Una Introducción al Algebra de Procesos de mCRL2

Alejandro Sánchez

Departamento de Informática
Universidad Nacional de San Luis

Universidad Tecnológica Nacional
Facultad Regional Tucumán
14-15 Junio

Contenidos

Una Introducción al Algebra de Procesos de mCRL2

- Algebras de Proceso
- Calculus of Communicating Systems (CCS)
- mCRL2

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."

Soporta el trabajo del ingeniero

Estrategia de resolución de problemas del ingeniero:

- entender el problema
- construir un modelo matemático de él
- animar y razonar en tal modelo
- actualizar el modelo cuando sea necesario
- calcular una solución e implementarla

Discusión:

¿Cómo cree que se relaciona con los métodos ágiles?

Soporta el trabajo del ingeniero

Estrategia de resolución de problemas del ingeniero:

- entender el problema
- construir un modelo matemático de él
- animar y razonar en tal modelo
- actualizar el modelo cuando sea necesario
- calcular una solución e implementarla

Discusión:

¿Cómo cree que se relaciona con los métodos ágiles?

Action

$$\alpha ::= \tau \mid a \mid \alpha \mid \alpha$$

- Unidad elemental de comportamiento, atómica (sin duración), y observable
- Latencia de una interacción
- τ es una acción no observable
- $a \mid b \mid \ldots \mid z$ representa una colección de acciones que ocurren simultáneamente

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

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 correción

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.

Ingredientes del *frameworks*

- *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

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, pero con prefijo de acciones

Sintaxis

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

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

para $a \in \alpha$ y $K \subseteq Act$

Abreviaturas

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

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

Activación de procesos

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

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 - substitución 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 substitución 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 los mismo que $\{\vec{b}/\vec{a}\}E_A$

Prefijo de acción

(*Action prefix*)

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

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

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

Sumatoria

(*Summation*)

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

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

Composición paralela

(*Parallel composition*)

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

$$\frac{E_0 \xrightarrow{a} E_0'}{E_0 \between E_1 \xrightarrow{a} E_0' \between E_1} \qquad \frac{E_1 \xrightarrow{a} E_1'}{E_0 \between E_1 \xrightarrow{a} E_0 \between E_1'}$$

$$\frac{E_0 \xrightarrow{a} E_0' \qquad E_1 \xrightarrow{\bar{a}} E_1'}{E_0 \between E_1 \xrightarrow{\tau} E_0' \between E_1'}$$

Restricción

(*Restriction*)

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

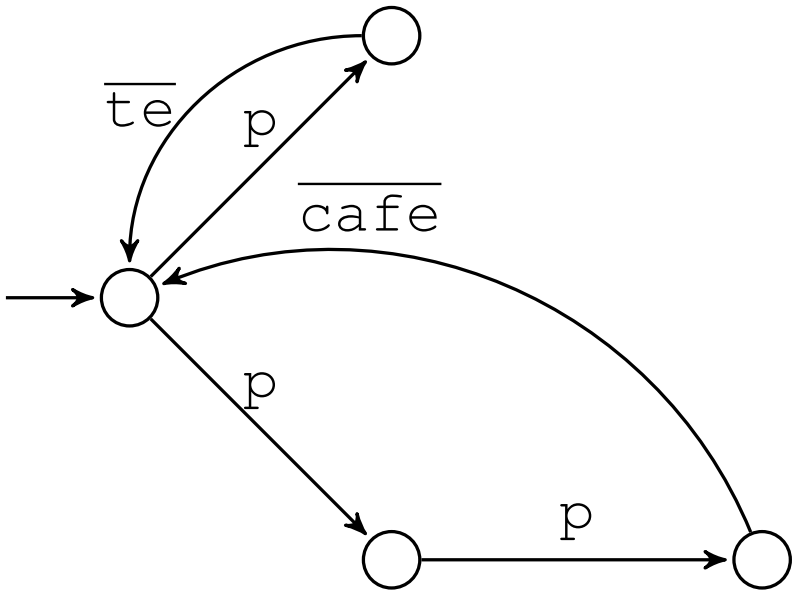
$$\frac{E \xrightarrow{a} E'}{\textit{new } K \; E \xrightarrow{a} \textit{new } K \; E'} \quad (\text{ if } a \notin \{K, \bar{K}\})$$

Máquina expendedora ME

$$ME \triangleq p.MET + p.MEC$$
$$MET \triangleq \overline{te}.ME$$
$$MEC \triangleq p.\overline{cafe}.ME$$

¿ Cómo es el LTS correspondiente ?

LTS de la máquina expendedora ME



¿ Puede dar un LTS determinístico *equivalente* ?
En caso de poder,
¿ Cómo es la expresión CCS correspondiente ?

¿Qué esta mal con este sistema?

Investigador

$$I \triangleq \overline{p}.IT + \overline{p}.IC$$
$$IT \triangleq te.I$$
$$IC \triangleq \overline{p}.cafe.I$$

Realice el LTS correspondiente

El box

$$BOX \triangleq I \mid ME$$

Un escenario

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

¿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 $Val : V_e \to V$

Buffer con incremento

$$B \triangleq in(x).B'(x)$$
$$B'(x) \triangleq \overline{out}\langle x+1\rangle.B$$

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

- Dos formas de prefijo: $a(x).E$ y $\overline{a}\langle v\rangle.E$
- Parámetros de datos en declaración de procesos, $A(x_1,\ldots,x_n) \triangleq E_A$, con cada x_i variable de datos distinta
- Condicionales:
  - *if b then P*
  - *if b then* P_1 *else* P_2

Semántica adicional - Prefijo

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

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

Semántica adicional - Condicionales

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

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

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

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

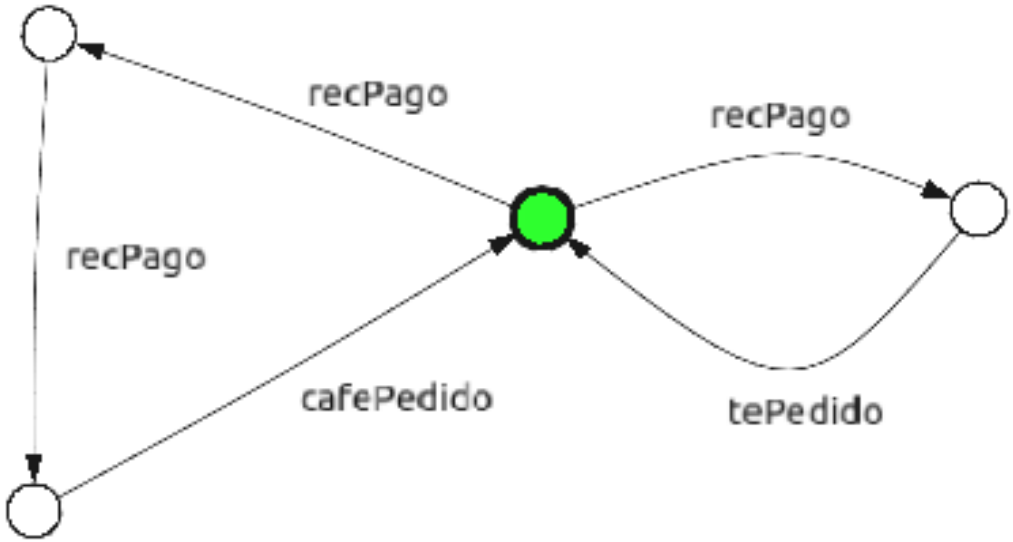
$$\alpha ::= \tau \mid a \mid a(d) \mid \alpha \mid \alpha$$

- 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;
```

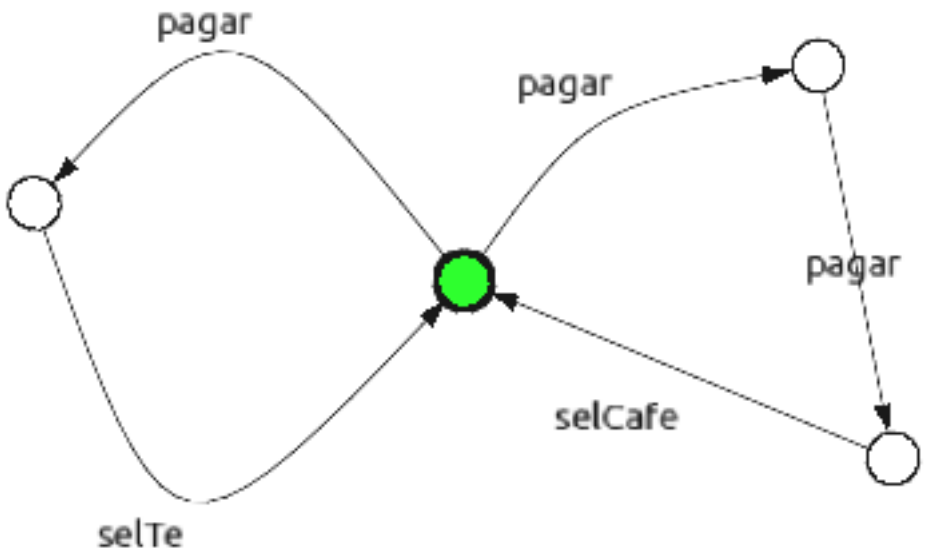
LTS Máquina expendedora no determinista – *ltsgraph*



Investigador

```
act
  pagar, selTe, selCafe;

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

init I;
```

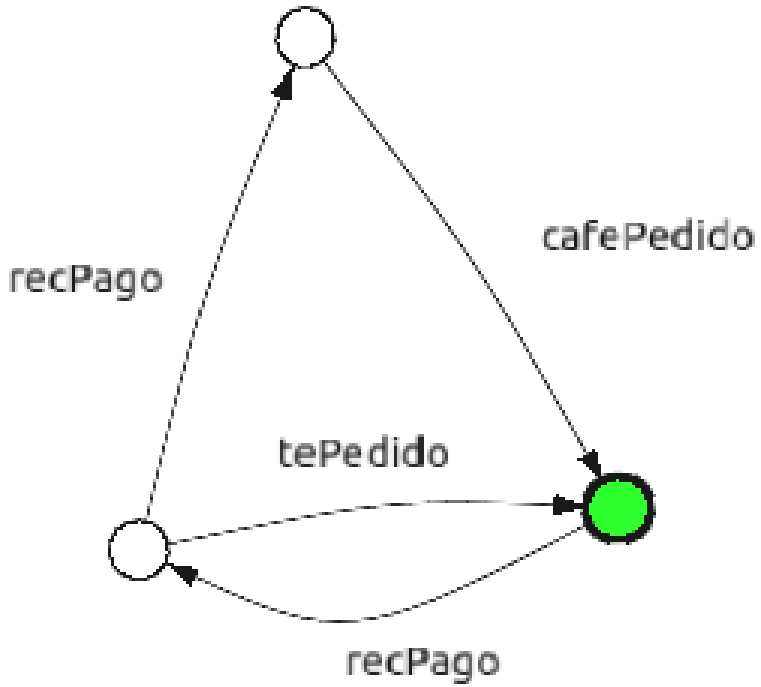
LTS investigador – *ltsgraph*



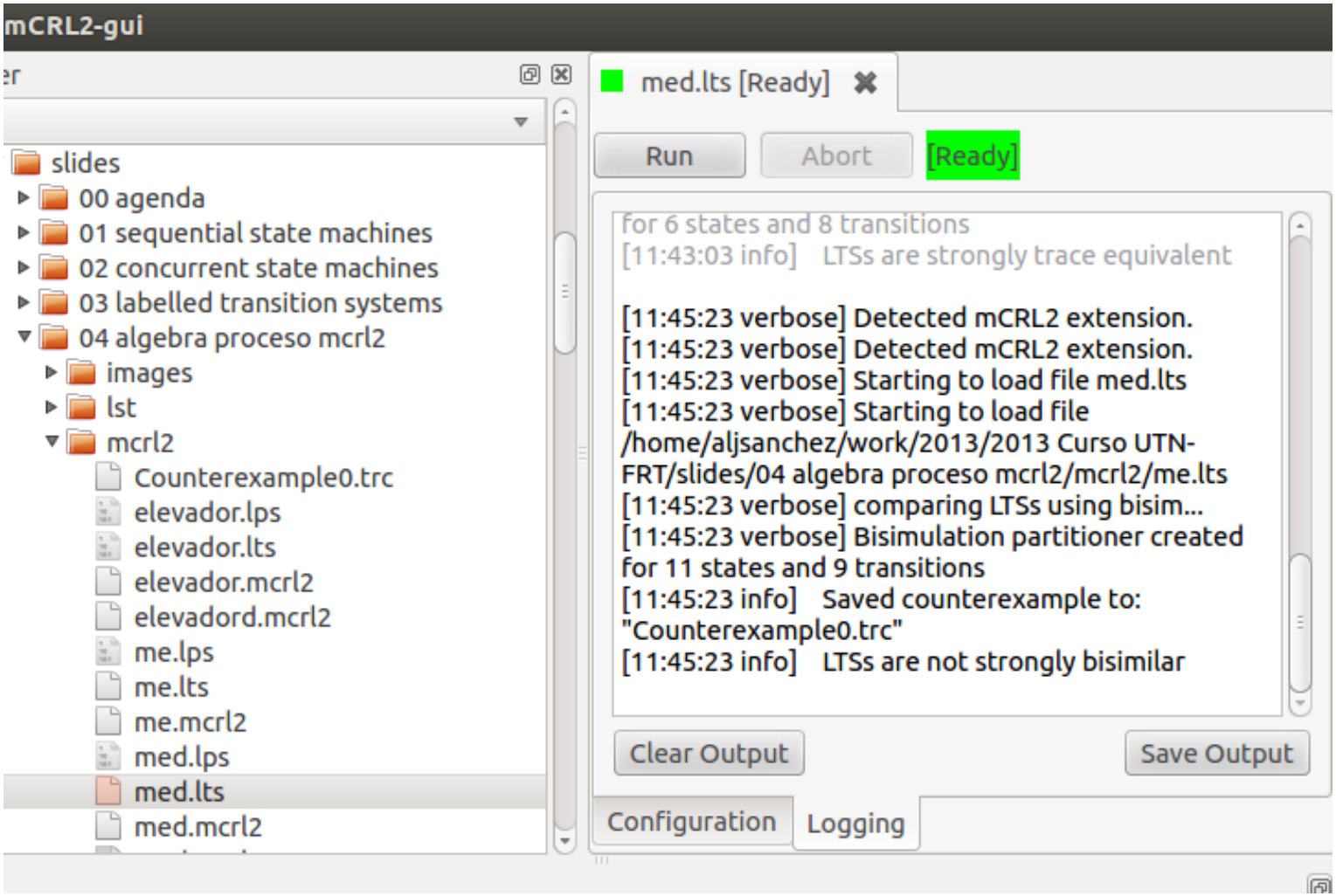
Máquina expendedora determinística

```
act
  recPago,tePedido,cafePedido;

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

init ME;
```

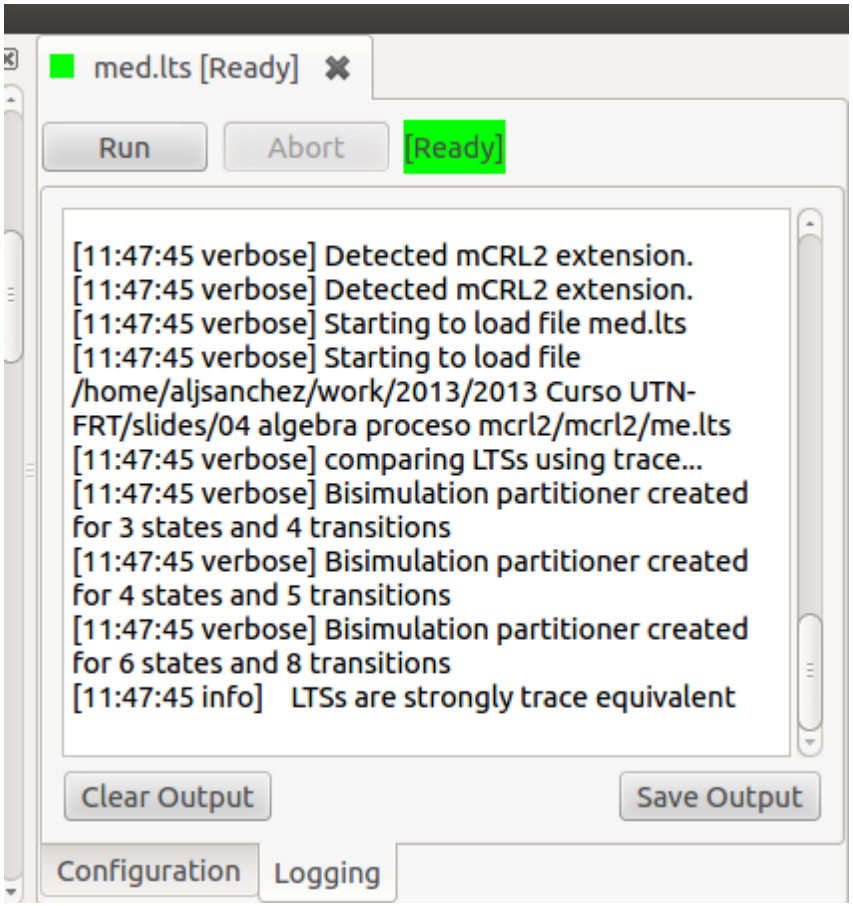
LTS Máquina expendedora determinista – *ltsgraph*



Máquinas expendedoras – *ltscompare bisim*



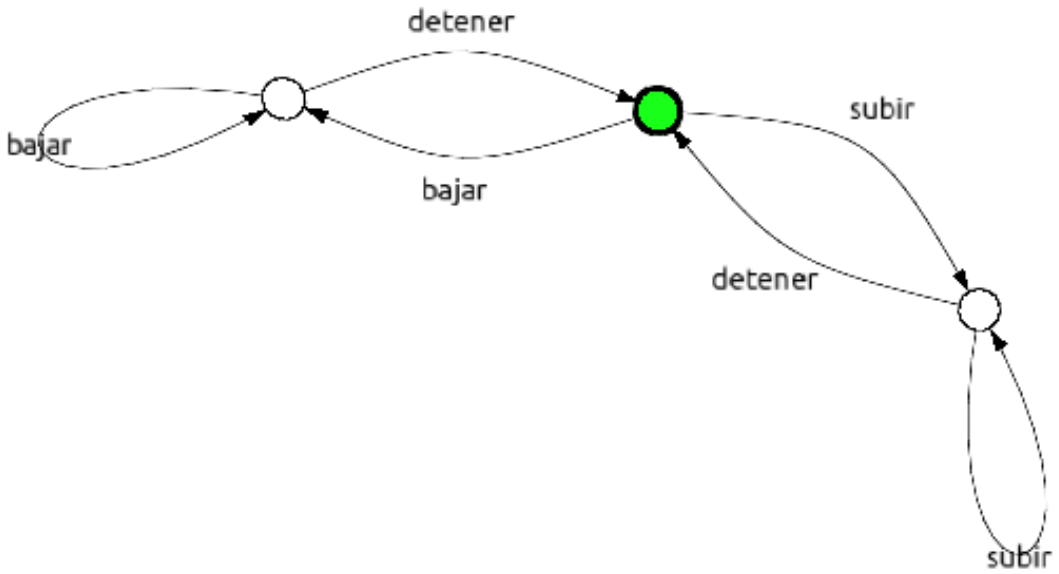
Máquinas expendedoras – *ltscompare trace*



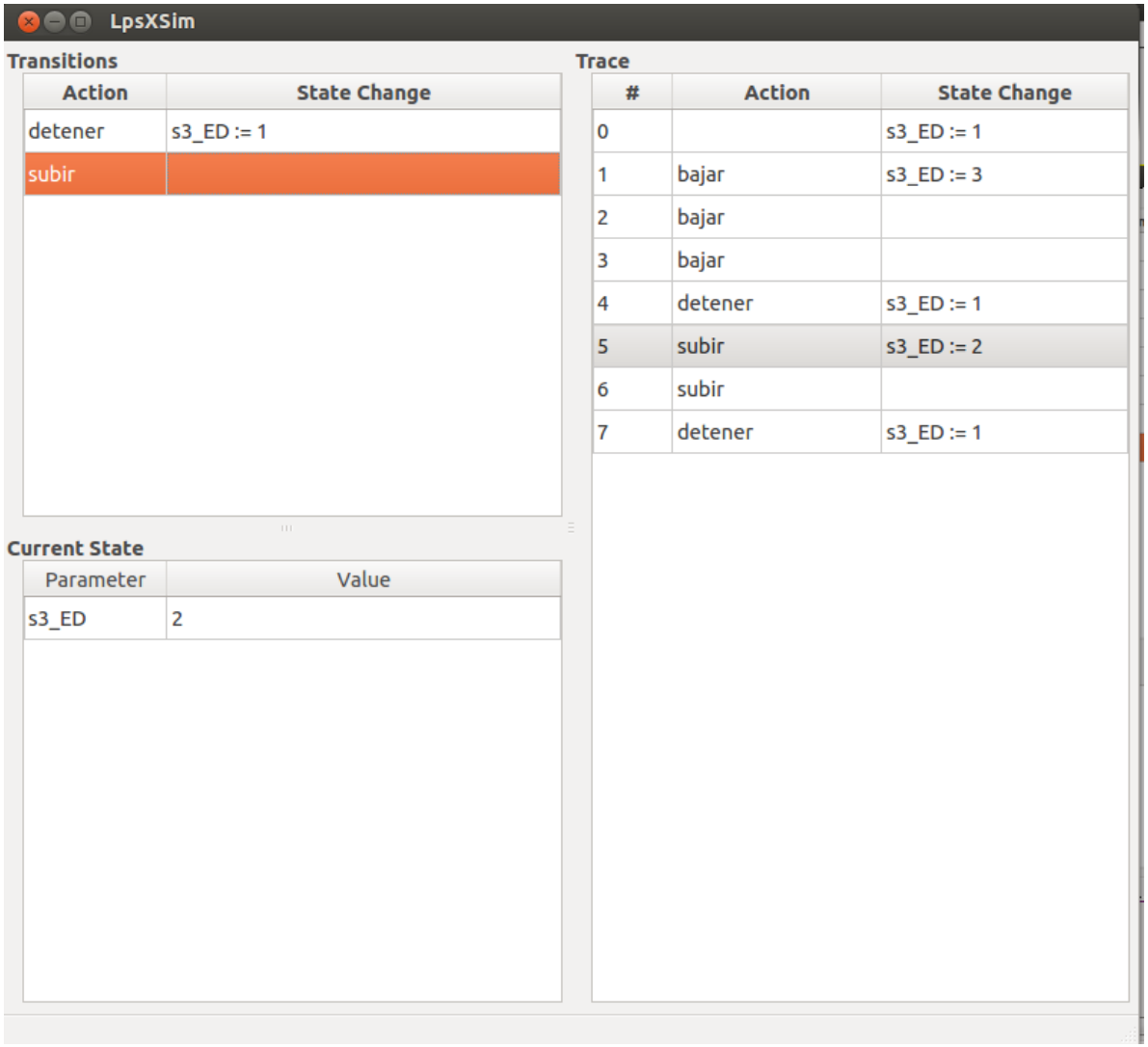
Elevador simple

```
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



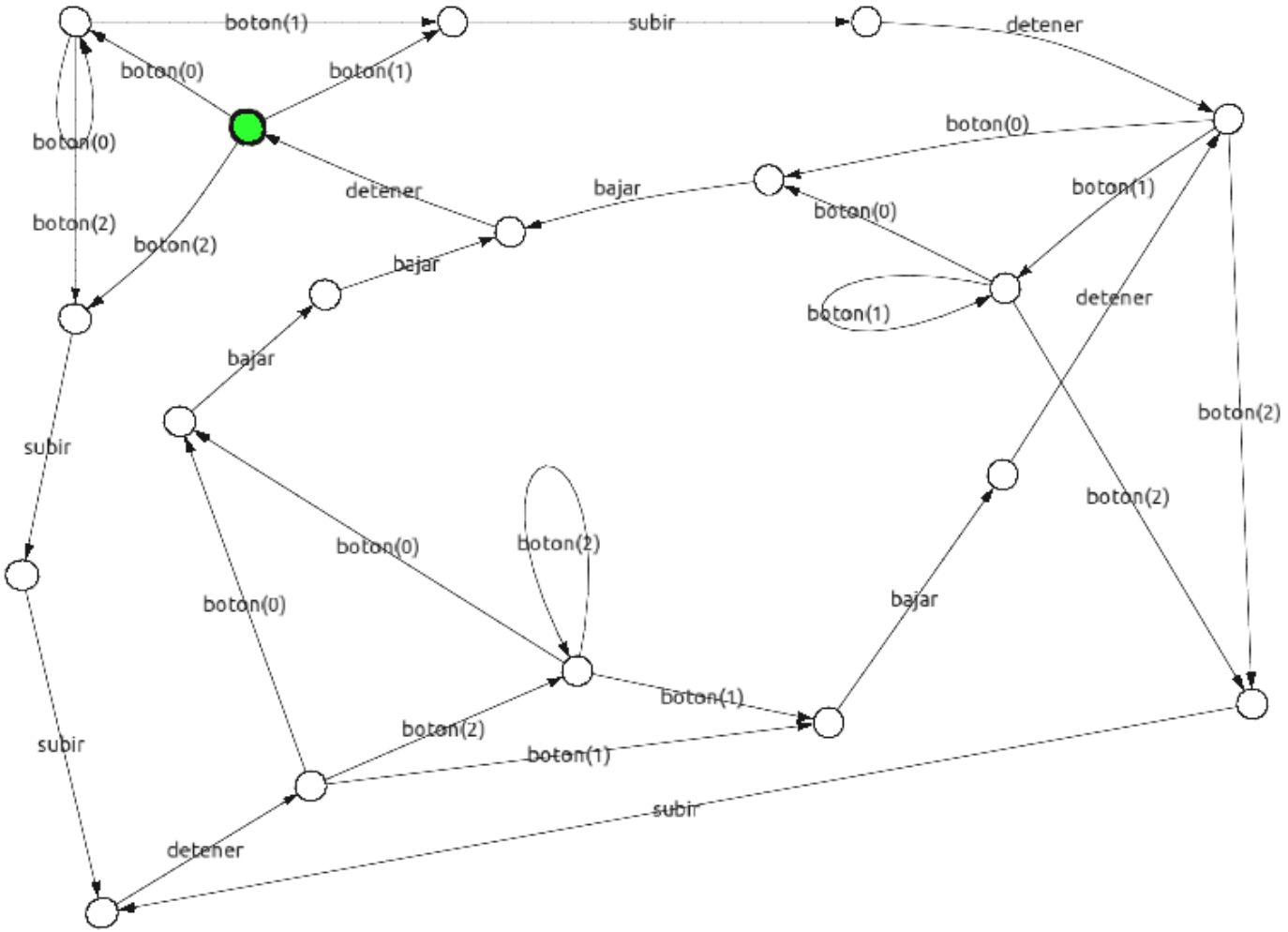
LTS Elevador simple – simulador



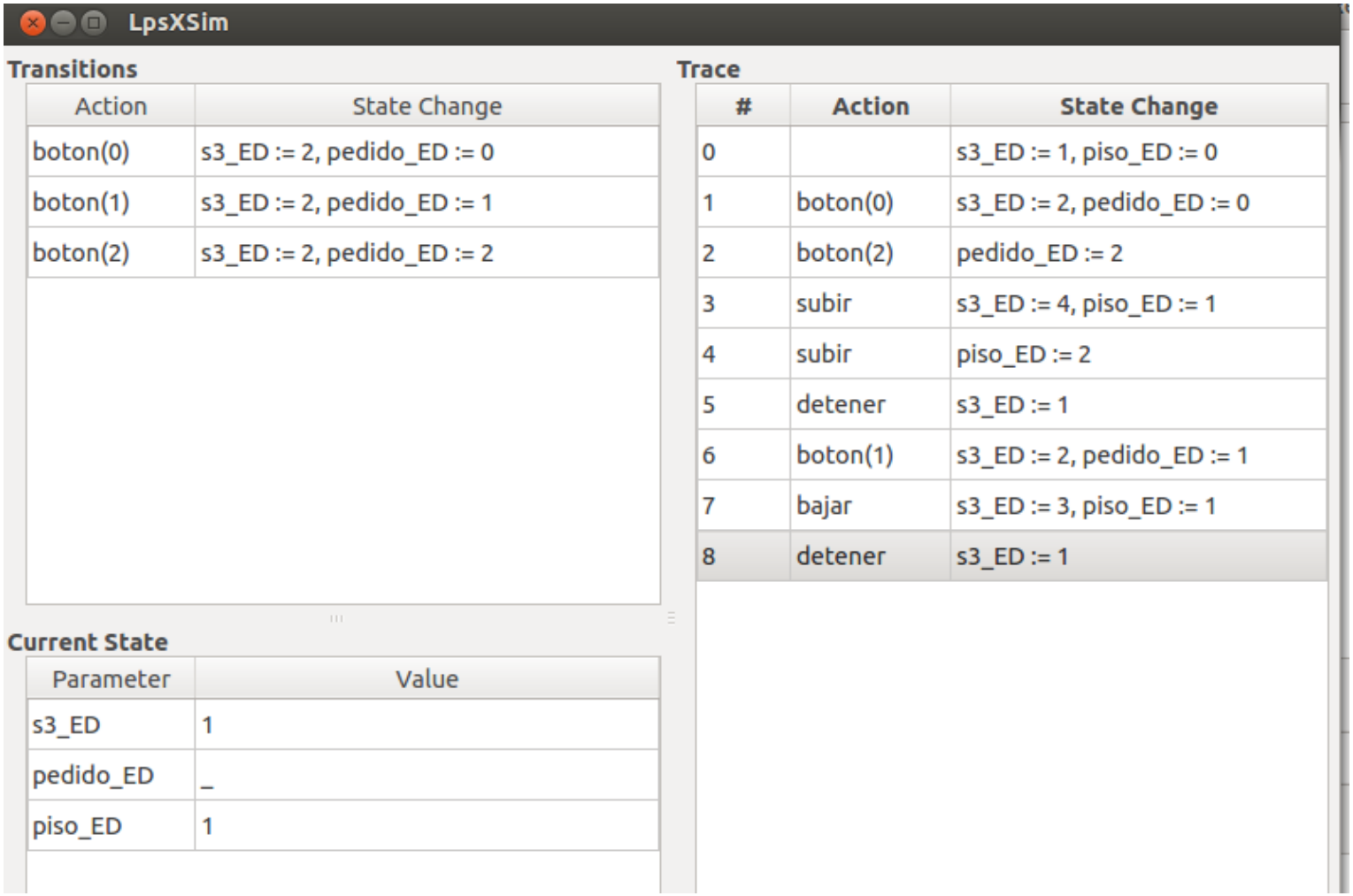
Elevador con datos

```
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



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**
  Bloque un conjunto de acciones y permite las otras
- **Renombrado**
  Renombra acciones
- **Ocultamiento**
  Transforma acciones en invisibles (τ)

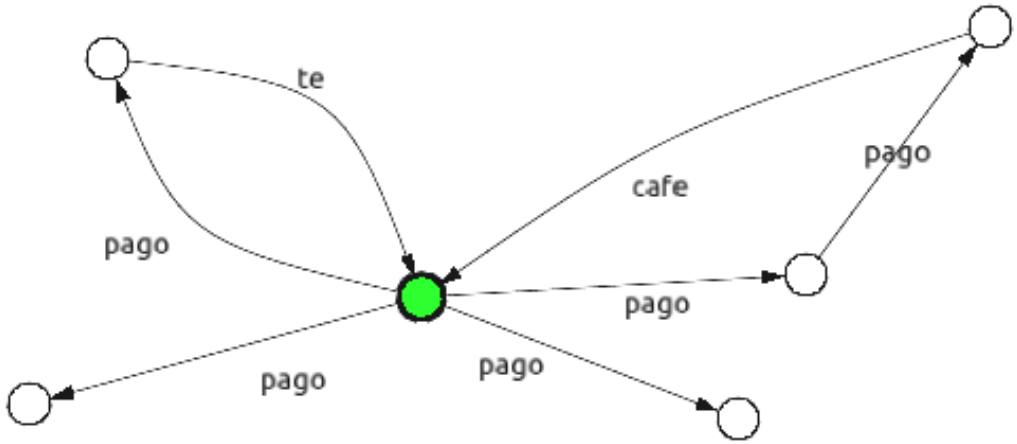
Composición paralela y comunicación

```
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));
```

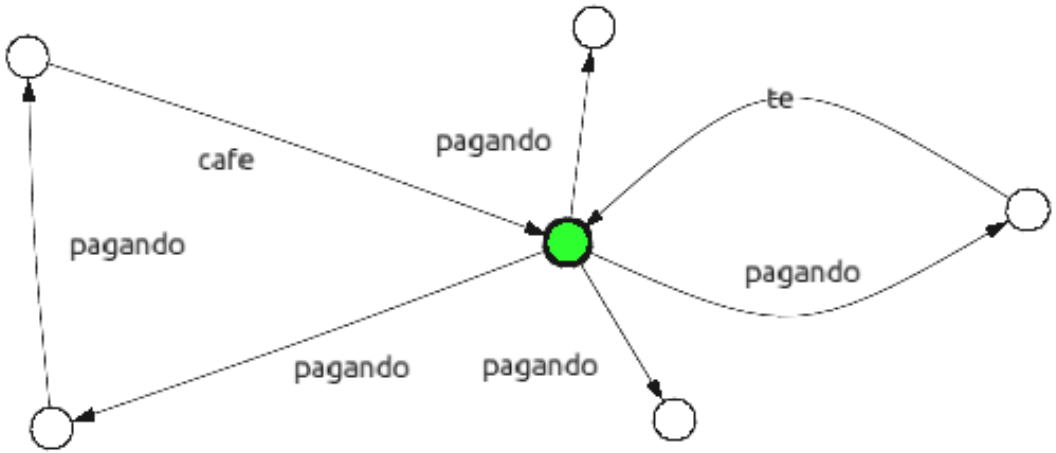


Operador bloquear

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

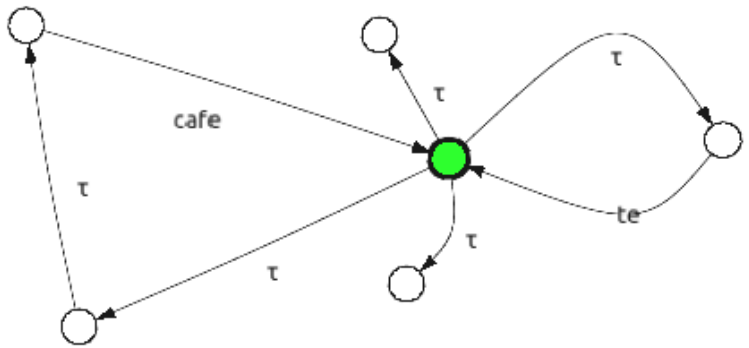
Operador rename

```
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
- Modelar el servicio de integración de actualización de clientes.