados}(p)$$

donde  $e_1 \vdash_0 e_3$  ssi existe  $t_1 \xrightarrow{t_2} t_3 \in R$  y  $e_1 \in I$  t.q. la regla es aplicada a partir del estado  $e_1$  i.e. existe  $\theta: \text{var}(t_1) \cup \text{var}(t_2) \rightarrow H_F(p)$

$$\text{t.q. } \theta t_1 = e_1, \theta t_2 = e_2, \theta t_3 = e_3$$

$e_1 \vdash_n e_3$  ssi existe  $t_1 \xrightarrow{t_2} t_3 \in R$  y  $e_1' \vdash_{n-1}^{e_2'} e_1$  t.q. la regla es aplicada a partir del estado  $e_1$

### 3. LA CONSTRUCCION DE UN SPC.

La especificación de sistemas de procesos que se comunican supone la posibilidad de "combinar" otros procesos especificados. Esto permite un grado de abstracción que facilita la "construcción" de sistemas muy complejos.

Para definir esta combinación, C.A.R. Hoare [Hoa 81a], R. Milner [Mil 81] W.C. Rounds-S.D. Brookes [Rou-Bro 81] utilizan las álgebras de procesos.

Una expresión en estas álgebras:

- tiene como operandos expresiones del álgebra.
- describe un sistema de procesos, i.e. el "sentido" de una expresión es en [Hoa 81a] un conjunto de historias, [Mil 81] un árbol de sincronización o de comunicación, [Rou-Bro 81] un conjunto de historia y un árbol

En el modelo que nosotros presentamos aquí, se define también un álgebra de expresiones. Esta álgebra está fundada sobre las ideas de Ph. Jorrand expuestas en [Jor 81, Jor 84]. En esta álgebra a diferencia de las anteriores, una expresión

- tiene también como operandos expresiones del álgebra
- da como resultado una descripción de procesos

Es decir esta álgebra es simplemente una manipulación simbólica de descripciones. Para conocer el "sentido" de una expresión es necesario interpretar la descripción resultante por una de las semánticas dadas para el lenguaje formal.

De esta forma todo sistema de procesos se reduce a un proceso simple.

#### 3.1. EL ALGEBRA DE DESCRIPCIONES: CALCULO DE JORRAND.

Puesto que el álgebra que se va a definir es una manipulación formal de descripciones, nosotros vamos a utilizar tres funciones  $\Psi, \Phi, \Pi$  que nos permitan hacer ciertas combinaciones de símbolos y definir el "producto de estados" y el "producto de términos de estados" necesarios para la descripción de los operadores del álgebra.

Sea  $\Psi$  una combinación arbitraria de secuencias finitas (por ejemplo la concatenación de secuencias).

Sea  $E_1, E_2$  dos conjuntos de símbolos de estados con funciones de aridad  $\rho_{E_1}, \rho_{E_2}$ . El producto de estados  $E_1 \times E_2$  es un conjunto de símbolos de estados  $\hat{E}$  con función de aridad  $\rho_{\hat{E}}$  t.q. existe una función  $\phi : E_1 \times E_2 \rightarrow \hat{E}$  donde

$$\text{si } \phi(e_1, e_2) = e \text{ entonces } \rho_{\hat{E}}(e) = \Psi(\rho_{E_1}(e_1), \rho_{E_2}(e_2))$$

El producto de términos de estados  $T_{\text{estado}_{E_1}} \times T_{\text{estado}_{E_2}}$  es un conjunto de términos de estados definidos por la función

$$\Pi_{E_1 \times E_2} : T_{\text{estado}_{E_1}} \times T_{\text{estado}_{E_2}} \rightarrow T_{\text{estado}_{E_1 \times E_2}}$$

t.q.

$$\Pi_{E_1 \times E_2} (e(t_1, \dots, t_n), e'(t'_1, \dots, t'_m)) = e_1(\psi(\langle t_1, \dots, t_n \rangle, \langle t'_1, \dots, t'_m \rangle))$$

$$\text{donde } \phi(e, e') = e_1$$

El algebra de descripciones puede ahora ser definida como la estructura algebraica

$$\langle A, \parallel, +, - \rangle$$

donde

- A es el conjunto de todas las descripciones de procesos  $p = \langle L, I, R \rangle$
- los operadores son definidos por:

El operador unión  $= \parallel : A \times A \rightarrow A$

Este operador tiene por objetivo poner en paralelo sus procesos operados denotados por las descripciones de procesos  $p_1$  y  $p_2$ . Así la descripción del proceso resultante de evaluar la expresión  $p_1 \parallel p_2$  puede comportarse en cada etapa de su evolución:

1. Sea, como el proceso  $p_1$  evolucionando solo i.e. dejando el proceso  $p_2$  en el mismo estado.
2. Sea, como el proceso  $p_2$  evolucionando solo i.e. dejando el proceso  $p_1$  en el mismo estado.
3. Sea, como los dos procesos  $p_1$  y  $p_2$  evolucionando simultaneamente

Formalmente

Sea  $p_1 = \langle L_{p_1}, I_{p_1}, R_{p_1} \rangle$  donde  $L_{p_1} = \langle D_{p_1}, L_{p_1} \rangle$  con  $L_{p_1}$  compuesto de  $F_{p_1}, K_{p_1}, E_{p_1}$

Sea  $p_2 = \langle L_{p_2}, I_{p_2}, R_{p_2} \rangle$  donde  $L_{p_2} = \langle D_{p_2}, L_{p_2} \rangle$  con  $L_{p_2}$  compuesto de  $F_{p_2}, K_{p_2}, E_{p_2}$ .

Los tres conjuntos de reglas siguientes describen los tres puntos mencionados anteriormente

$$\begin{aligned} \hat{R}_1 &= \{ \Pi(t_1, t'_1) \stackrel{t_2}{\neq} \Pi(t_2, t'_1) / t_1 \stackrel{t_2}{\neq} t_2 \in R_{p_1} \wedge \\ &\quad (t'_1 = S(x_1, \dots, x_n) \wedge \rho_{E_{p_2}}(s) = d_1 \dots d_n \wedge \\ &\quad x_1, \dots, x_n \notin \text{var}(t_1 \stackrel{t_2}{\neq} t_2)) \text{ para cada } s \in E_{p_2} \} \\ \hat{R}_2 &= \{ \Pi(t_1, t'_1) \stackrel{t_2}{\neq} \Pi(t_1, t'_1) / t_1 \stackrel{t_2}{\neq} t_2 \in R_{p_2} \wedge \\ &\quad (t_1 = S(x_1, \dots, x_n) \wedge \rho_{E_{p_1}}(s) = d_1 \dots d_n \wedge \\ &\quad x_1, \dots, x_n \notin \text{var}(t_1 \stackrel{t_2}{\neq} t_2)) \text{ para cada } s \in E_{p_1} \} \end{aligned}$$

$$\hat{R}_3 = \{\Pi(t_1, t_1') \xrightarrow{t_2 \cup t_2'} \Pi(t_3, t_3') / t_1 \xrightarrow{t_2} t_3 \in R_{p_1} \wedge t_1' \xrightarrow{t_2'} t_3' \in R_{p_2}\}$$

Luego la descripción definida por  $p_1 \parallel p_2$  es la siguiente:

$$p_1 \parallel p_2 = \langle \hat{L}, \hat{I}, \hat{R} \rangle \quad \text{donde} \quad \hat{L} = \langle D_{p_1} \cup D_{p_2}, \hat{L} \rangle \quad \text{con} \quad \hat{L} \quad \text{compuesto de}$$

$$F_{p_1} \cup F_{p_2}, K_{p_1} \cup K_{p_2}, E_1 \times E_2$$

$$\hat{I} = (\Pi(e_1, e_2) / e_1 \in I_{p_1} \wedge e_2 \in I_{p_2})$$

$$\hat{R} = \hat{R}_1 \cup \hat{R}_2 \cup \hat{R}_3$$

El operador de conexión  $= + : A \times K_{IN} \rightarrow A$

Sea  $p = \langle L, I, R \rangle$  donde  $L = \langle D, L \rangle$  con  $L$  compuesto de  $F, K, E$ .

El operador de conexión construye un nuevo proceso en el cual dos puertas (definidas sobre el mismo dominio), una de entrada  $\delta.i$  y otra de salida  $o.\delta$  son conectadas. El nuevo proceso permite así todas las comunicaciones del antiguo proceso, mas las comunicaciones sobre  $o.i$  en los estados en los cuales  $\delta.i$  y  $o.\delta$  son posibles si simultaneamente.

Formalmente

$$p + o.i = \langle \hat{L}, \hat{I}, \hat{R} \rangle$$

donde

$$\hat{L} = \langle D, \hat{L} \rangle \quad \text{con} \quad \hat{L} \quad \text{conteniendo} \quad F, \hat{K} = K \cup \{o.i\}, E$$

donde  $\rho_{\hat{K}}(k) = \rho_K(k)$  para  $k \in K$

$$\text{y} \quad \rho_{\hat{K}}(o.i) = \rho_K(\delta.i) = \rho_K(o.\delta)$$

$$\hat{I} = I$$

$$\hat{R} = R \cup \{ \theta t_1 \xrightarrow{\theta t_2'} \theta t_3 / t_1 \xrightarrow{t_2} t_3 \in R \wedge$$

$$\exists t \cup \in T_{\rho_{\hat{K}}(o.i)}, o.\delta(t) \in t_2 \wedge \delta.i(u) \in t_2$$

$$\wedge t_2' = t_2 - \{o.\delta(t), \delta.i(u)\} \cup \{i.o(\theta t)\}$$

$$\wedge \theta = \text{pgu}(t, u) \}$$

$\text{pgu}(t, u)$  denota el mas grande unificador de  $t$  y  $u$ , i.e. una substitución

$\theta : \text{var}(t) \cup \text{var}(u) \rightarrow T_{\text{ope}}$  t.q.  $\theta t = \theta u$  es llamado un unificador de  $t$  y  $u$ . Si se tienen dos unificadores  $\theta'$  y  $\theta''$ , se dice que  $\theta''$  es mas general que  $\theta'$ , si existe una substitución  $\theta$  t.q.  $\theta' = \theta \circ \theta''$ . El  $\text{pgu}(t, u)$  es calculado por el algoritmo de unificación [Rob 65, 71].

El operador de restricción  $= - : A \times K \rightarrow A$

Sea  $p = \langle L, I, R \rangle$  donde  $L = \langle D, L \rangle$  con  $L$  compuesto por  $F, K, E$ .

El objetivo de este operador de restricción es construir un nuevo proceso en el cual la conexión  $k$ , (argumento del operador) se vuelve invisible. Esto quiere decir, en

1.  $k \in K_{EX}$ , el nuevo proceso no permite ninguna comunicación por esa puerta.
2.  $k \in K_{IN}$  el nuevo proceso puede efectuar las comunicaciones a través de la conexión  $k$ , pero ellas no pueden ser observadas por el medio exterior.

Formalmente

$$p-k = \langle \hat{L}, \hat{I}, \hat{R} \rangle \text{ donde}$$

$$\hat{L} = \langle D, \hat{L} \rangle \text{ con } \hat{L} \text{ conteniendo } F, \hat{K} = K - \{k\}, E$$

$$\hat{I} = I$$

$$\hat{R} = (R - R') \cup R''$$

con

$$R' = \{t_1 \xrightarrow{t_2} t_3 / t_1 \xrightarrow{t_2} t_3 \in R \wedge \exists u \in T_{\rho_{k(k)}}. k(u) \in t_2\}$$

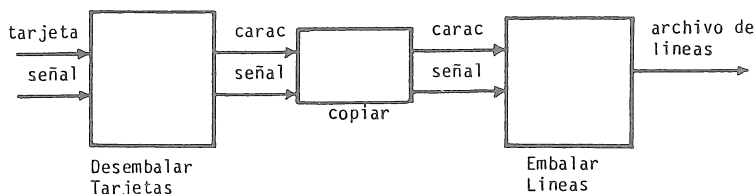
$$R'' = \{t_1 \xrightarrow{t_2 - \{k(u)\}} t_3 / t_1 \xrightarrow{t_2} t_3 \in R \wedge k \in K_{IN} \wedge k(u) \in t_2\}$$

#### 4. LA POTENCIA DEL LENGUAJE Y EL ANALISIS DE SPC.

El algebra de descripciones  $\langle A, \parallel, +, - \rangle$  tiene un interés multiple, así:

1. el algebra reduce el estudio de un SPC al estudio de un proceso simple.
2. el diseño de un SPC es simple y claro : se describen los procesos elementales y utilizando los operadores, se obtienen sistemas complejos, sin necesidad de dar su descripción.

Por ejemplo la expresión asociado a



describe el proceso de pasar archivos de tarjetas de 80 caracteres a archivos de líneas de 125 caracteres cada una. Cada tarjeta esta seguida de un espacio en blanco y la última línea esta completada por espacios blancos si es necesario.

3. en [Jor 84] se muestra que con las bases de esta álgebra, se pueden describir los operadores del calculo de R. Milner [Mil 80], C.A.R. Hoare - S. D. Brookes - A. W. Roscoe [Hoa-Bro-Ros 81], Austrey-Boudol [Aus-Bou 82] y W. C. Rounds - S. D. Brookes [Rou-Bro 81].
4. en [Per 84] se demuestra que con este lenguaje se pueden expresar todas las funciones calculables.

5. en [Jor 84a][Per 84] se interpreta la sintaxis formal de los procesos 537  
 $p = \langle L, I, R \rangle$  como la presentación de una variedad de álgebras.  
 Esta semántica permite analizar los procesos en el mismo sentido de los trabajos realizados en tipos abstractos de datos. Esta semántica algebraica da las bases necesarias para introducir este lenguaje en una extensión del lenguaje de tipos abstractos LPG (Langage de Programmation Générique) [Ber 79, 82a, 82b].
6. La semántica operacional dada al lenguaje pensamos que permite el estudio de propiedades tales como: invariantes, trayectorias, terminación, "deadlock",... en el estilo de las investigaciones realizadas por J. Sifakis [Sif 80], sobre los sistemas de transición de estado.
7. La propiedad de divergencia i.e. la posibilidad, para el proceso de estar calculando indefinidamente sin comunicarse con el medio exterior, es indecidible. Una demostración de esto se encuentra en [Per 84]. Sin embargo nosotros creemos que existe una gran familia para la cual esta propiedad es decidible. Una vía de como decidir esta propiedad se encuentra en [Per 84] a través del estudio de funciones noetherianas y su relación con el problema de decidibilidad de terminación de una regla de transición de términos.

## BIBLIOGRAFIA

- |                  |                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| [Aus-Bou 82]     | D. Austrey - G. Boudol<br><i>"Algèbre de Processus et Synchronisation"</i><br>A ser publicado en TCS                                                   |
| [Bér 79]         | D. Bert<br><i>"La programmation Générique"</i><br>Thèse d'Etat (Juin 1979). Université de Grenoble.                                                    |
| [Ber 82a]        | D. Bert<br><i>"Refinements of Generic Specifications with Algebraic Tools"</i><br>R. R. 335, IMAG 1982.                                                |
| [Ber 82b]        | D. Bert<br><i>"Generic Programming: A tool for Designing Universal Operator Application to Program Algebra"</i><br>R. R. 336, IMAG 1982.               |
| [Hoa 78]         | C.A.R. Hoare<br><i>"Communication Sequential Processes"</i><br>C.A.M. (Vol. 21, N° 8), 1978.                                                           |
| [Hoa-Bro-Ros 81] | C.A.R. Hoare - S.D. Brookes - A. W. Roscoe<br><i>"A theory of Communicating Sequential Processes"</i><br>PRG Monograph N°16, 1981 - Oxford University. |

- [Hoa 81a] C.A.R. Hoare  
*"A model for Communicating Sequential Processes"*  
 PRG Monograph N°22, 1981 - Oxford University.
- [Jor 81a] Ph. Jorrand  
*"Bases for the Specification of Communicating Processes"*  
 ICS 81, London - April 1981.
- [Jor 81b] Ph. Jorrand  
*"Specification of Communicating Processes and Process Implementation Correctness"*  
 International Symposium on Programming.  
 5th Colloquium, Turin - April 1982 Proceedings LNC 137.
- [Jor 84a] Ph. Jorrand  
*"Communicating Processes as Term Algebras"*  
 A ser publicado
- [Jor 84b] Ph. Jorrand  
*"FP2: Functional Parallel Programming Based on Term Substitution"*  
 A ser publicado.
- [Mil 80] R. Milner  
*"A calculus of Communicating Systems"*  
 LNCS 92, 1980.
- [Per 84] J. M. Pereira  
*"Processus Communicants: un langage formel et ses modèles. Problèmes d'Analyse".*  
 Thèse de 3 ème cycle (Juin 1984).  
 Université de Grenoble
- [Rob 65] J. A. Robinson  
*"A Machine-Oriented Logic Based on the Resolution Principle"*  
 JACM 12, 1965.
- [Rou-Bro 81] W.C. Rounds - S. D. Brookes  
*"Possibles Futures, Acceptances, Refusals and Communicating Processes".*  
 Proceedings of 22nd Annual IEEE Symposium on Foundations of Computer Science, 1981.
- [Sif 82] J. Sifakis  
*"An Unified Approach for Studying the Properties of Transition Systems"*  
 TCS 18 (1982)
- [Kel 76] R.M. Keller  
*"Formal verification of Parallel Programs"*  
 CACM (Vol. 19 N°7) 1976