

Arquitectura de Sistemas de Software y Tecnologías Asociadas

Una Introducción a las álgebras de procesos CCS y mCRL2

Alejandro Sánchez
asanchez@unsl.edu.ar

Departamento de Informática
Universidad Nacional de San Luis

Maestría en Informática
Universidad Nacional de San Juan
8-9 Nov 2013

Estructura de la presentación

- 1 Algebras de Proceso
 - Motivación
 - ¿Qué son las álgebras de procesos?
- 2 Calculus of Communicating Systems
 - Introducción
 - Procesos concurrentes
 - Procesos concurrentes con parámetros de datos
- 3 Introducción a mCRL2
 - Procesos secuenciales
 - Procesos concurrentes

Estructura de la presentación

- 1 Algebras de Proceso
 - Motivación
 - ¿Qué son las álgebras de procesos?

- 2 Calculus of Communicating Systems
 - Introducción
 - Procesos concurrentes
 - Procesos concurrentes con parámetros de datos

- 3 Introducción a mCRL2
 - Procesos secuenciales
 - Procesos concurrentes

Álgebra de procesos en términos generales

(Process algebra)

*“El **álgebra de procesos** se define, en términos generales, como el estudio del comportamiento de sistemas distribuidos por medios algebraicos.”*

Comportamiento es el registro estructurado de interacciones

Álgebras de procesos en términos más técnicos

Proceso

... una representación abstracta
del comportamiento del sistema

Algebra

... una estructura matemática
que satisface un conjunto particular de axiomas

Algebra de procesos

... un *framework* para la especificación y manipulación de
términos de procesos (*process terms*)
inducidos por una colección de símbolos de operadores,
incluyendo una teoría operacional y axiomática.

Proceso

Un proceso es una descripción de como evolucionan las capacidades de interacción de un sistema, es decir, su comportamiento (*behaviour*).

Analogía: autómata finito y lenguajes regulares

Ejemplo: $E \triangleq a.b + c.E$

Acción

(*Action*)

$$\alpha ::= \tau \mid \mathbf{a} \mid \alpha \mid \alpha$$

- Unidad elemental de comportamiento, atómica (sin duración), y observable
- Latencia de una interacción
- τ es una acción no observable
- $\mathbf{a} \mid \mathbf{b} \mid \dots \mid \mathbf{z}$ representa una colección de acciones, llamada multi-acción (*multi-action*), que ocurren simultáneamente

Enfoques para dar semántica

¿Cuál es el significado de la descripción de un programa?

Se modela como

- Operacional: la ejecución de una máquina abstracta, por ej. LTS
- Denotacional: una función que transformando entradas en salidas
- Axiomatic: enfatiza métodos de prueba de corrección

Ingredientes del *framework*

- *Transition systems*: representación operacional del comportamiento del sistema a través de grafos etiquetados
- *Behavioural equivalences*: distinguir estados entre sistemas de transición
- *Process terms*: representación algebraica de sistemas de transición
- *Structural operational semantics*: reglas de prueba inductiva para proveer a cada término de proceso con su sistema de transición pretendido
- *Equational theory*: teoría axiomática de procesos, expresado en una lógica ecuacional sobre términos de procesos

Instanciaciones del Framework más notables

- *Calculus of Communicating Systems* – CCS
Robin Milner, 1973 - 1980
- *Communicating Sequential Processes* – CSP
Tony Hoare, 1976
- *Algebra of Communicating Processes*: – ACP
Jan Bergstra, 1982

Estructura de la presentación

- 1 Algebras de Proceso
 - Motivación
 - ¿Qué son las álgebras de procesos?
- 2 Calculus of Communicating Systems
 - Introducción
 - Procesos concurrentes
 - Procesos concurrentes con parámetros de datos
- 3 Introducción a mCRL2
 - Procesos secuenciales
 - Procesos concurrentes

Introducción

Calculus of Communicating Systems (CCS)

Publicado en 1980 por Robin Milner

Acciones:

$$\alpha ::= a \mid \bar{a} \mid \tau,$$

para $a \in Act$, con Act un conjunto de nombres

Procesos:

- Comunicación por *binary handshake* de acciones complementarias
- No distingue entre *termination* y *deadlock*
- Sin composición secuencial ($P.Q$), pero con prefijo de acciones ($a.P$)

Sintaxis

El conjunto de procesos \mathbb{P} se expresa como un término generado por la siguiente sintaxis

$$E ::= A(x_1, \dots, x_n) \mid a.E \mid \sum_{i \in I} E_i \mid E_0 \mid E_1 \mid \text{new } K \ E$$

para $a \in \alpha$ y $K \subseteq \text{Act}$

Abreviaturas

$$0 = \sum_{i \in \emptyset} E_i$$

$$E_0 + E_1 = \sum_{i \in \{0,1\}} E_i$$

Activación de procesos

(*Process activation*)

$$E ::= A(x_1, \dots, x_n) \mid a.E \mid \sum_{i \in I} E_i \mid E_0 \mid E_1 \mid \text{new } K \ E$$

Asume declaraciones de procesos

$$A(\vec{x}) \triangleq E_A$$

con $fn(E_A) \subseteq \vec{x}$,

donde $fn(P)$ es el conjunto de variables libres de P

Ejemplo:

$$A(a, b, c) \triangleq a.b.\mathbf{0} + c.A(d, e, f)$$

Activación de procesos - sustitución sintáctica

Dadas dos secuencias de nombres, \vec{a} y \vec{b} ,
del largo n , con \vec{a} distintos entre sí,
y dada la expresión de procesos E

La **sustitución sintáctica** $\{\vec{b}/\vec{a}\}E$
es el resultado de reemplazar a_i por b_i en E

Dada una declaración de proceso $A(\vec{a}) \triangleq E_A$,
la expresión $A(\vec{b})$ significa lo mismo que $\{\vec{b}/\vec{a}\}E_A$

Prefijo de acción

(*Action prefix*)

$$E ::= A(x_1, \dots, x_n) \mid a.E \mid \sum_{i \in I} E_i \mid E_0 \mid E_1 \mid \text{new } K \ E$$

$$\frac{}{a.E \xrightarrow{a} E}$$

$$\frac{\{\vec{b}/\vec{a}\} E_A \xrightarrow{a} E'}{A(\vec{b}) \xrightarrow{a} E'} \quad (\text{ if } A(\vec{a}) \triangleq E_A)$$

Sumatoria

(*Summation*)

$$E ::= A(x_1, \dots, x_n) \mid a.E \mid \sum_{i \in I} E_i \mid E_0 \mid E_1 \mid \text{new } K \ E$$

$$\frac{E_j \xrightarrow{a} E'_j}{\sum_{i \in I} E_i \xrightarrow{a} E'_j} \quad (\text{ where } j \in I)$$

Composición paralela

(*Parallel composition*)

$$E ::= A(x_1, \dots, x_n) \mid a.E \mid \sum_{i \in I} E_i \mid E_0 \mid E_1 \mid \text{new } K \ E$$

$$\frac{E_0 \xrightarrow{a} E'_0}{E_0 \mid E_1 \xrightarrow{a} E'_0 \mid E_1}$$

$$\frac{E_1 \xrightarrow{a} E'_1}{E_0 \mid E_1 \xrightarrow{a} E_0 \mid E'_1}$$

$$\frac{E_0 \xrightarrow{a} E'_0 \quad E_1 \xrightarrow{\bar{a}} E'_1}{E_0 \mid E_1 \xrightarrow{\tau} E'_0 \mid E'_1}$$

Restricción

(*Restriction*)

$$E ::= A(x_1, \dots, x_n) \mid a.E \mid \sum_{i \in I} E_i \mid E_0 \mid E_1 \mid \text{new } K \ E$$

$$\frac{E \xrightarrow{a} E'}{\text{new } K \ E \xrightarrow{a} \text{new } K \ E'} \quad (\text{ if } a \notin \{K, \bar{K}\})$$

Máquina expendedora de bebidas ME

Máquina expendedora de té y café

$$ME \triangleq p.MET + p.MEC$$

$$MET \triangleq \overline{te}.ME$$

$$MEC \triangleq p.\overline{cafe}.ME$$

Máquina expendedora de bebidas ME

Máquina expendedora de té y café

$$ME \triangleq p.MET + p.MEC$$

$$MET \triangleq \overline{te}.ME$$

$$MEC \triangleq p.\overline{cafe}.ME$$

¿Qué es más caro?

Máquina expendedora de bebidas ME

Máquina expendedora de té y café

$$ME \triangleq p.MET + p.MEC$$

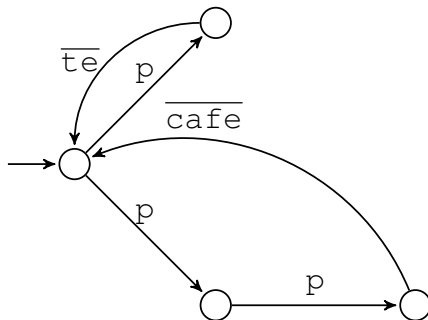
$$MET \triangleq \overline{te}.ME$$

$$MEC \triangleq p.\overline{cafe}.ME$$

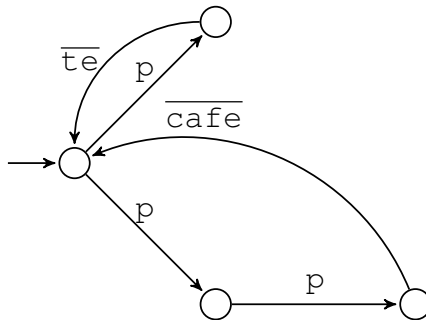
¿Qué es más caro?

¿Cómo es el LTS correspondiente?

LTS de la máquina expendedora de bebidas ME

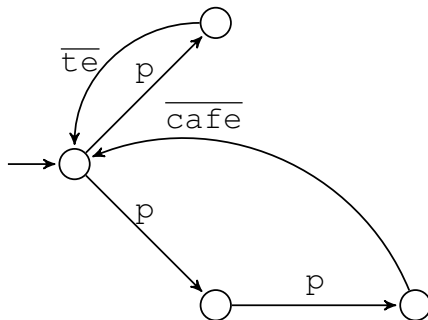


LTS de la máquina expendedora de bebidas ME



¿Puede dar un LTS determinístico *equivalente*?

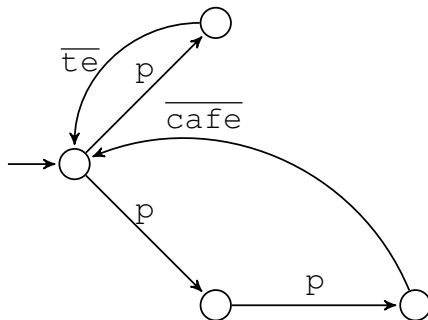
LTS de la máquina expendedora de bebidas ME



¿Puede dar un LTS determinístico *equivalente*?

En caso de poder: ¿Cómo es la expresión CCS correspondiente?

LTS de la máquina expendedora de bebidas ME



¿Puede dar un LTS determinístico *equivalente*?

En caso de poder: ¿Cómo es la expresión CCS correspondiente?

¿Qué está mal con este sistema?

Investigador

El investigador que toma té o café

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$IT \triangleq te.I$$

$$IC \triangleq \bar{p}.cafe.I$$

Investigador

El investigador que toma té o café

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$IT \triangleq te.I$$

$$IC \triangleq \bar{p}.cafe.I$$

Realice el LTS correspondiente

El box

$$BOX \triangleq I \mid ME$$

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$IT \triangleq te.I$$

$$IC \triangleq \bar{p}.cafe.I$$

$$ME \triangleq p.MET + p.MEC$$

$$MET \triangleq \overline{te}.ME$$

$$MEC \triangleq p.\overline{cafe}.ME$$

Un escenario

$$I \mid ME$$

El box

$$BOX \triangleq I \mid ME$$

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$IT \triangleq te.I$$

$$IC \triangleq \bar{p}.cafe.I$$

$$ME \triangleq p.MET + p.MEC$$

$$MET \triangleq \overline{te}.ME$$

$$MEC \triangleq p.\overline{cafe}.ME$$

Un escenario

$$I \mid ME \\ (\bar{p}.IT + \bar{p}.IC) \mid (p.MET + p.MEC)$$

El box

$$BOX \triangleq I \mid ME$$

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$IT \triangleq te.I$$

$$IC \triangleq \bar{p}.cafe.I$$

$$ME \triangleq p.MET + p.MEC$$

$$MET \triangleq \overline{te}.ME$$

$$MEC \triangleq p.\overline{cafe}.ME$$

Un escenario

$$\begin{aligned} & I \mid ME \\ & (\bar{p}.IT + \bar{p}.IC) \mid (p.MET + p.MEC) \\ & IC \mid MEC = \bar{p}.cafe.I \mid p.\overline{cafe}.ME \end{aligned}$$

El box

$$BOX \triangleq I \mid ME$$

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$ME \triangleq p.MET + p.MEC$$

$$IT \triangleq te.I$$

$$MET \triangleq \overline{te}.ME$$

$$IC \triangleq \bar{p}.cafe.I$$

$$MEC \triangleq p.\overline{cafe}.ME$$

Un escenario

$$\begin{aligned} & I \mid ME \\ & (\bar{p}.IT + \bar{p}.IC) \mid (p.MET + p.MEC) \\ & IC \mid MEC = \bar{p}.cafe.I \mid p.\overline{cafe}.ME \\ & \quad cafe.I \mid \overline{cafe}.ME \end{aligned}$$

El box

$$BOX \triangleq I \mid ME$$

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$ME \triangleq p.MET + p.MEC$$

$$IT \triangleq te.I$$

$$MET \triangleq \overline{te}.ME$$

$$IC \triangleq \bar{p}.cafe.I$$

$$MEC \triangleq p.\overline{cafe}.ME$$

Un escenario

$$\begin{aligned}
 & I \mid ME \\
 & (\bar{p}.IT + \bar{p}.IC) \mid (p.MET + p.MEC) \\
 & IC \mid MEC = \bar{p}.cafe.I \mid p.\overline{cafe}.ME \\
 & \quad cafe.I \mid \overline{cafe}.ME \\
 & I \mid ME
 \end{aligned}$$

El box

$$BOX \triangleq I \mid ME$$

$$I \triangleq \bar{p}.IT + \bar{p}.IC$$

$$ME \triangleq p.MET + p.MEC$$

$$IT \triangleq te.I$$

$$MET \triangleq \overline{te}.ME$$

$$IC \triangleq \bar{p}.cafe.I$$

$$MEC \triangleq p.\overline{cafe}.ME$$

Un escenario

$$\begin{aligned}
 & I \mid ME \\
 & (\bar{p}.IT + \bar{p}.IC) \mid (p.MET + p.MEC) \\
 & IC \mid MEC = \bar{p}.cafe.I \mid p.\overline{cafe}.ME \\
 & cafe.I \mid \overline{cafe}.ME \\
 & I \mid ME
 \end{aligned}$$

¿Puede ocurrir un deadlock?

El box descafeinado

$$BOX \triangleq I \mid new \{cafe\} ME$$

¿Qué ocurre al agregar la restricción?

Extensión con parámetros de datos

Dado un universo de datos V , un conjunto V_e de expresiones sobre V , y una función de evaluación $Val : V_e \rightarrow V$

Buffer con incremento

$$B \triangleq in(x).B'_x$$

$$B'_v \triangleq \overline{out}\langle v + 1 \rangle.B$$

La extensión de \mathbb{P} a \mathbb{P}_V

- Dos formas de prefijo: $a(x).E$ y $\bar{a}\langle v \rangle.E$
- Parámetros de datos en declaración de procesos,
 $A_S(x_1, \dots, x_n) \triangleq E_A$, con $S \in V$
- Condicionales:
 - *if b then P*
 - *if b then P_1 else P_2*

Semántica adicional - Prefijo

$$\frac{}{a(x).E \xrightarrow{a(v)} \{v/x\}E} \quad (\text{con } v \in V)$$

$$\frac{}{\bar{a}\langle e \rangle.E \xrightarrow{\bar{a}\langle v \rangle} E} \quad (\text{con } Val(e) = v)$$

Semántica adicional - Condicionales

$$\frac{E1 \xrightarrow{a(v)} E1'}{\text{if } b \text{ then } E1 \text{ else } E2 \xrightarrow{a} E1'} \quad (\text{con } Val(b) = \text{true})$$

$$\frac{E2 \xrightarrow{a(v)} E2'}{\text{if } b \text{ then } E1 \text{ else } E2 \xrightarrow{a} E2'} \quad (\text{con } Val(b) = \text{false})$$

Codificación en \mathbb{P}

Como sería la traducción

El proceso

$$a(x).E$$

se traduce a

$$\sum_{v \in V} a(v).E'$$

donde E' es la traducción de E

Estructura de la presentación

- 1 Algebras de Proceso
 - Motivación
 - ¿Qué son las álgebras de procesos?
- 2 Calculus of Communicating Systems
 - Introducción
 - Procesos concurrentes
 - Procesos concurrentes con parámetros de datos
- 3 **Introducción a mCRL2**
 - **Procesos secuenciales**
 - **Procesos concurrentes**

El entorno mCRL2

mCRL2

- Basado principalmente en ACP y en CCS
- Extendido con datos y tiempo real
- Tiene como lógica de especificación a la lógica μ -calculus
- Herramientas para la simulación y verificación de sistemas reactivos

<http://www.mcrl2.org/>

Acción

$$\alpha ::= \tau \mid a \mid a(d) \mid \alpha \mid \alpha$$

- Se ejecutan de manera atómica
- Puede tener parámetros de datos
- La *multiaction*, $\alpha \mid \alpha$, es la unidad de interacción

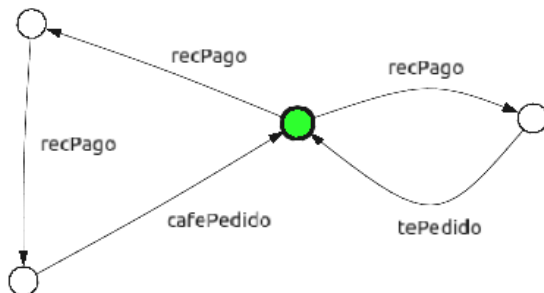
Máquina expendedora no determinística

```
act
  recPago, tePedido, cafePedido;

proc
  ME = recPago.MET + recPago.MEC;
  MET = tePedido.ME;
  MEC = recPago.cafePedido.ME;

init ME;
```

LTS Máquina expendedora no determinista – *ltsgraph*



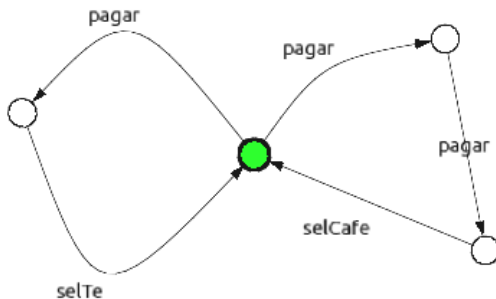
Investigador

```
act
  pagar, selTe, selCafe;

proc
  I = pagar.IT + pagar.IC;
  IT = selTe.I;
  IC = pagar.selCafe.I;

init I;
```

LTS investigador – *Itsgraph*



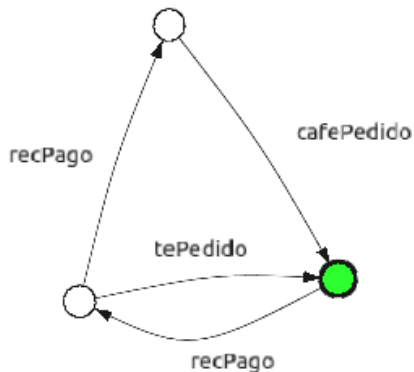
Máquina expendedora determinística

```
act
  recPago, tePedido, cafePedido;

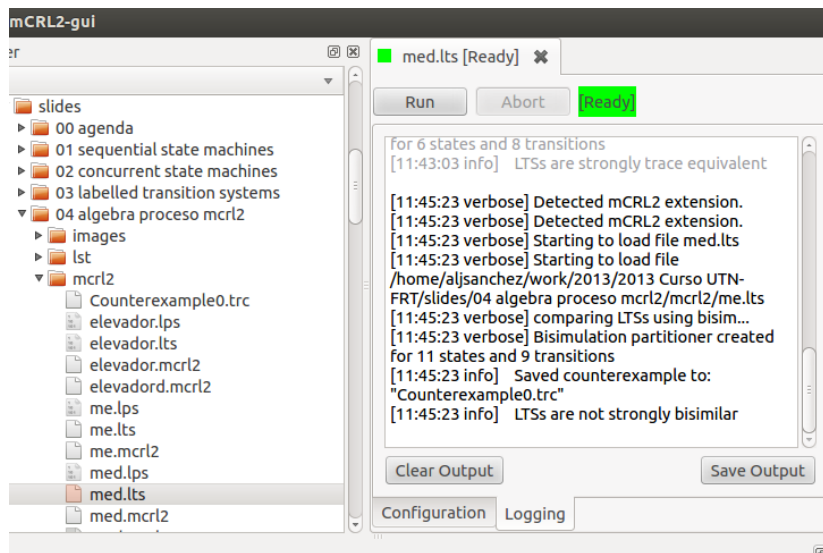
proc
  ME = recPago.(tePedido.ME + recPago.cafePedido.ME);

init ME;
```

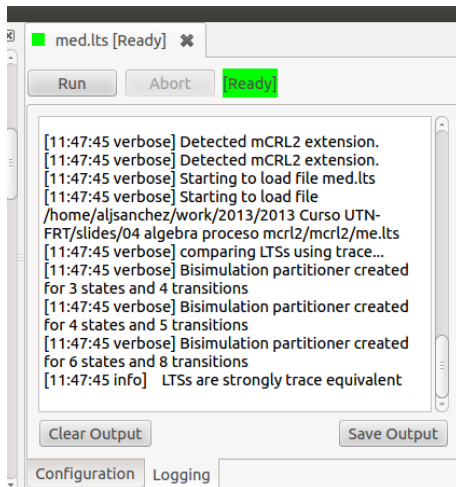
LTS Máquina expendedora determinista – *ltsgraph*



Máquinas expendedoras – *ltscompare bisim*



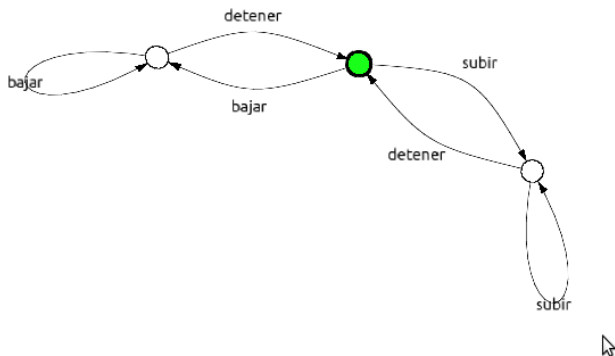
Máquinas expendedoras – *ltscompare trace*



Elevador simple

```
act
  subir, bajar, detener;
proc
  ED = subir.ES + bajar.EB;
  ES = subir.ES + detener.ED;
  EB = bajar.EB + detener.ED;
init ED;
```

LTS Elevador simple – graficador



LTS Elevador simple – simulador

LpsXSim

Transitions

Action	State Change
detener	s3_ED := 1
subir	

Trace

#	Action	State Change
0		s3_ED := 1
1	bajar	s3_ED := 3
2	bajar	
3	bajar	
4	detener	s3_ED := 1
5	subir	s3_ED := 2
6	subir	
7	detener	s3_ED := 1

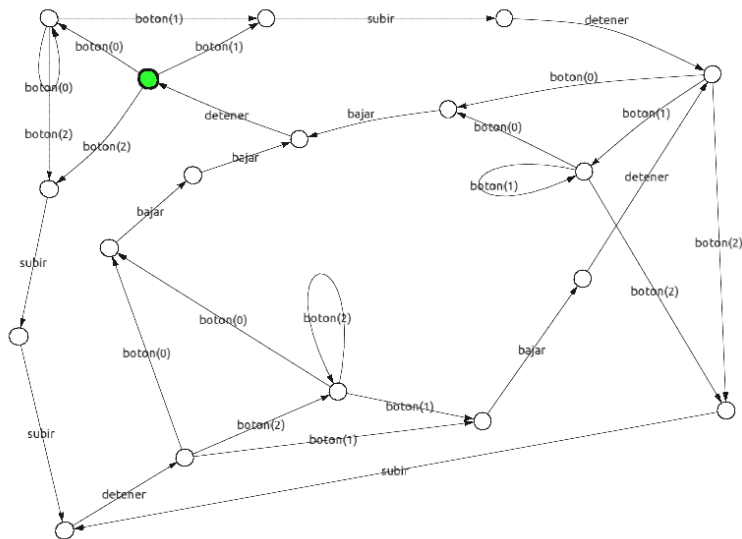
Current State

Parameter	Value
s3_ED	2

Elevador con datos

```
act
  boton : Int;
  detener, subir, bajar;
proc
  ED(piso:Int) =
    sum pedido:Int.
      (pedido>=0 && pedido<3)->boton(pedido).
        ((piso>pedido)->EB(piso,pedido)<>
          ((piso<pedido)->ES(piso,pedido)<>ED(piso))) );
  EB(piso:Int,pedido:Int) =
    (pedido==piso)->detener.ED(piso)
    <>bajar.EB(piso-1,pedido);
  ES(piso:Int,pedido:Int) =
    (pedido==piso)->detener.ED(piso)
    <>subir.ES(piso+1,pedido);
init ED(0);
```

LTS Elevador con datos



Simulador Elevador con datos

LpsXSim

Transitions

Action	State Change
boton(0)	s3_ED := 2, pedido_ED := 0
boton(1)	s3_ED := 2, pedido_ED := 1
boton(2)	s3_ED := 2, pedido_ED := 2

Trace

#	Action	State Change
0		s3_ED := 1, piso_ED := 0
1	boton(0)	s3_ED := 2, pedido_ED := 0
2	boton(2)	pedido_ED := 2
3	subir	s3_ED := 4, piso_ED := 1
4	subir	piso_ED := 2
5	detener	s3_ED := 1
6	boton(1)	s3_ED := 2, pedido_ED := 1
7	bajar	s3_ED := 3, piso_ED := 1
8	detener	s3_ED := 1

Current State

Parameter	Value
s3_ED	1
pedido_ED	_
piso_ED	1

Operadores de comunicación

- **Paralelo**
Colocar un proceso *al lado* de otro
- **Comunicación**
Indicar que multiacciones sincronizan (comunican) procesos
- **Acciones permitidas**
Permite un conjunto de acciones y bloquea las otras
- **Acciones bloqueadas**
Bloquea un conjunto de acciones y permite las otras
- **Renombrado**
Renombra acciones
- **Ocultamiento**
Transforma acciones en invisibles (τ)

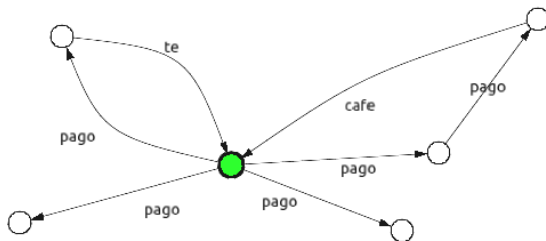
Composición paralela y comunicación

```
act
  pagar, selTe, selCafe;
  recPago, tePedido, cafePedido;
  pago, te, cafe;
proc
  I = ...
  ME = ...
init comm(
  { pagar|recPago -> pago,
    selTe|tePedido -> te,
    selCafe|cafePedido -> cafe}, I || ME);
```

Demasiados eventos !

Operador permitir

```
...  
init  
  allow({pago,te,cafe},  
    comm({ pagar|recPago -> pago,  
      selTe|tePedido -> te,  
      selCafe|cafePedido -> cafe}, I || ME));
```



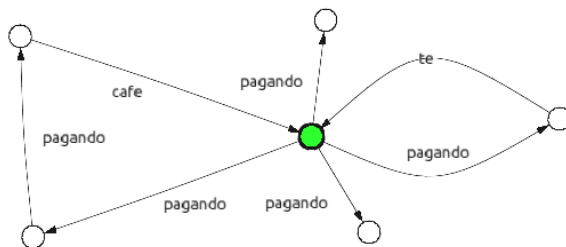
Operador bloquear

```
...  
init  
  block({pagar,recPago,selTe,  
        tePedido,selCafe,cafePedido},  
        comm({ pagar|recPago -> pago,  
              selTe|tePedido -> te,  
              selCafe|cafePedido -> cafe}, I || ME));
```

Operador rename

init

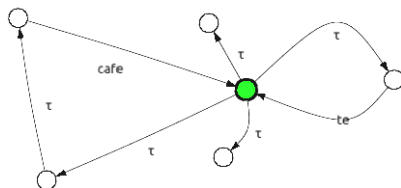
```
rename({pago -> pagando},
  allow({pago,te,cafe},
    comm({ pagar|recPago -> pago,
      selTe|tePedido -> te,
      selCafe|cafePedido -> cafe}, I || ME)));
```



Operador hide

```

init
hide({pago},
  allow({pago,te,cafe},
    comm({ pagar|recPago -> pago,
          selTe|tePedido -> te,
          selCafe|cafePedido -> cafe}, I || ME)));
  
```



Ejercicios

- Componer un nuevo proceso box, utilizando la **máquina expendedora determinística**.
- Definir el ambiente para el **elevador simple** y componer los dos procesos.