

Modelos de Comportamiento

¿Qué significa Comportamiento?

- Traducción del inglés - Behaviour
 - *...the aggregate of the responses or reactions or movements made by an organism in any situation...*
 - *...el conjunto de respuestas o reacciones o movimientos hechos por un organismo en cualquier situación...*
- En este caso el "organismo" puede ser
 - El sistema
 - La máquina
 - Un agente
 - Una entidad

Modelos Hasta Ahora

	Comportamiento	Semántica	Organismo
Jackson (I, G, R, D)	Si	Trazas, "Prunning"	Sistema
Contexto	No	Grafo, ...	n/a
Objetivos	Si	Trazas, "Prunning"	Sistema, agentes y/o maquina
Operaciones (Pre y Post, OCL)	Si	Pares de Estados, "Prunning"	Sistema, agentes y/o máquina
Operaciones (Casos de Uso)	Si	Secuencias de Estados, "Mixto"	Máquina
Conceptual (Clases y Objetos, OCL)	No	Grafo, "Prunning"	n/a

Pero entonces....????

- Ya vimos modelos que describen comportamiento!
- Llamamos modelos de comportamiento a una familia de notaciones que describen comportamiento pero utilizando una semántica "generativa" o "ejecutable"

Notaciones y lenguajes

- Basados en Interacciones
 - Foco es en describir cómo es la interacción entre varias entidades
 - Ej: Casos de uso, use case maps, sequence diagrams
 - Foco en ejemplificar interacciones importantes
 - Traza global - visión omnisciente
- Basado en Estados
 - Comportamiento completo de una entidad
 - ej: State charts, SDL, redes de petri, etc.
 - Visión local

Maquinas de Estado

- Un área con despelote terminológico.
- *Aquí Máquina de Estado* es un término genérico que denota una familia de notaciones que tienen
 - Estados, transiciones y etiquetas
 - Ejemplos:
 - Statecharts
 - FSM (Finite State Machine - Máquina de estados finitos)
 - LTS (Labelled Transition System - Sistema de Transiciones Etiquetadas)
 - Automatas (a la teoría de lenguajes)
 - Timed Automata
 -

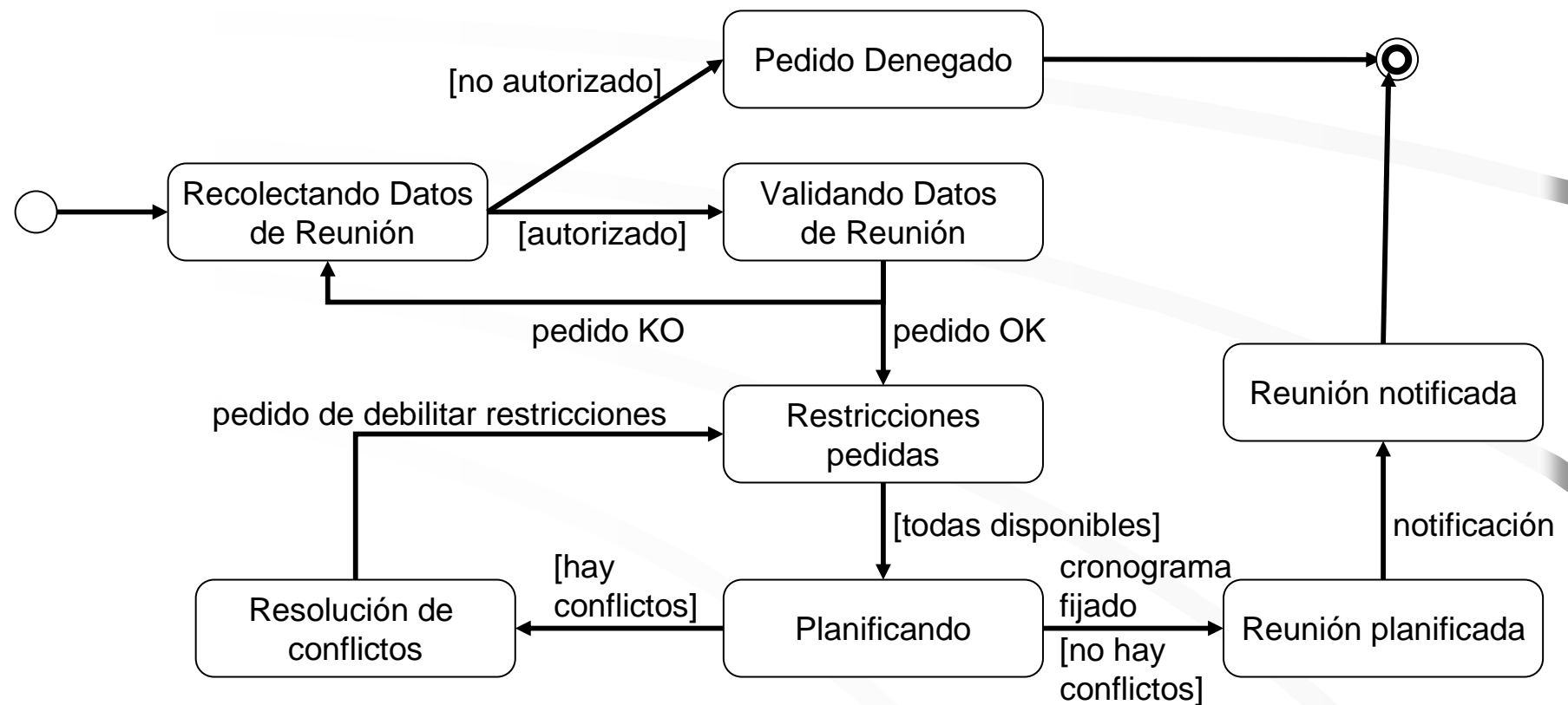
Maquinas de Estados

- Estructuran el estado del mundo con respecto a la relación "sigue a"
 - Estados (discretización del estado)
 - Transiciones (relación "sigue a")
 - Etiquetas (enriquecen noción de estado y transición)
- Permiten representar secuencias de estados de manera compacta
- Usos
 - Descripción del problema y de la solución
 - IR: Comportamiento de agentes y estado de entidades pasivas
 - Diseño: Protocolos, arquitecturas de software, estado de objetos y clases.

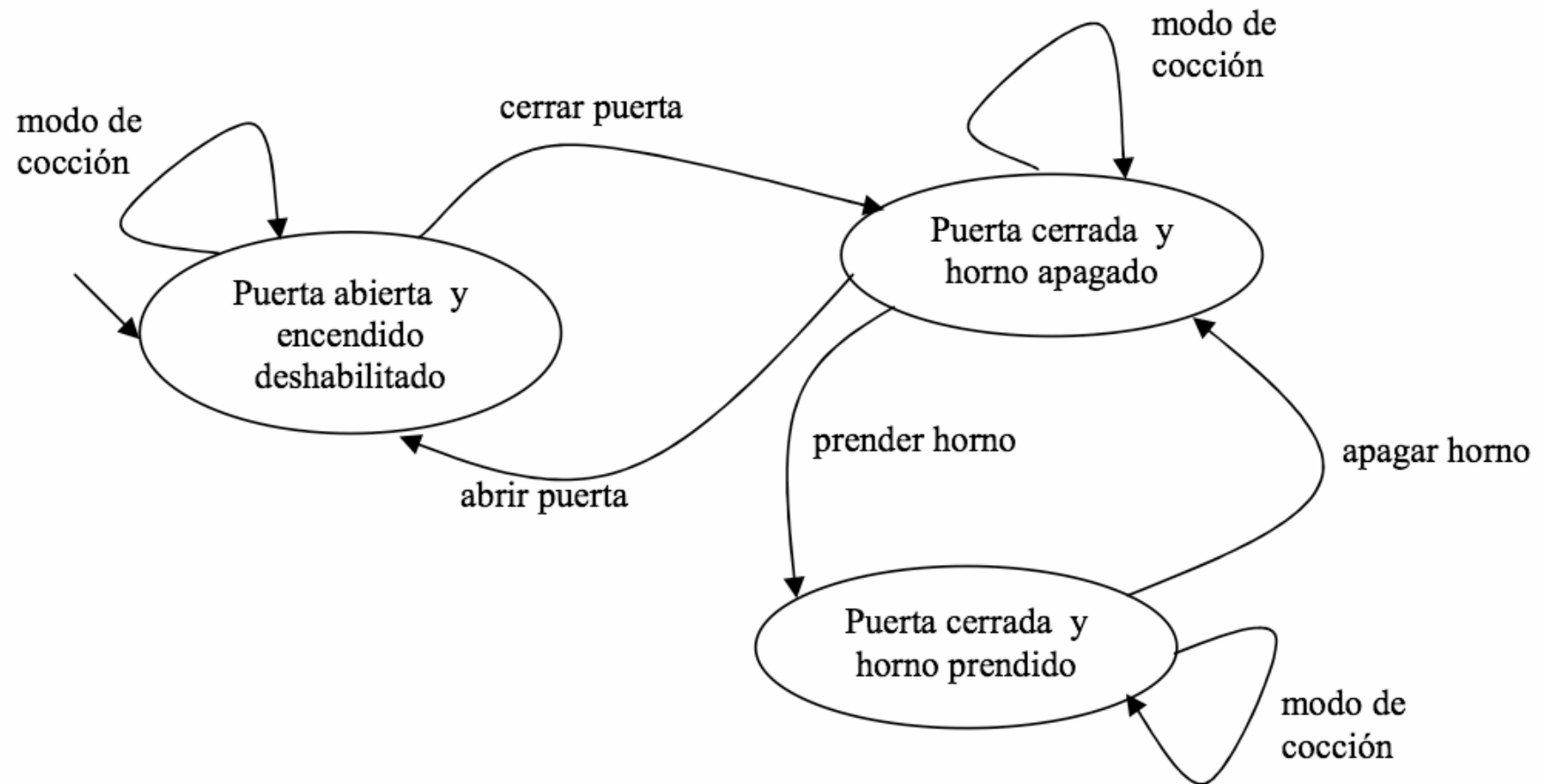
Lo que se viene

- Maquinas de Estado. Ejemplos
- LTS
 - Sintaxis Concreta (los diagramitas)
 - Sintaxis abstracta (def. matemática)
 - Composicion en paralelo
 - No determinismo
 - Semantica
 - Setting experimental
 - trazas vs bisimulacion
- Variantes
 - Input output
 - Variables (estado rico)
 - Temporizado
 - Variantes malas

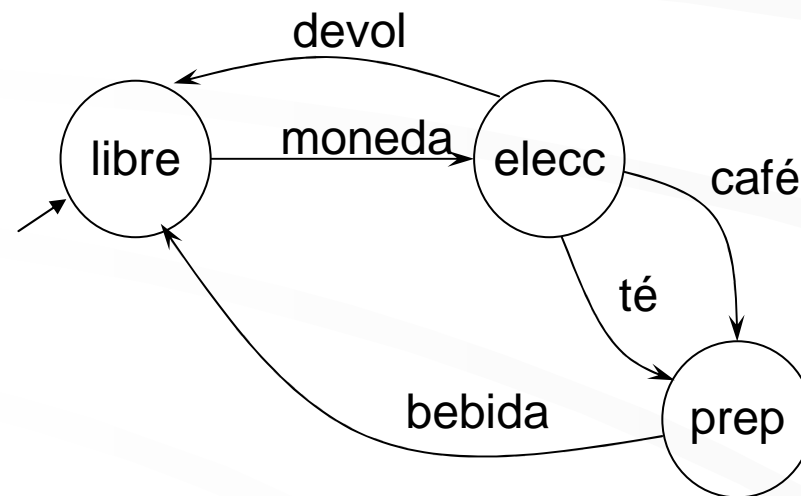
Maquinas de Estado: Ejemplo 1



Maquina de Estado: Ejemplo 2

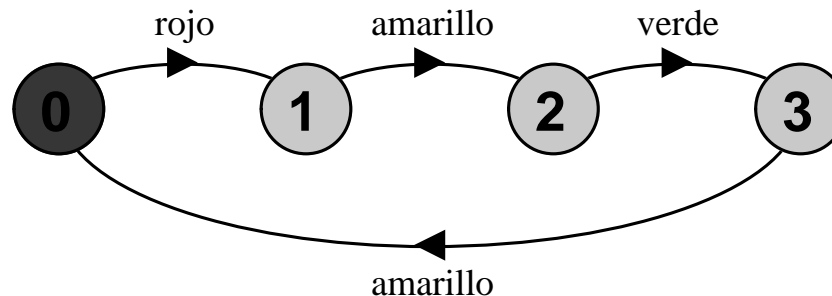


Maquina de Estados: Ejemplo 3



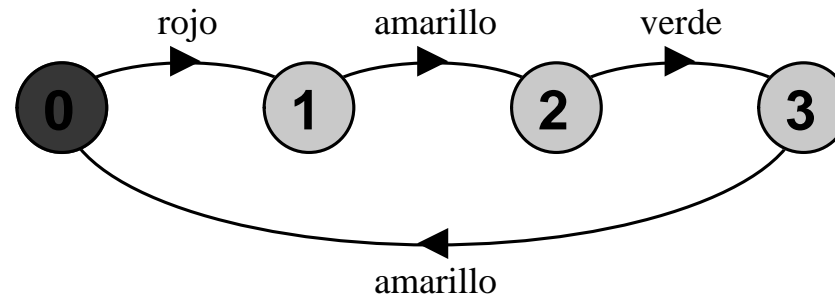
Maquinas de Estado: Ejemplo 4

- Esta variante la llamaremos LTS
 - LTS : Labelled Transition System
 - LTS : Sistema de Transiciones Etiquetadas
- Sintaxis Concreta
 - Círculos = Estados
 - Numero no tienen significado
 - Transiciones tienen etiquetas (con significado)
 - Estado rojo es el estado inicial



Sintaxis Abstracta de LTS

Def. 1 (LTS) Sea *Estados* el universo de estados, *Act* el universo de acciones observables, y $Act_\tau = Act \cup \{\tau\}$. Un LTS es una tupla $P = (S, L, \Delta, s_0)$, donde $S \subseteq \text{Estados}$ es un conjunto finito, $L \subseteq Act_\tau$ es un conjunto de etiquetas, $\Delta \subseteq (S \times L \times S)$ es un conjunto de transiciones etiquetadas, y $s_0 \in S$ es el estado inicial. Definimos el alfabeto de comunicacion de P como $\alpha P = L \setminus \{\tau\}$.



(tau es la acción *no observable* o *silenciosa*)

Semántica

- Trazas y Ejecuciones -

Definition 4.3 (Executions) Let $P = (S, L, \Delta, q)$ be a LTS. An execution of P is a sequence $w = q_0 l_0 q_1 l_1 \dots$ of states q_i and labels $l_i \in L$ such that $q_0 = q$ and $(q_i, l_i, q_{i+1}) \in \Delta$ for all $0 \leq i < |w/2|$.

Definition 4.4 (Projection) Let w be a word $w_0 w_1 w_2 w_3 \dots$ and A an alphabet. The projection of w onto A , which we denote $w|_A$, is the result of eliminating from the word w all elements w_i such that $w_i \notin A$.

Definition 4.5 (Traces) Let P be a LTS. A word w over the alphabet $\alpha(P)$ is a trace of P if there is an execution w' of P such that $w = w'|_{\alpha(P)}$. Note that traces do not include τ actions. We also define $tr(P) = \{w \mid w \text{ is a trace of } P\}$.

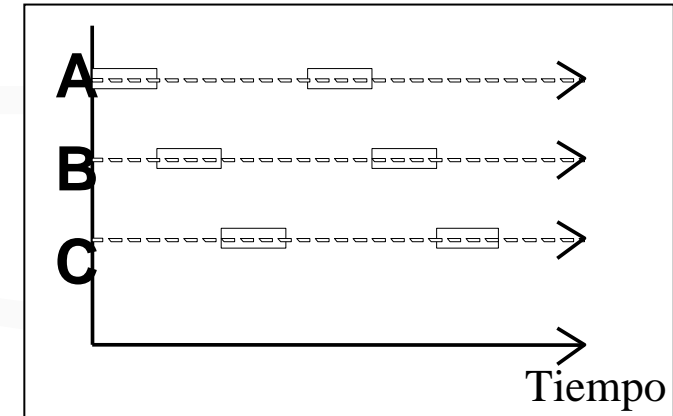
Definiciones

- **Concurrencia**

- Procesamiento *lógicamente* simultáneo.
- No implica múltiples unidades de procesamiento (UP). Si hay una sola UP, se necesita ejecución "interleaved".

- **Paralelismo**

- Procesamiento *físicamente* simultáneo.
- Involucra múltiples UPs.



Tanto concurrencia como paralelismo requieren acceso controlado a recursos comunes. Usaremos "paralelo" y "concurrente" de manera intercambiable.

Composición en Paralelo

- Semántica de Interleaving -

Transits es una relación entre LTS describiendo como un LTS P se convierte en P' al tomar una transición etiquetada por l .

Definition 4.7 (Transits) A LTS $P = (S, L, \Delta, q)$ transits with a label l into an LTS P' , denoted as $P \xrightarrow{l} P'$, if $P' = (S, L, \Delta, q')$ and $(q, l, q') \in \Delta$

Definition 4.8 (Parallel Composition of LTS) Let P_1 and P_2 be LTSs where $P_i = (S_i, L_i, \Delta_i, q_i)$. The parallel composition $P_1 \parallel P_2$ is an LTS $(S_1 \times S_2, L_1 \cup L_2, \Delta, (q_1, q_2))$ where Δ is the smallest relation that satisfies:

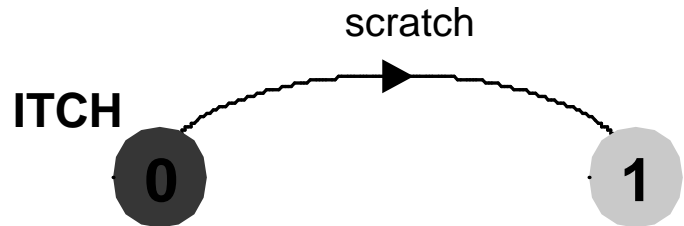
$$\frac{P_1 \xrightarrow{l} P'_1}{P_1 \parallel P_2 \xrightarrow{l} P'_1 \parallel P_2} \quad (l \notin \alpha(P_2))$$

$$\frac{P_2 \xrightarrow{l} P'_2}{P_1 \parallel P_2 \xrightarrow{l} P_1 \parallel P'_2} \quad (l \notin \alpha(P_1))$$

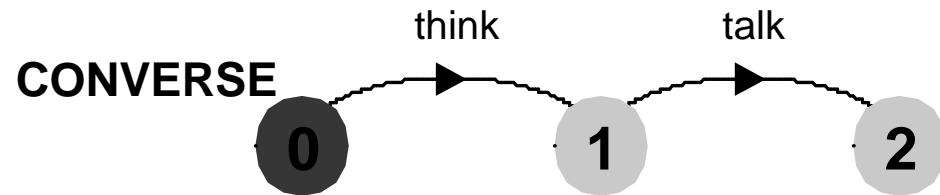
$$\frac{P_1 \xrightarrow{l} P'_1 \quad P_2 \xrightarrow{l} P'_2}{P_1 \parallel P_2 \xrightarrow{l} P'_1 \parallel P'_2} \quad (l \neq \tau)$$

Composición en paralelo

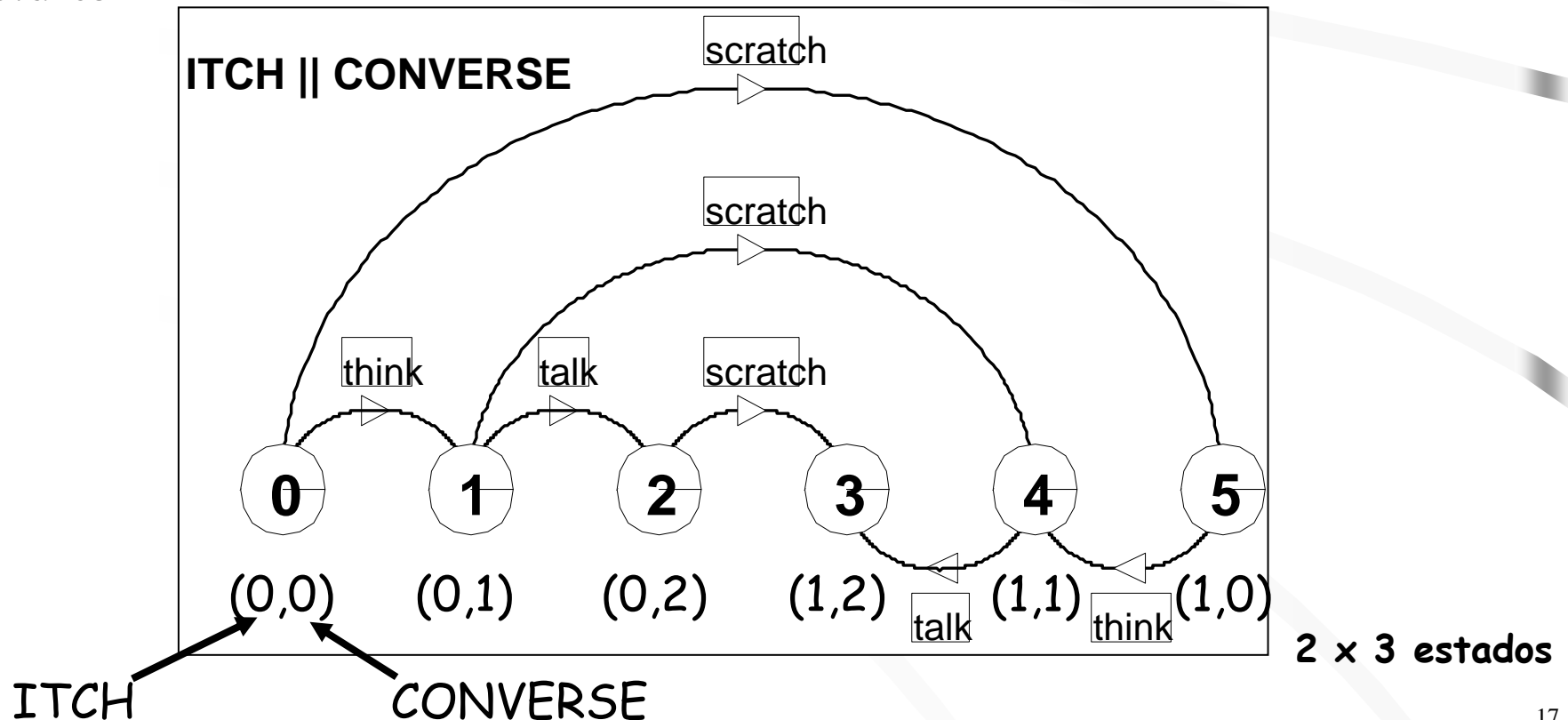
- Interleaving de acciones -



2 estados

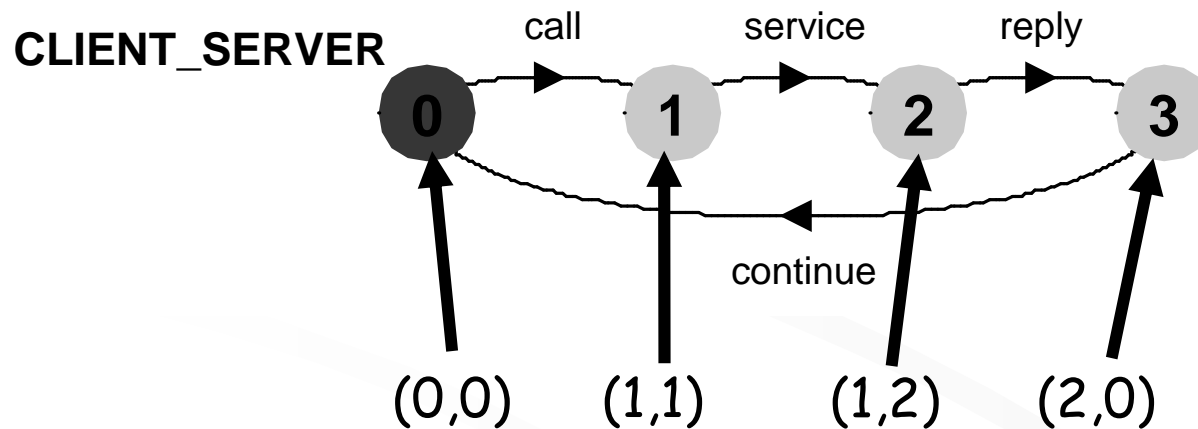
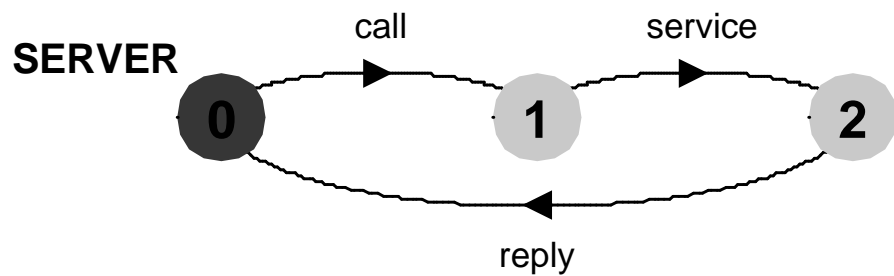
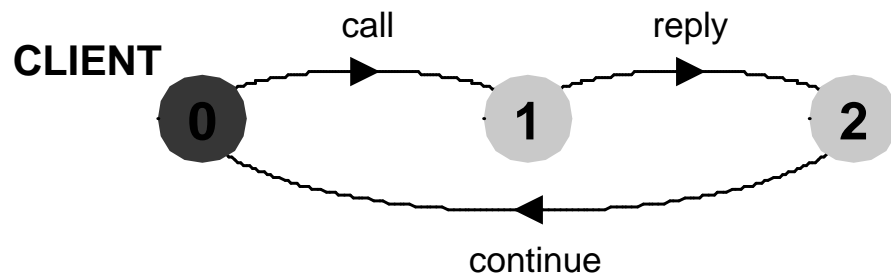


3 estados



Composición en paralelo

- Sincronización por Acciones Compartidas -



¿Qué pasó con los pares (0,1), (0,2), (1,0), (2,1), (2,2)?

Composición en paralelo

- Sincronización por Acciones Compartidas -

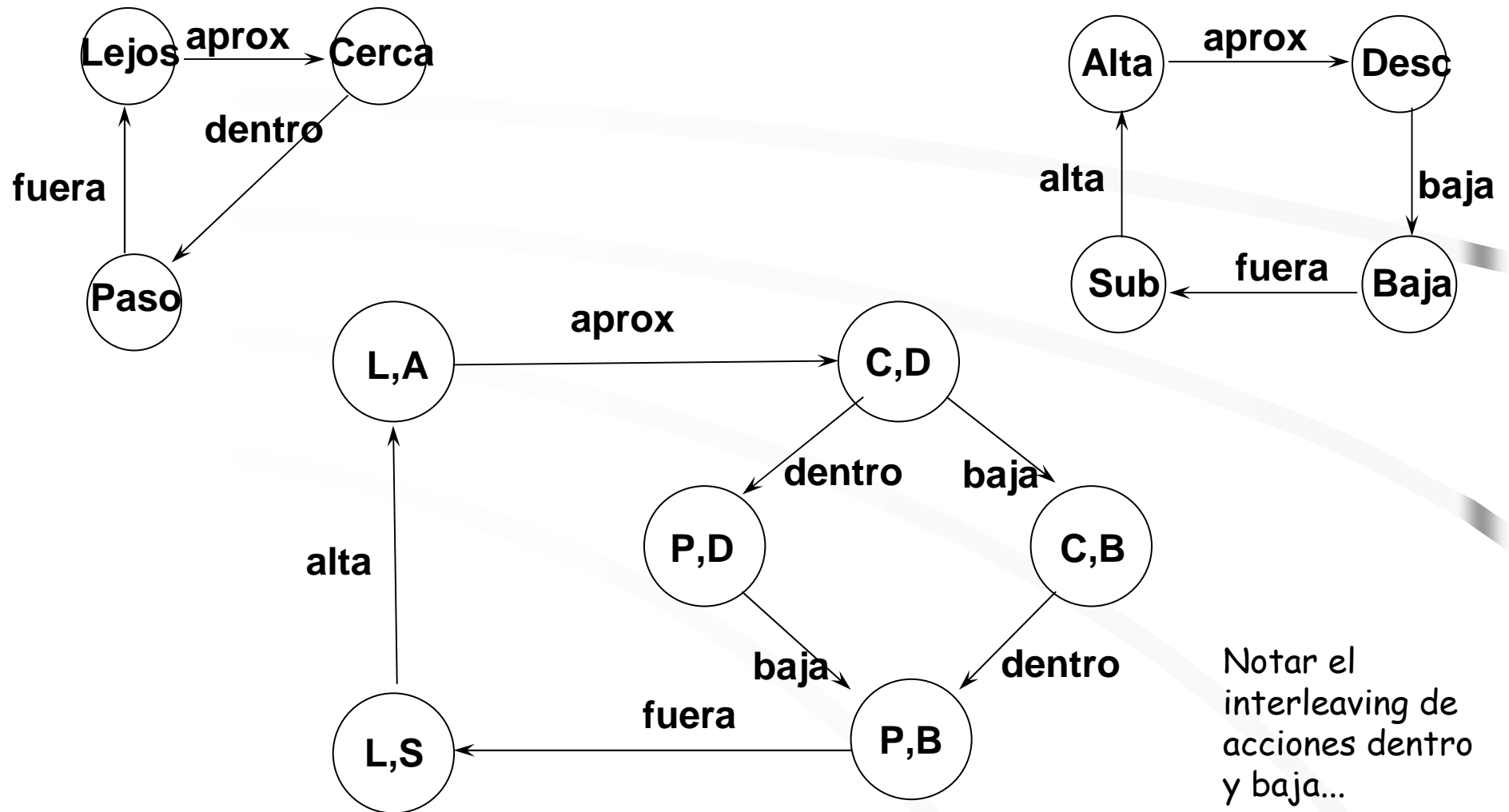
Si dos procesos en una composición tienen acciones en común, se dice que acciones son *compartidas*. Acciones compartidas son la manera en que se modela interacción de procesos.

Mientras se hace interleaving de acciones no compartidas, una acción común debe ser ejecutada al mismo tiempo por todos los procesos que participan en esa acción compartida.

Como vimos en el ejemplo anterior, las interacciones restringen el comportamiento global.

Otro Ejemplo de Composición en Paralelo

Ignorar etiquetas en estados



Modelado de Concurrency

- ¿Cómo modelamos la velocidad de ejecución?
 - Asumimos que las máquinas pueden funcionar a velocidades arbitrarias (nos abstraemos del tiempo)
- ¿Cómo modelamos concurrencia?
 - Orden arbitrario de acciones de distintos (**interleaving** de acciones pero preservación de acciones de cada proceso)
- El resultado?
 - Un modelo general, independiente de scheduling (modelo de ejecución **asíncrono**)

Composición en paralelo

- Leyes algebraicas -

Conmutatividad: $(P \parallel Q) = (Q \parallel P)$

Asociatividad: $(P \parallel (Q \parallel R)) = ((P \parallel Q) \parallel R)$
 $= (P \parallel Q \parallel R).$

¿Reflexividad: $(P \parallel P) = P$?

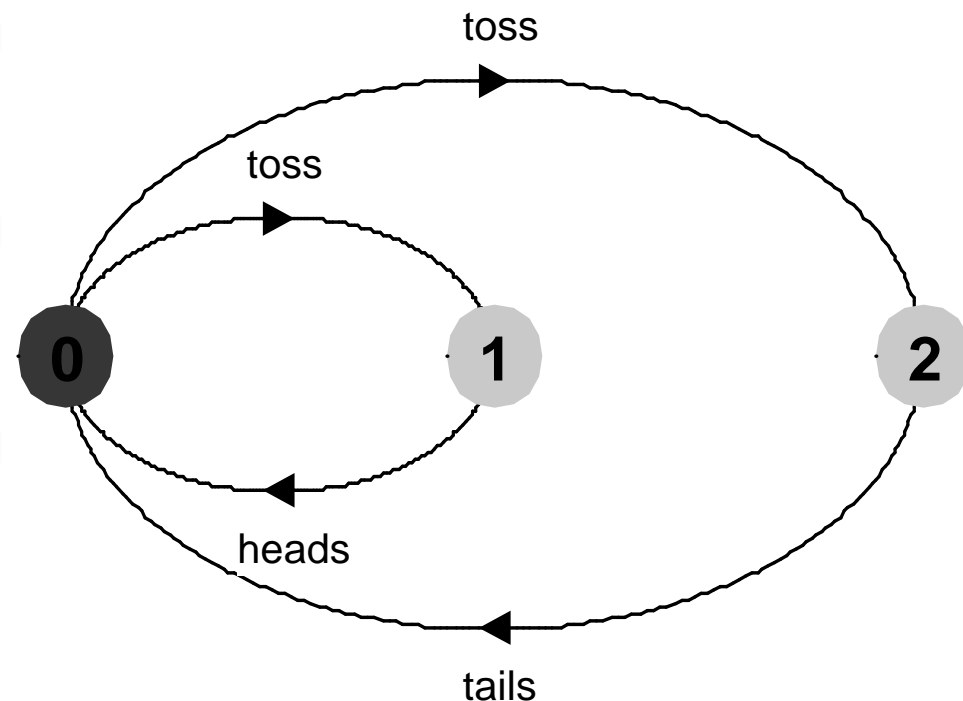
Elección no-determinística

Un LTS no-determinístico tiene al menos un estado con más de una transición etiquetada con la misma acción.

Tirar una moneda...

Trazas posibles?

¿Existe una versión determinística con mismas trazas?

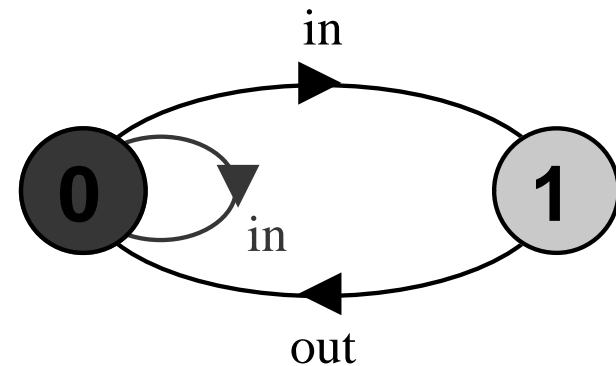


No determinismo, para qué?

No determinismo permite abstraer los detalles involucrados en la toma de una decisión en una alternativa.

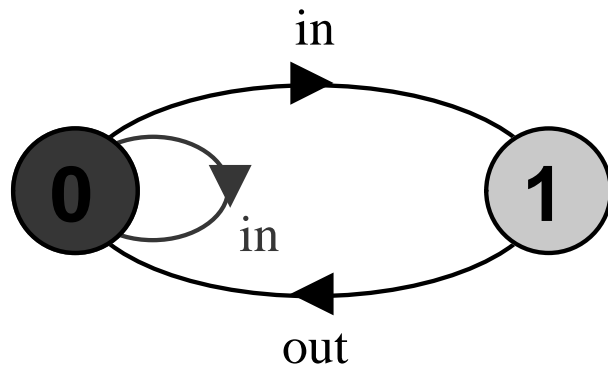
Un canal de comunicación que falla...

¿Existe una versión determinística con mismas trazas?



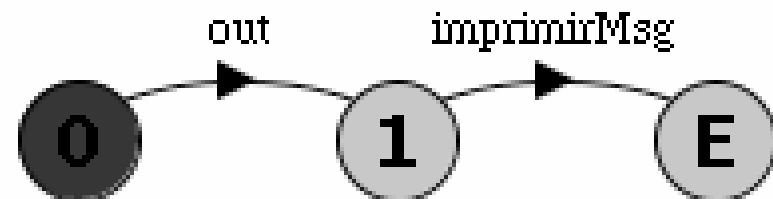
Composición en Paralelo

- No Determinismo -



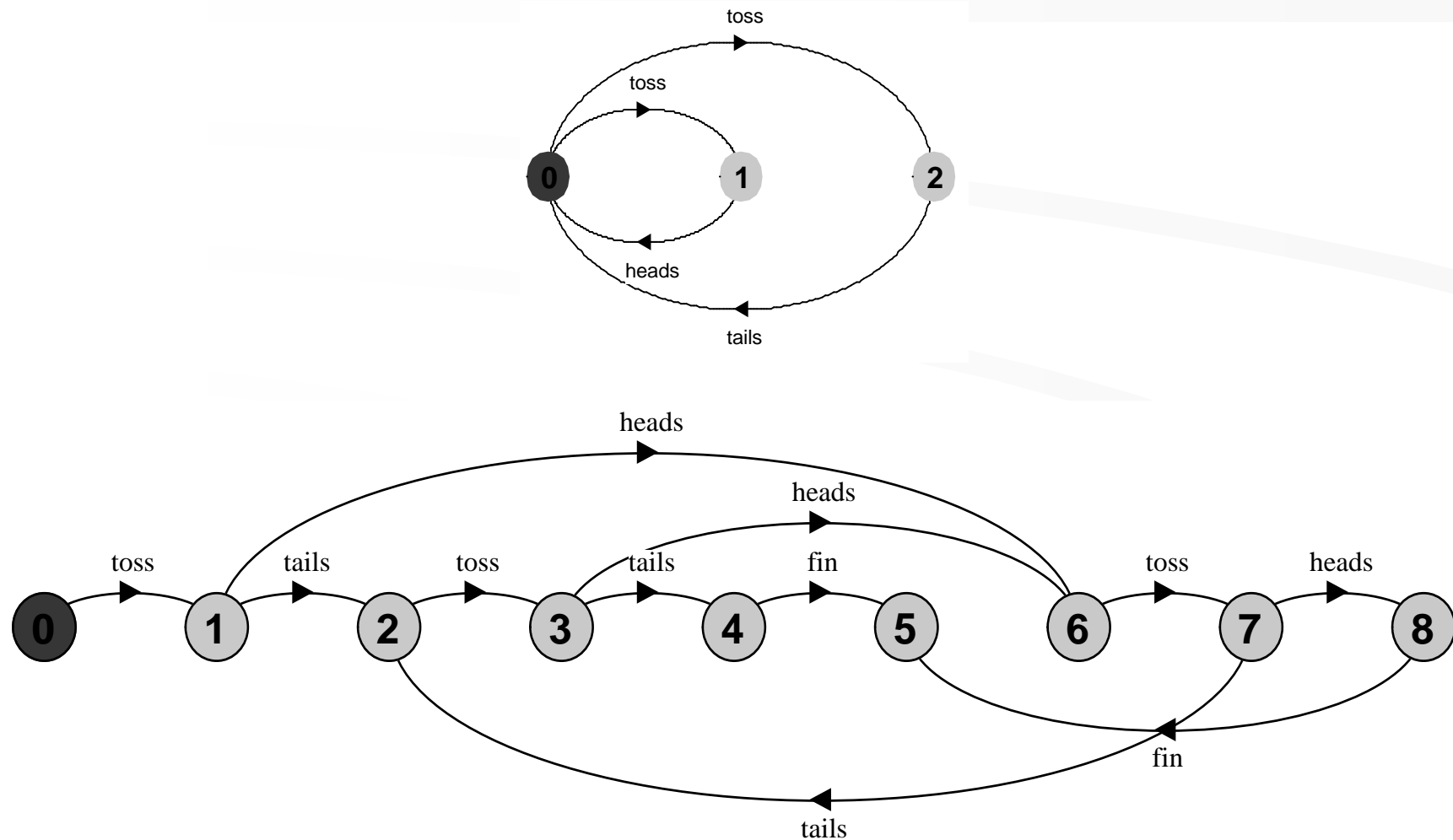
¿Qué pinta tiene la composición?

... y si usamos un canal determinístico?

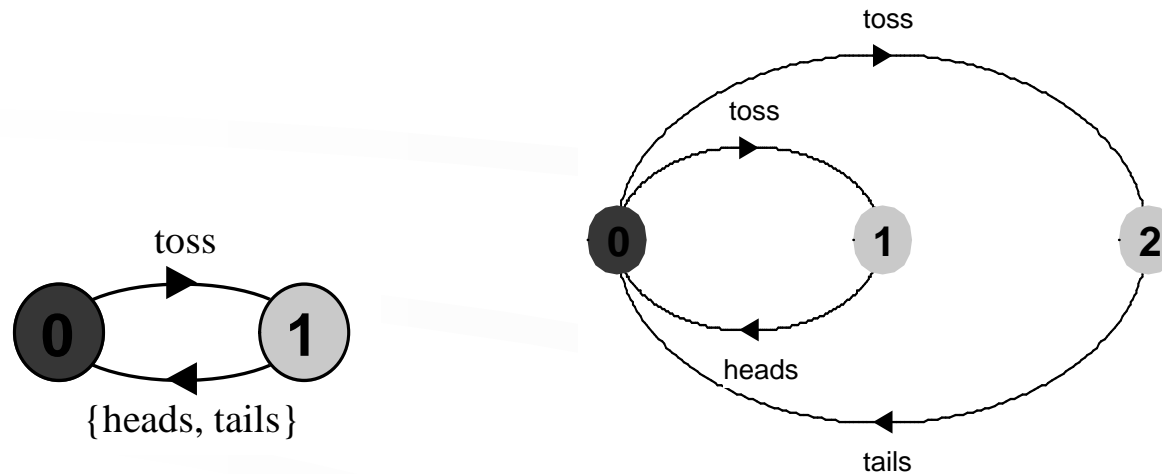


¿Quién controla la elección no determinística?

Otro Ejemplo de No Determinismo



Corolario acerca de Equivalencia de LTS



Estos modelos deben ser considerados no equivalentes

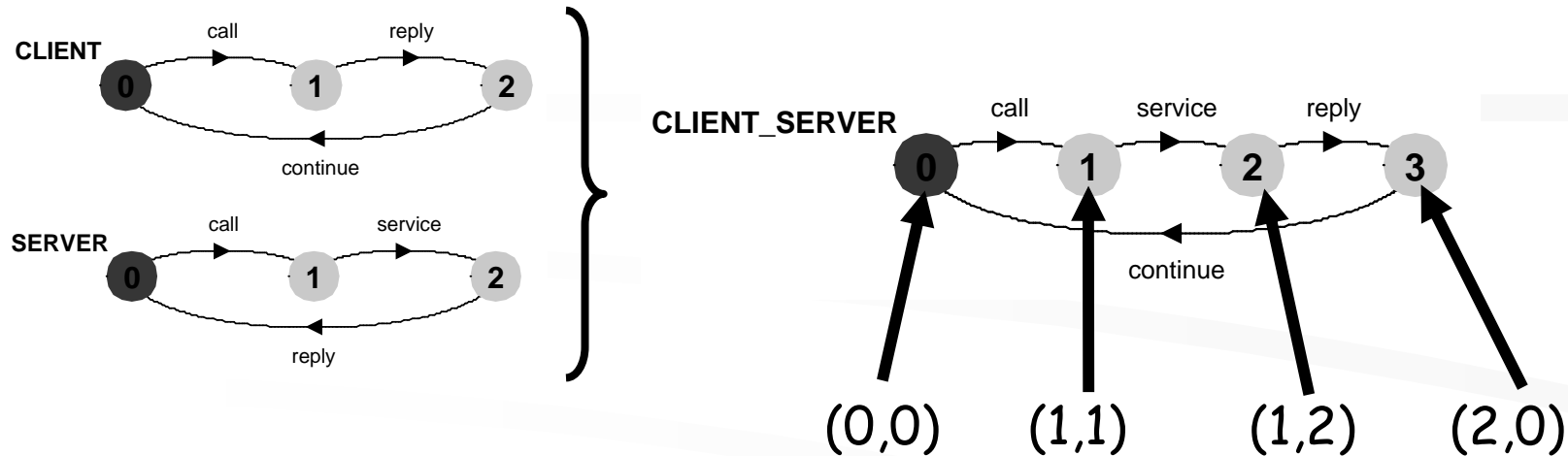
Es decir que igualdad de trazas no criterio válido...

¿Cuál es la Semántica de LTS?

¿Igualdad de Trazas?
¿Isomorfismo?

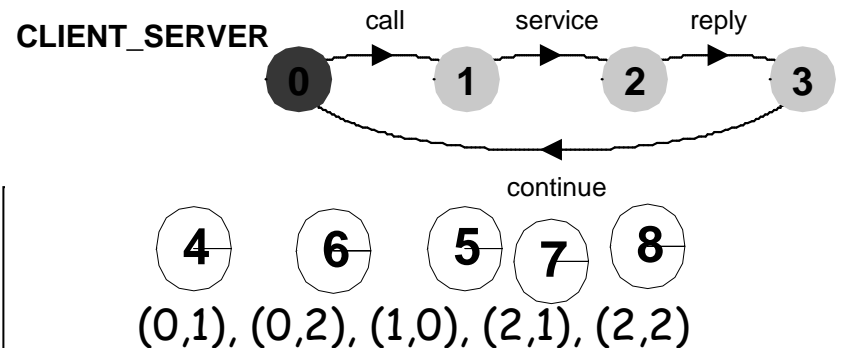
Equivalencia de LTS

- Estados Alcanzables -



¿Qué pasó con los pares (0,1), (0,2), (1,0), (2,1), (2,2)?

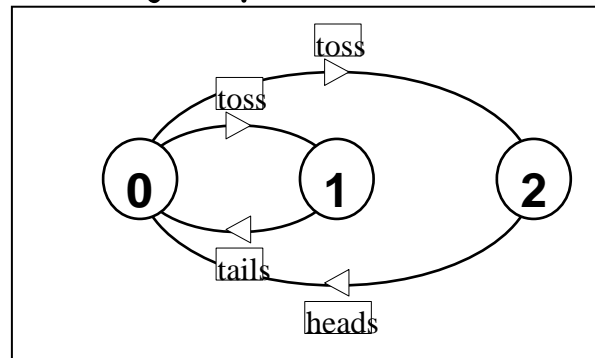
¿Deberían ser equivalentes?



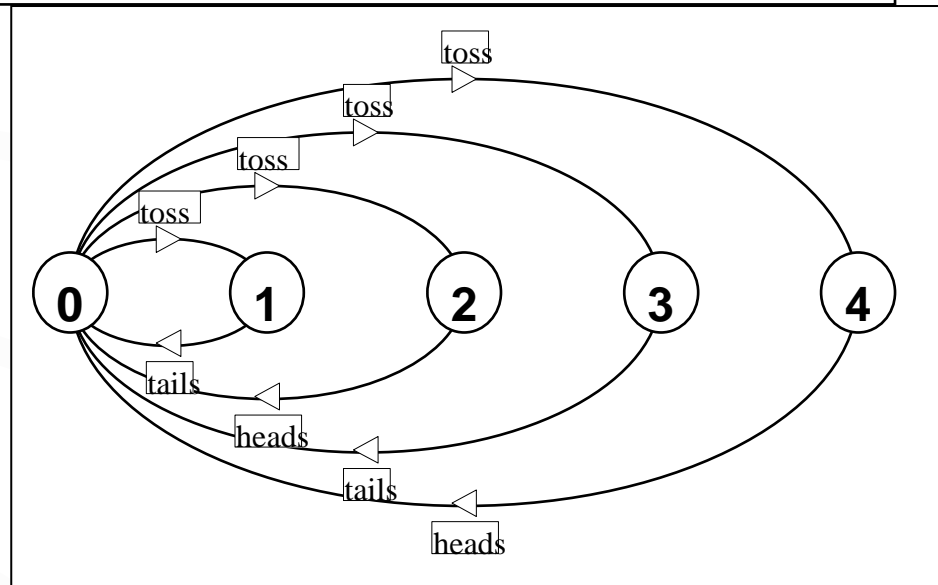
Propuesta Fallida 3

"Dos LTS son equivalentes si y solo si al quitar sus estados *no alcanzables* son isomorfos"

Contraejemplo:



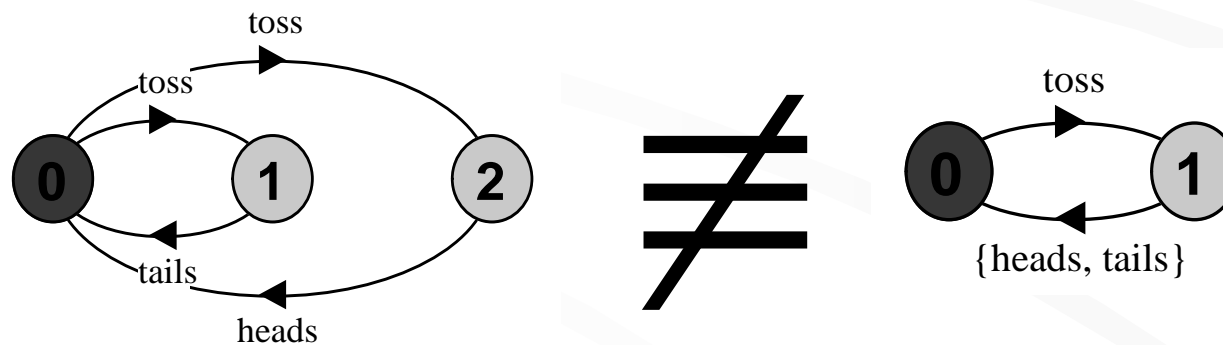
≡



Propuesta Fallida 4

"Dos LTS son equivalentes si definen el mismo mismo conjunto de trazas"

Contraejemplo:



Equivalencia por trazas es una equivalencia demasiado *gruesa (coarse)* o débil (*weak*)

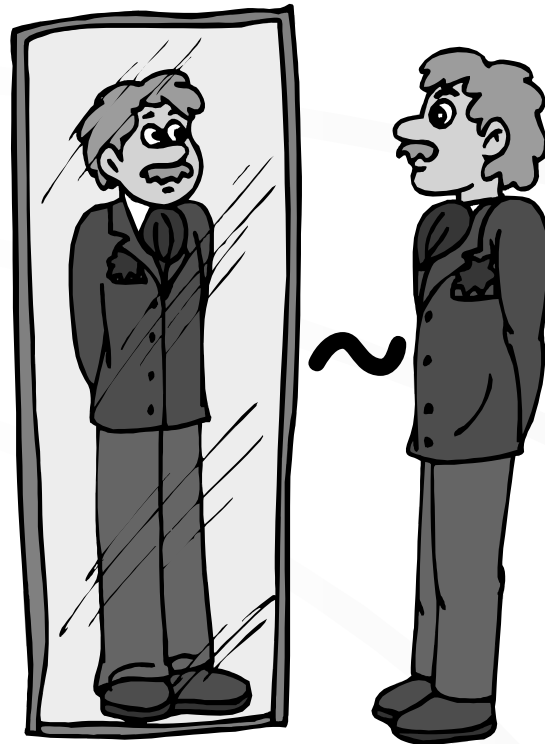
Que es lo que buscamos?

Una relación entre LTS que:

- sea una equivalencia
 - Reflexivo ($P \equiv P$)
 - Transitivo ($P \equiv Q$ y $Q \equiv R$ implica $P \equiv R$)
 - Simétrico ($P \equiv Q$ implica $Q \equiv P$)
- sea abstracta con respecto
 - Número de estados del modelo
 - Estructura del modelo
 - Trazas del modelo
- coincida con alguna noción empírica - experimento de caja negra?
- "encaje" con nuestro lenguaje de especificación
 - e.j. si $P \equiv Q$ entonces $P \parallel R \equiv Q \parallel R$

Equivalencias de LTS

Bisimulaciones



Bisimulación Fuerte

Definition 1 (Strong Bisimulation) *Sea \mathcal{P} el universo de todos los LTS. Una relacion binaria $R \subseteq \mathcal{P} \times \mathcal{P}$ es una bisimulacion fuerte si y solo si cuando $(P, Q) \in R$ entonces para cada accion $a \in \text{Act}$:*

- $(P \xrightarrow{a} P') \Rightarrow (\exists Q' . Q \xrightarrow{a} Q' \wedge (P', Q') \in R)$
- $(Q \xrightarrow{a} Q') \Rightarrow (\exists P' . P \xrightarrow{a} P' \wedge (P', Q') \in R)$

Definition 2 (Strong Bisimilarity) *Dos LTS $P, Q \in \mathcal{P}$ son fuertemente bisimilares ($P \sim Q$) si y solo si existe una bisimulacion fuerte R tal que $(P, Q) \in R$.*

$$\sim = \bigcup \{ R \mid R \text{ es una bisimulacion fuerte} \}$$

Propiedades de \sim

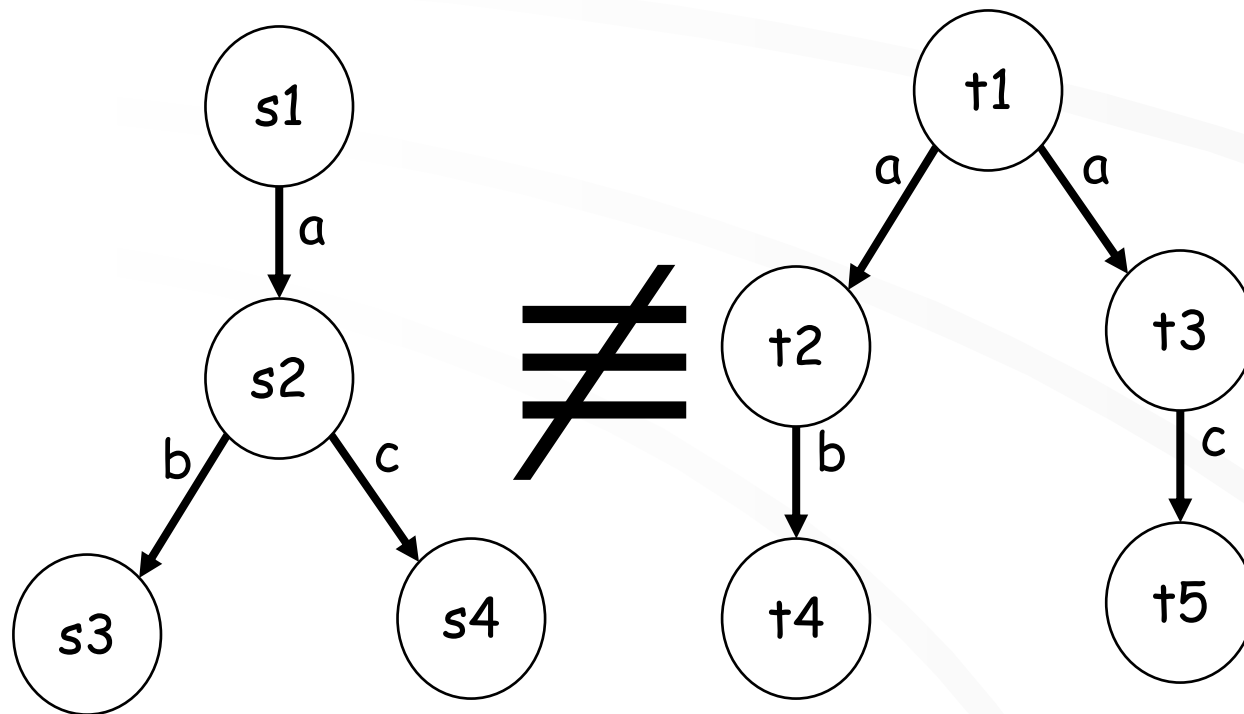
- \sim es una bisimulación fuerte
- \sim es una relación de equivalencia
- El siguiente teorema permite usar una definición alternativa de \sim

Teorema 1 $(P \sim Q)$ si y solo si para cada acción $a \in \text{Act}$:

- $(P \xrightarrow{a} P') \Rightarrow (\exists Q' . Q \xrightarrow{a} Q' \wedge P' \sim Q')$
- $(Q \xrightarrow{a} Q') \Rightarrow (\exists P' . P \xrightarrow{a} P' \wedge P' \sim Q')$

Como demostrar no-bisimilaridad

1. Enumerar todas las relaciones binarias y mostrar que ninguna contiene $(s1 \sim t2)^*$ y es una bisimulación fuerte (muy costoso),
2. Realizar algunas observaciones que nos permiten eliminar relaciones candidatas a bisimulación en un paso. Es decir usar un juego que caracteriza bisimilaridad fuerte.



* Abuso de notación

El Gran Juego de la Bisimulación Fuerte

- Sean P y Q dos LTS
- El juego consiste en un atacante y un defensor que empiezan en P y Q respectivamente.
 - El juego se juega en rondas y con configuraciones del juego que son pares de $\wp \times \wp$.
 - En cada ronda, existe una configuración corriente. Inicialmente, la configuración (P, Q) es la corriente.

Intuición

El defensor quiere mostrar que P y Q son bisimilares mientras que el atacante quiere probar lo contrario.

Reglas del Juego de Bisimulación

- En cada ronda, los jugadores cambian la configuración de la siguiente manera:
 1. El atacante elige uno de los procesos en la configuración actual y hace una movida por alguna transición con etiqueta $a \in A$
 2. El defensor deber responder haciendo una movida por una transición etiquetada con a en el otro proceso.
- I La configuración resultante se transforma en la configuración corriente y el juego continua con otra ronda.

Resultado del Juego

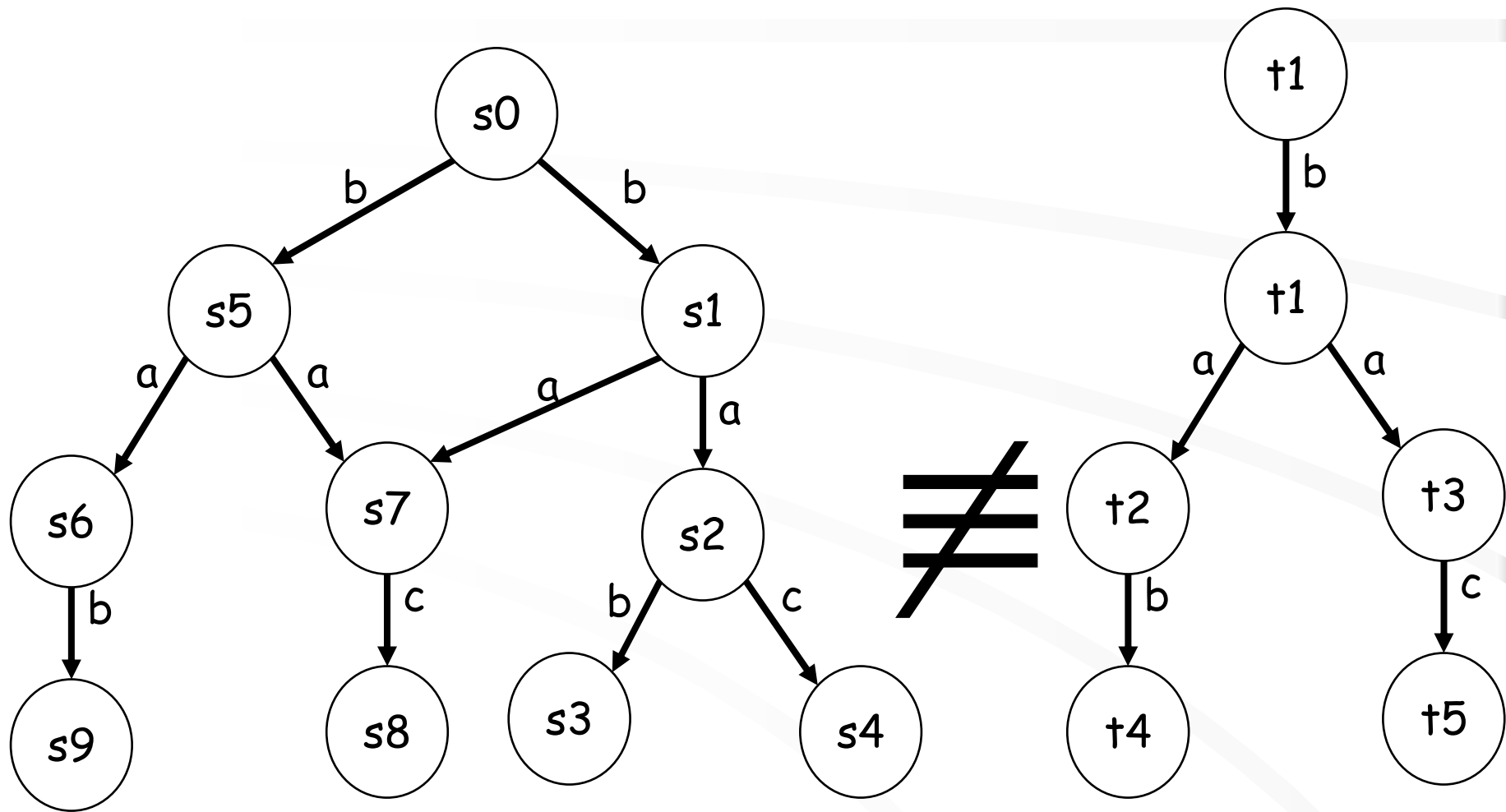
- Si un jugador no puede mover, el otro gana
- Si el juego es infinito, el defensor gana

Caracterización de Bisimilaridad Fuerte vía el Juego

- P y Q son fuertemente bisimilares si y solo si el defensor tiene una estrategia ganadora universal empezando desde la configuración (P, Q)
- P y Q no son fuertemente bisimilares si y solo si el atacante tiene una estrategia ganadora universal empezando desde la configuración (P, Q)

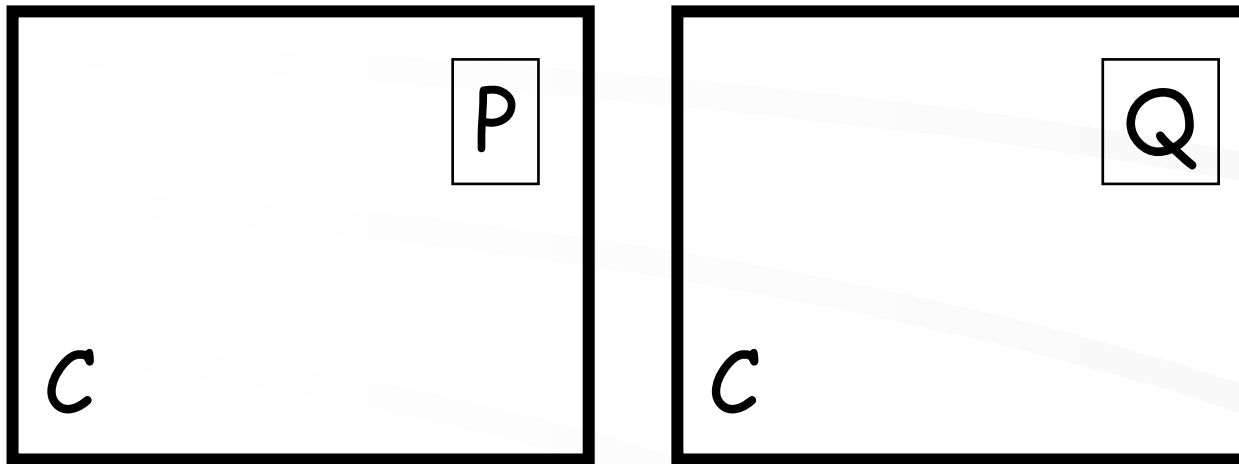
El juego provee un argumento elegante para el caso negativo. Para el caso positivo es más fácil simplemente dar una relación de bisimulación

Cuál es la estrategia ganadora para el atacante?



Congruencia

Dado un contexto para P , querríamos poder cambiar P por un proceso equivalente sin alterar el sistema



Una equivalencia es una congruencia si y solo si $P \equiv Q$ implica que $C(P) \equiv C(Q)$

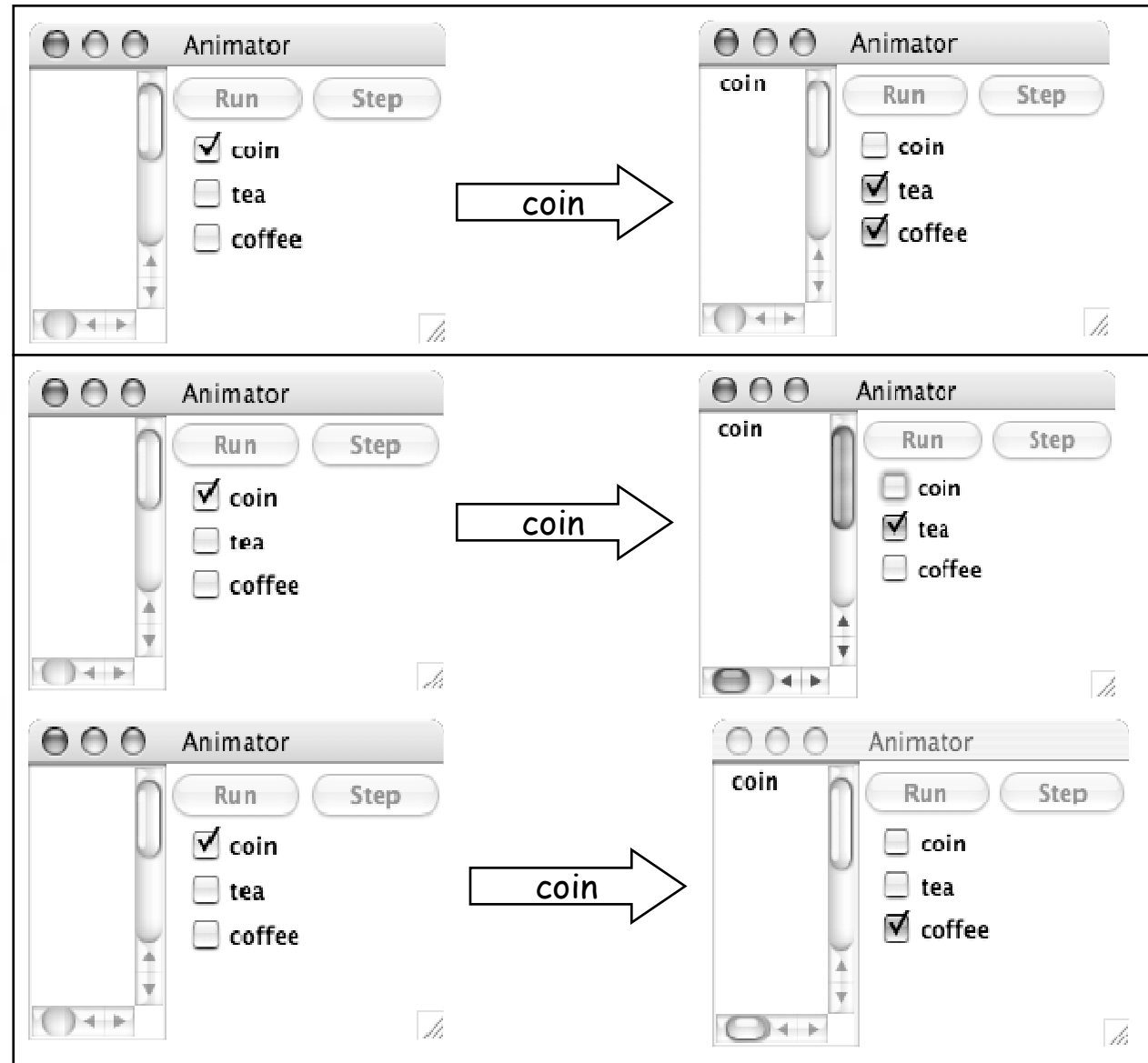
Ejemplo de contexto : $C(x) = (R \parallel x)$
¿Es verdad que si $P \sim Q \Rightarrow C(P) \sim C(Q)$?
es decir, vale que $P \sim Q \Rightarrow (R \parallel P) \sim (R \parallel Q)$?

Experimentos de Caja Negra

Experimento
sobre
proceso A

Dos procesos son
equivalentes si un
observador
externo no los
puede distinguir

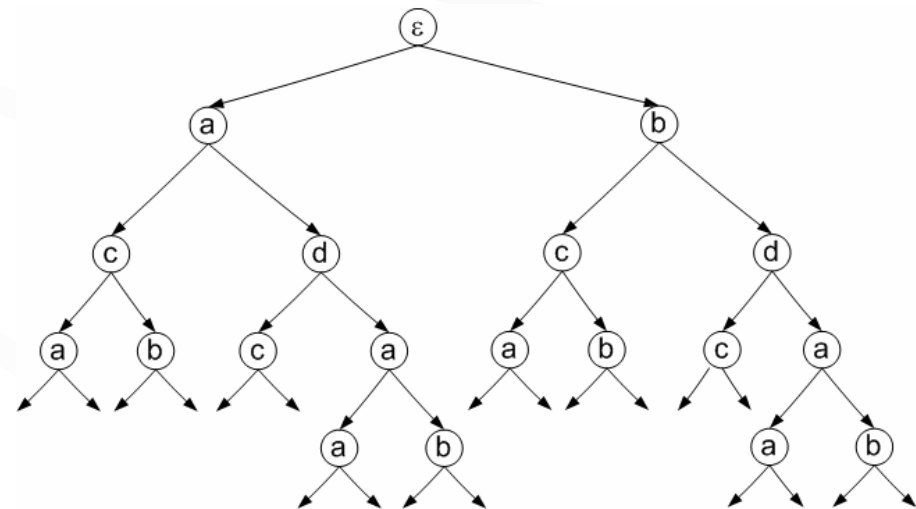
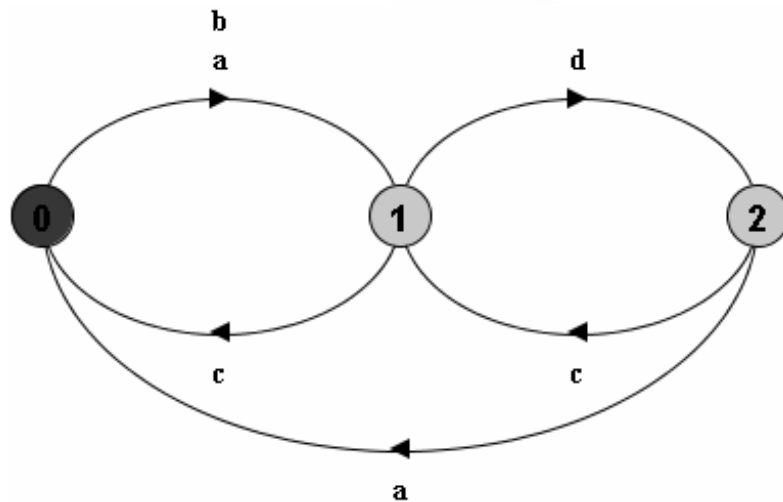
Experimentos
sobre
proceso B



Semántica de LTS

- Lineal vs Arbórea -

- Equivalencia por trazas induce una semántica lineal
 - Un LTS está caracterizado por un conjunto de secuencias
- Equivalencia por bisimulación induce una semántica arbórea
 - Un LTS está caracterizado por el árbol de ejecución resultante de desdoblar sus ciclos



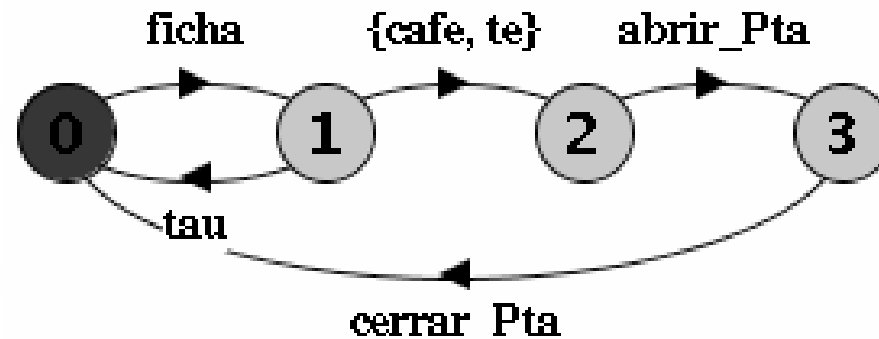
Semántica de LTS

- Lineal vs Arbórea -

- Para algunas aplicaciones semántica lineal es adecuada
 - Nos resulta intuitivo dar propiedades para trazas:
 - La ambulancia deberá arribar no mas de 14 minutos después del incidente
 - El usuario deberá ingresar su PIN correctamente antes de poder extraer efectivo
- Para otras aplicaciones una semántica arbórea es necesaria
 - Composición en paralelo
 - Casos de Uso: "Extrayendo efectivo" con precondition "PIN correcto ingresado"
 - Propiedades "Branching"
 - En cualquier punto del procedimiento de compra, el usuario podrá consultar su carrito y hacer un check out
- Existen lógicas temporales lineales y branching que permiten predicar sobre conjuntos de trazas y estructuras arbóreas
 - (mas de esto al final de la materia si tenemos tiempo)

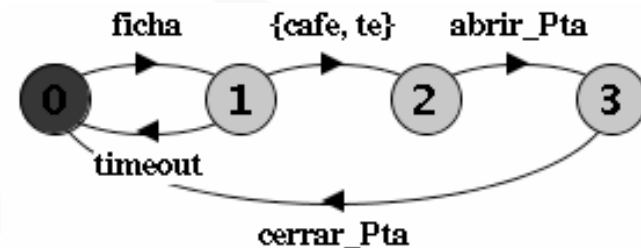
¿Qué pasa con Tau?

LTS vienen equipados con una etiqueta especial, tau, la acción no observable.



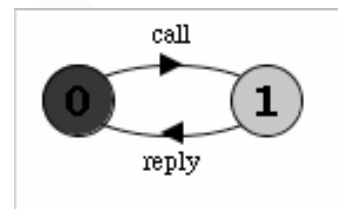
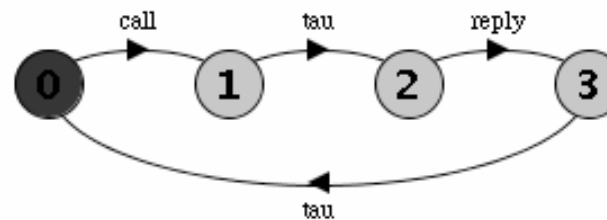
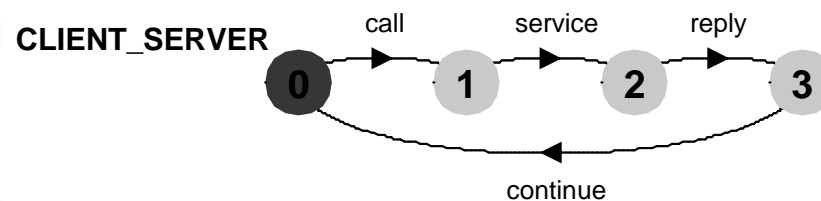
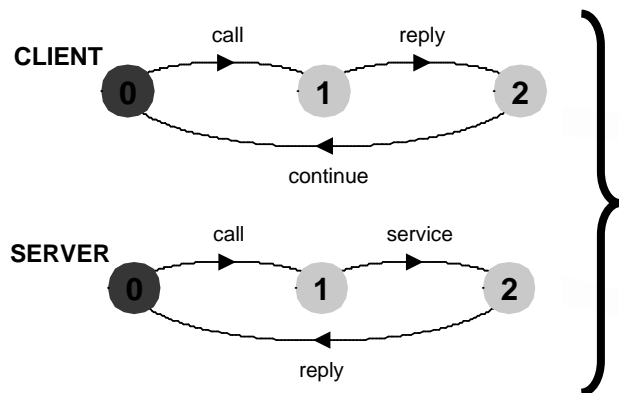
La tau modela una acción de timeout interna

Cual es la implicancia práctica de modelar la expendedora así:



¿Para qué sirve Tau?

- Para impedir sincronización externa (ver ejemplo anterior)
- Para reducir complejidad vía abstracción :

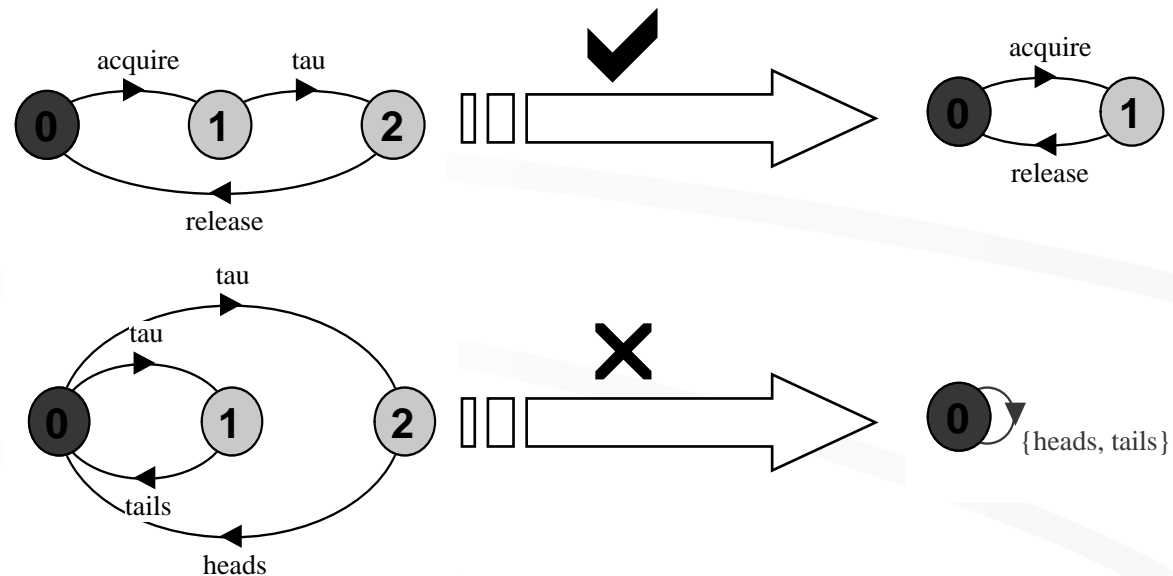


Ocultamiento
de service y
continue

Minimización
preservando
equivalencia

Bisimulación Considerando Tau

- Hay que "eliminar" las transiciones tau. Pero cómo?



- Definimos transiciones observacionales o débiles...

$$\Rightarrow_a = \begin{cases} (-\xrightarrow{\tau})^* \circ \xrightarrow{a} \circ (-\xrightarrow{\tau})^* & \text{if } a \neq \tau \\ (-\xrightarrow{\tau})^* & \text{if } a = \tau \end{cases}$$

Bisimilaridad Débil (Weak Bisimilarity)

Definición 3 (Weak Bisimulation) Sea \mathcal{P} el universo de todos los LTS. Una relación binaria $R \subseteq \mathcal{P} \times \mathcal{P}$ es una bisimulación débil si y solo si cuando $(P, Q) \in R$ entonces para cada acción $a \in \text{Act} \cup \{\tau\}$:

- $(P \xrightarrow{a} P') \Rightarrow (\exists Q' \cdot Q \xRightarrow{a} Q' \wedge (P', Q') \in R)$
- $(Q \xrightarrow{a} Q') \Rightarrow (\exists P' \cdot P \xRightarrow{a} P' \wedge (P', Q') \in R)$

Definición 4 (Weak Bisimilarity) Dos LTS $P, Q \in \mathcal{P}$ son débilmente bisimilares ($P \approx Q$) si y solo si existe una bisimulación débil R tal que $(P, Q) \in R$.

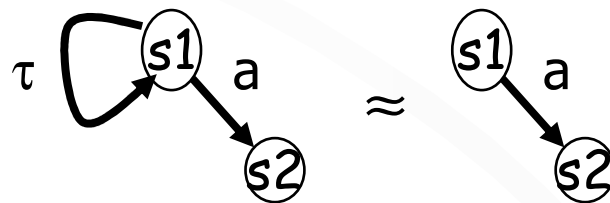
$$\approx = \bigcup \{ R \mid R \text{ es una bisimulación débil} \}$$

Intuición:

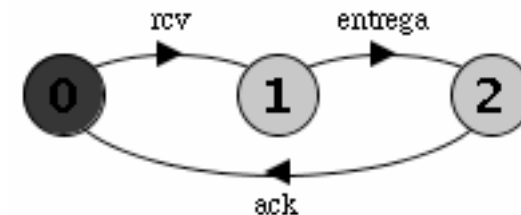
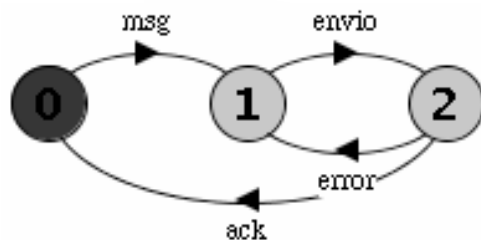
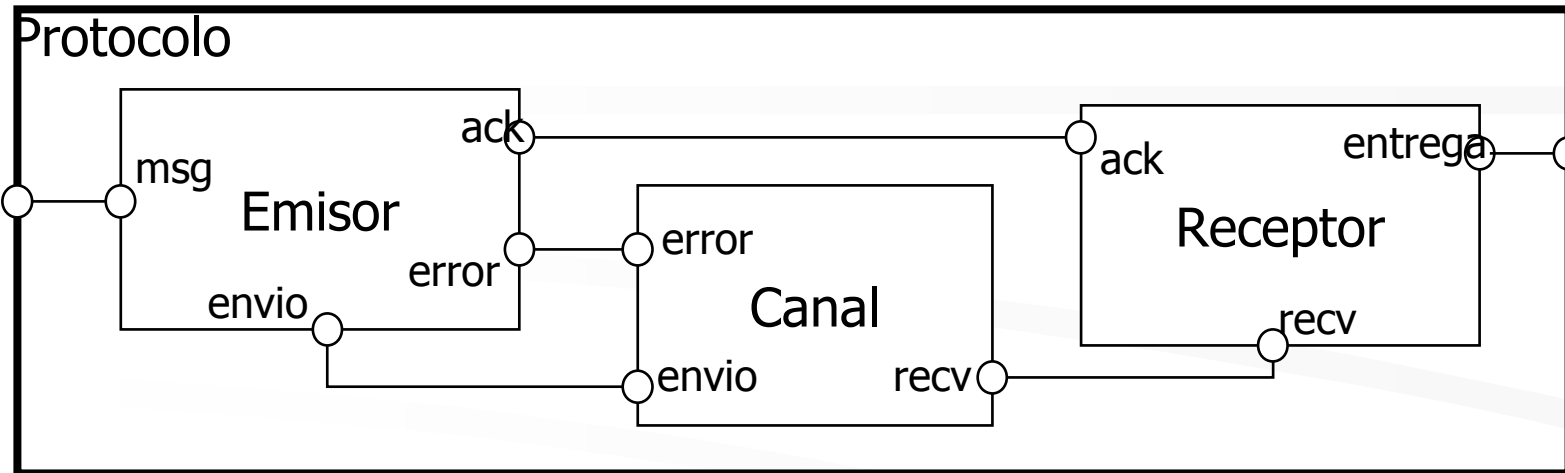
- Si P hace "a", entonces Q lo puede imitar haciendo algunos tau, un "a" y mas tau
- Si Q hace

Propiedades de \approx

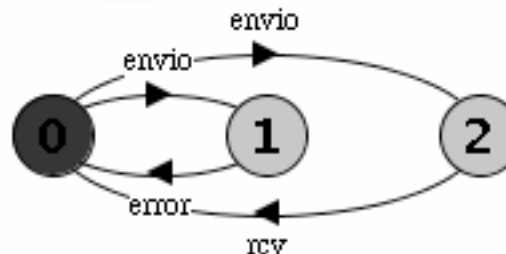
- Es una bisimulación débil
- Es una relación de equivalencia
- Tiene un juego que la caracteriza
 - Igual que antes pero el defensor puede hacer jugadas \Rightarrow
- Bisimilaridad fuerte implica bisimilaridad débil ($\sim \subseteq \approx$)
- Es una congruencia con respecto a \parallel
- Abstrae de loops sobre tau (Eliminación de ejecuciones divergentes)



Verificación - Un protocolo trivial

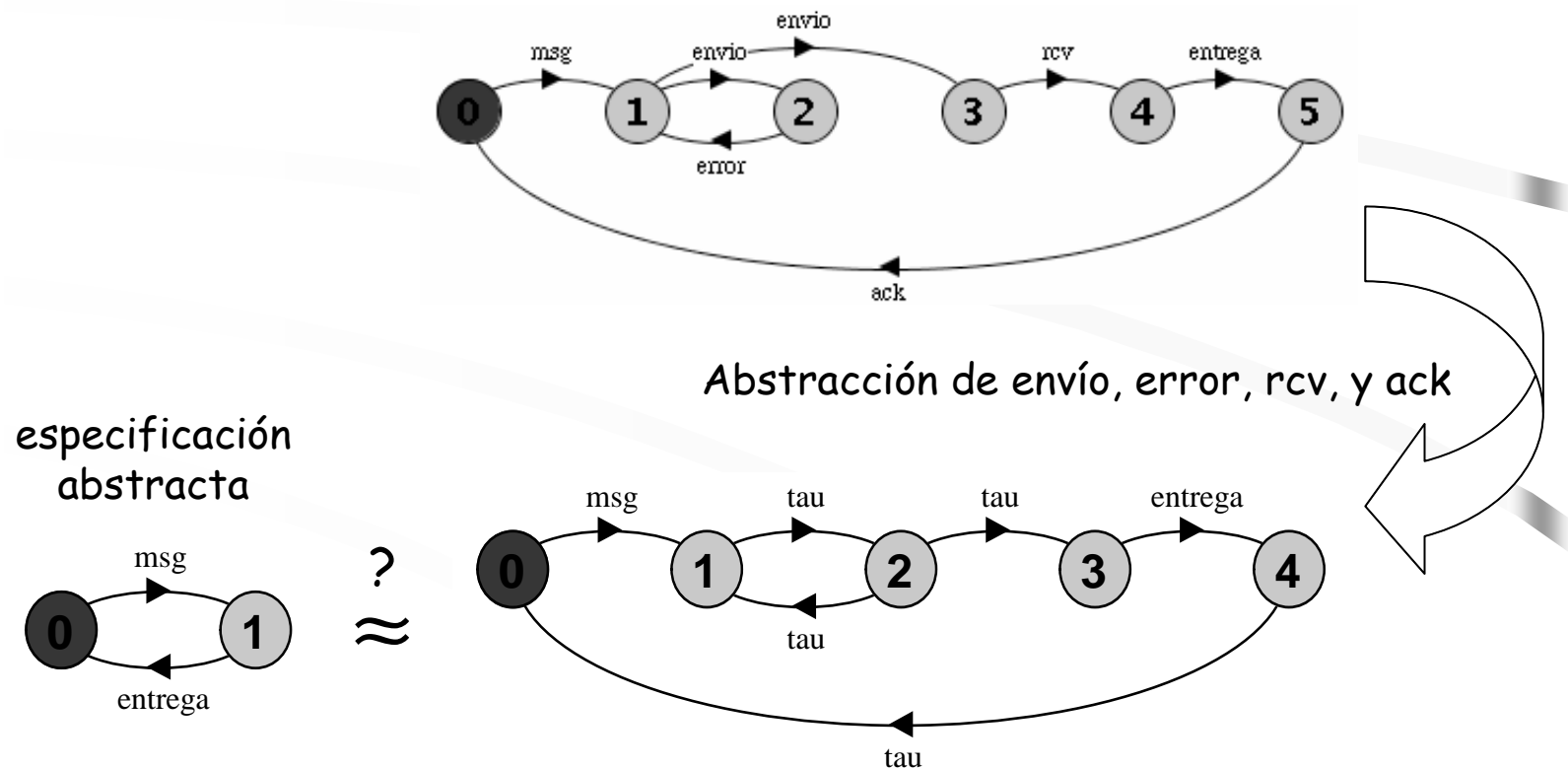


Como podemos saber
si el protocolo es
correcto?



Verificación Contra Especificación Abstracta

- ¿La especificación abstracta es equivalente a una vista abstracta del protocolo?



Resumen LTS

- Una máquina de estados muy básica para
 - describir comportamiento de entidades y
 - razonar sobre el comportamiento emergente resultante de su ejecución concurrente
- Semántica
 - Definimos una equivalencia que abstrae diferencias sintácticas pero preserva intuición: Bisimulación
 - Ejemplifiqué la utilidad de Bisimulación, Abstracción y minimización con respecto a equivalencias.
 - Lineal vs Branching
- Las variantes de máquinas de estados que veremos tienen nociones de equivalencia, abstracción y minimización similares

Modelado con LTS

- Ojo con la denotación
- Un LTS modela un agente
 - Eventos son fenómenos monitorieables / controlables
 - Trazas/Arboles de ejecución describen la relación temporal esperada entre lo monitoreable y lo controlables
 - Sincronización entre LTS representa la relación monitoreabilidad/controlabilidad
 - Ferte vínculo con el Modelo de Agentes
- Un LTS modela una entidad pasiva
 - Eventos son fenómenos aplicados a la entidad (i.e. operaciones sobre o que afectan la entidad)
 - Explica el protocolo de uso esperado de la entidad.

Denotación: Sutilezas

- Si "rcv" es un evento monitoreado por el receptor, y... existen estados donde "rcv" no está habilitado...

¿Qué denotación es razonable para "rcv"?

- "Un mensaje es recibido por el receptor"
 - "Un mensaje está disponible para el receptor"
 - "Algo intenta entregarle un mensaje al receptor"
- Una variante común:
 - Input-enabledness

Maquinas de Estado Finitos

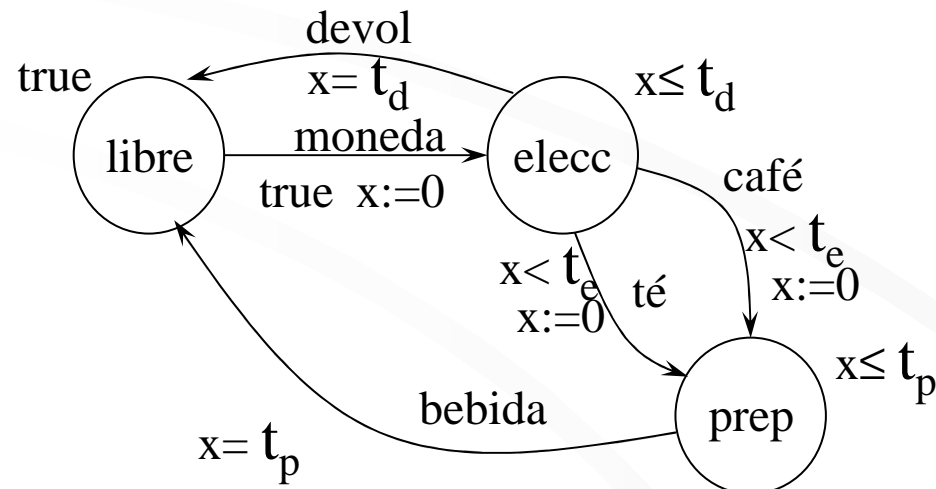
- Muchas variantes más expresivas
 - datos, tiempo, probabilísticas, etc.
 - cuanto mas expresivo
 - mas cercano al código
 - menos abstracto
 - menos analizable (tanto manual como automáticamente)
- Pariente cercanos: Algebras de Proceso

Veamos algunos ejemplos...

Especificación de sistemas

Agregemos al **qué** el **cuando** debe hacer una acción un sistema.

- Esperar una moneda (siempre que este libre).
- Permitir la elección café / té (antes de un tiempo t_e).
- Preparar la bebida seleccionada (demorando un tiempo t_p).
- Devolver la moneda si ninguna bebida fue seleccionada (luego de un tiempo t_d de colocada la moneda).



Semántica informal

- Un *estado* del sistema es un par (s, v) con
 - $s \in S$,
 - $v \in V$ y
 - $Is[v] = \text{true}$.
- La arista $a = \langle s, l, \psi, \alpha, s' \rangle \in A$ puede ser *atravesada* desde el estado (s, v) si v satisface la restricción ψ .
- El estado resultante (s', v') es tal que $\alpha(v) = v'$.
- El tiempo pasa solo en los nodos.
- Atravesar una arista no insume tiempo.
- Los valores de los relojes crecen *uniformemente* con el tiempo.
- El sistema puede permanecer en un nodo s mientras los valores de los relojes satisfagan Is .

Semántica formal

Conjunto
infinito

Sistema de Transición Etiquetado $T = \langle Q, \rightarrow \rangle$

- Estados $Q = \{ (s, v) \mid s \in S, \text{Is}[v] = \text{true} \}$
- Relación de transición $\rightarrow \subseteq Q \times \{L \cup \mathbb{R}_+\} \times Q$

con dos tipos de transiciones:

- **Temporales:** el paso de un tiempo t es representado por una transición etiquetada con t .

$$\frac{\text{Is}[v+t], 0 \leq t}{(s, v) \rightarrow t (s, v+t)}$$

- **Instantáneas:** dadas por la ejecución de una acción e . La transición llevará la etiqueta e .

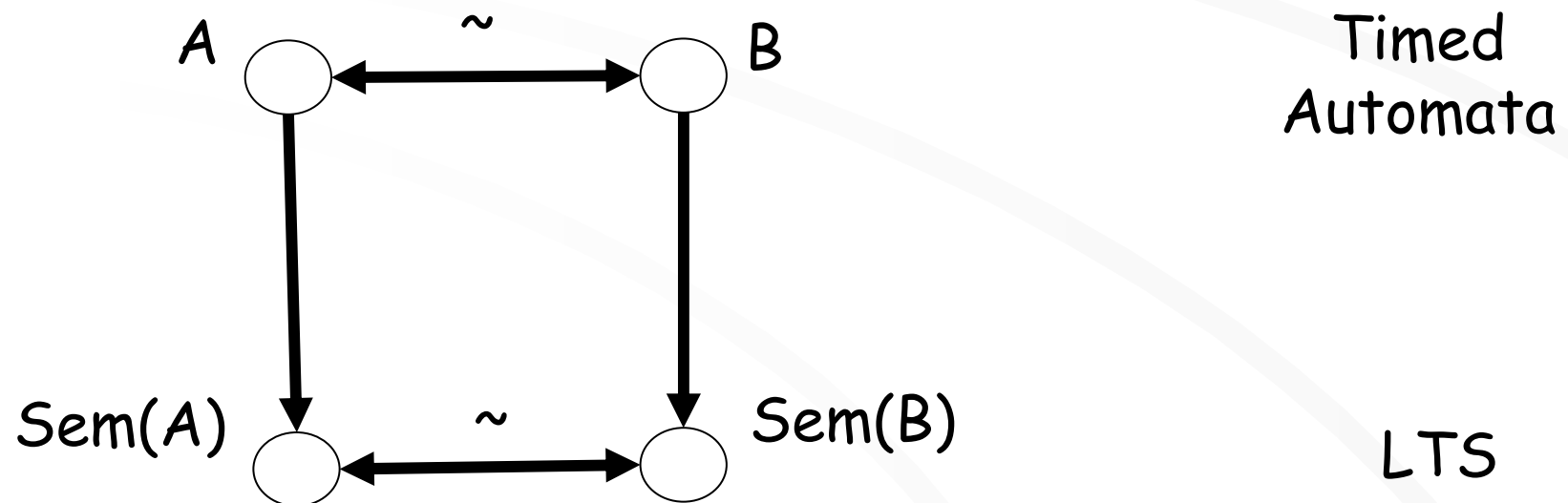
$$\frac{\langle s, e, \psi, \alpha, s' \rangle \in A, \psi[v] \text{ y } \alpha(v) = v'}{(s, v) \rightarrow e (s', v')}$$

Notación:

Escribimos $q \rightarrow_e q'$ en lugar de $(q, e, q') \in \rightarrow$.

Semántica de Autómata Temporalizada

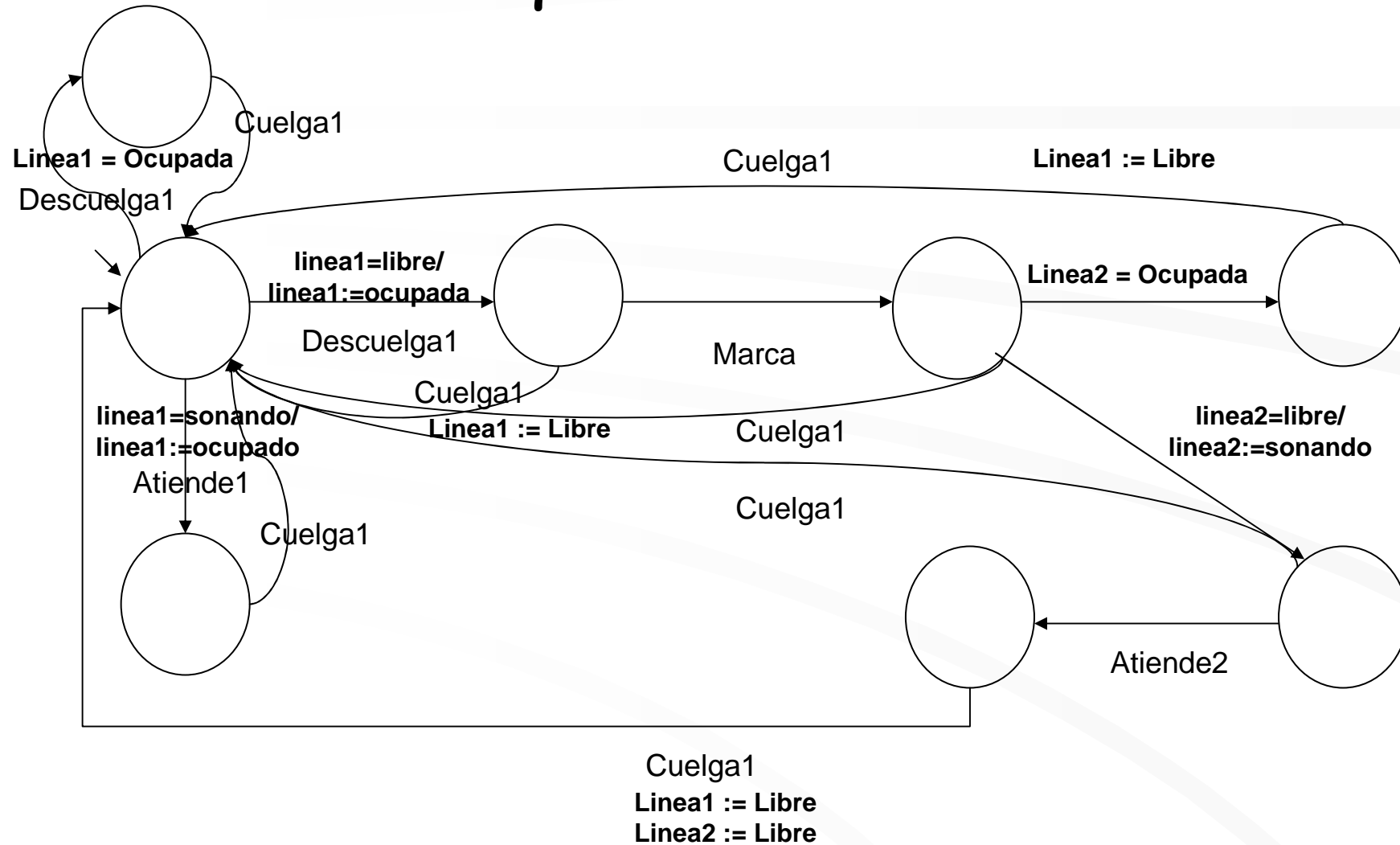
- ¿Cómo le damos semántica a TA?
 - Explicando TA en términos de otra notación conocida: LTS
- La semántica de TA "hereda" una noción de bisimulación y la operación de composición en paralelo
 - Podríamos intentar dar una noción de bisimulación y composición en paralelo que debería "encajar" con la de LTS



Máquinas de Estados con Variables

- Ejemplo
- Semántica

Maquinas de Estado Finitos:



Statecharts

- Características claves:
 - Estados jerárquicos y ortogonales
- Otros aspectos (según variante)
 - Distinición entre eventos internos y externos
 - Un evento "externo" por vez.
 - Broadcasting de eventos
 - Etiquetas causa/efecto
 - Colas de eventos
 - Transiciones multi-nivel
- Una notación con problemas

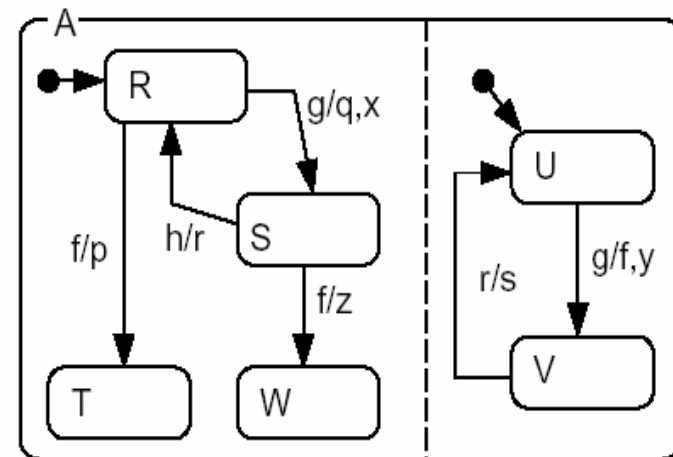
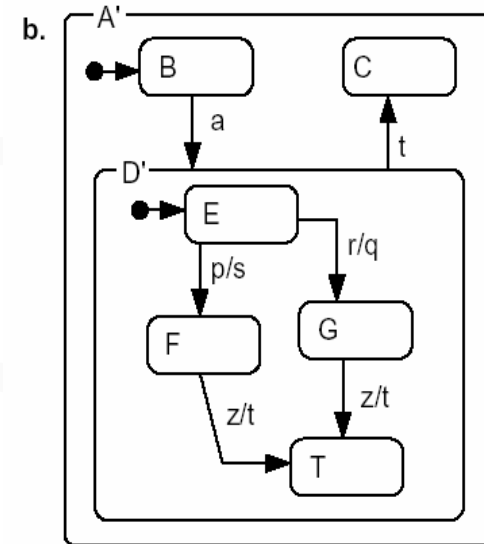
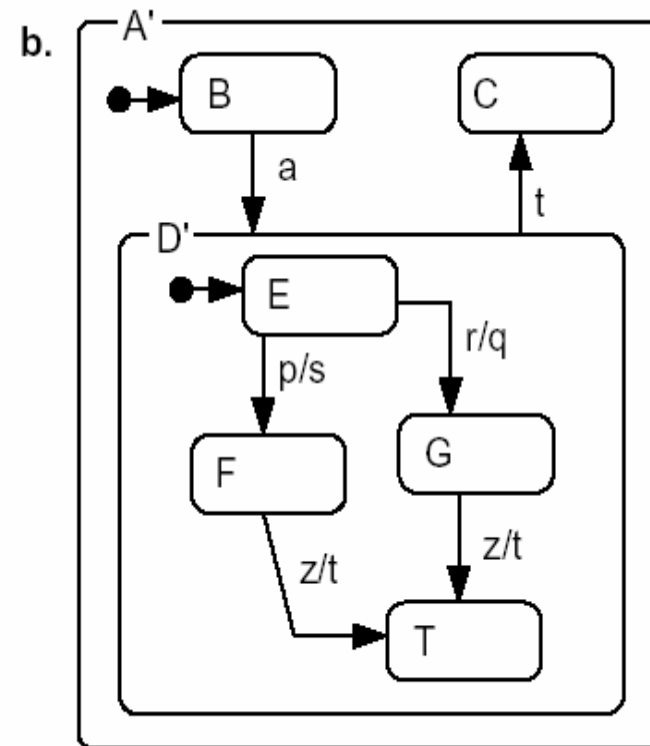
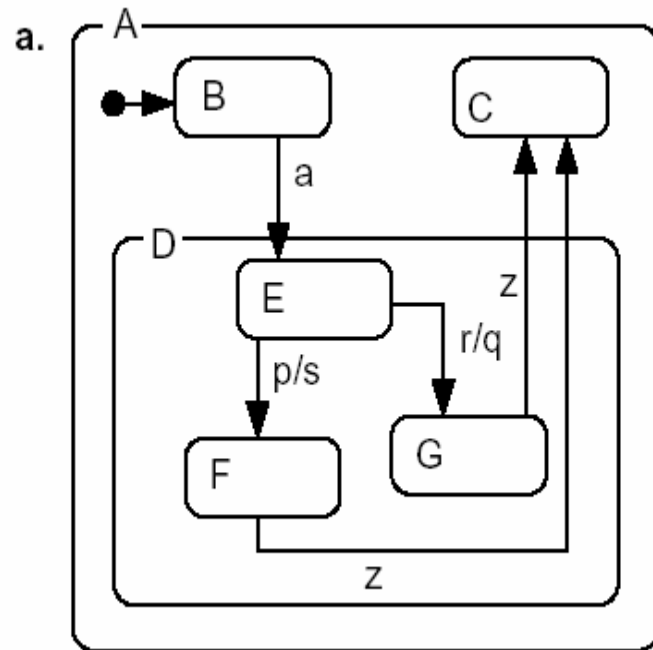
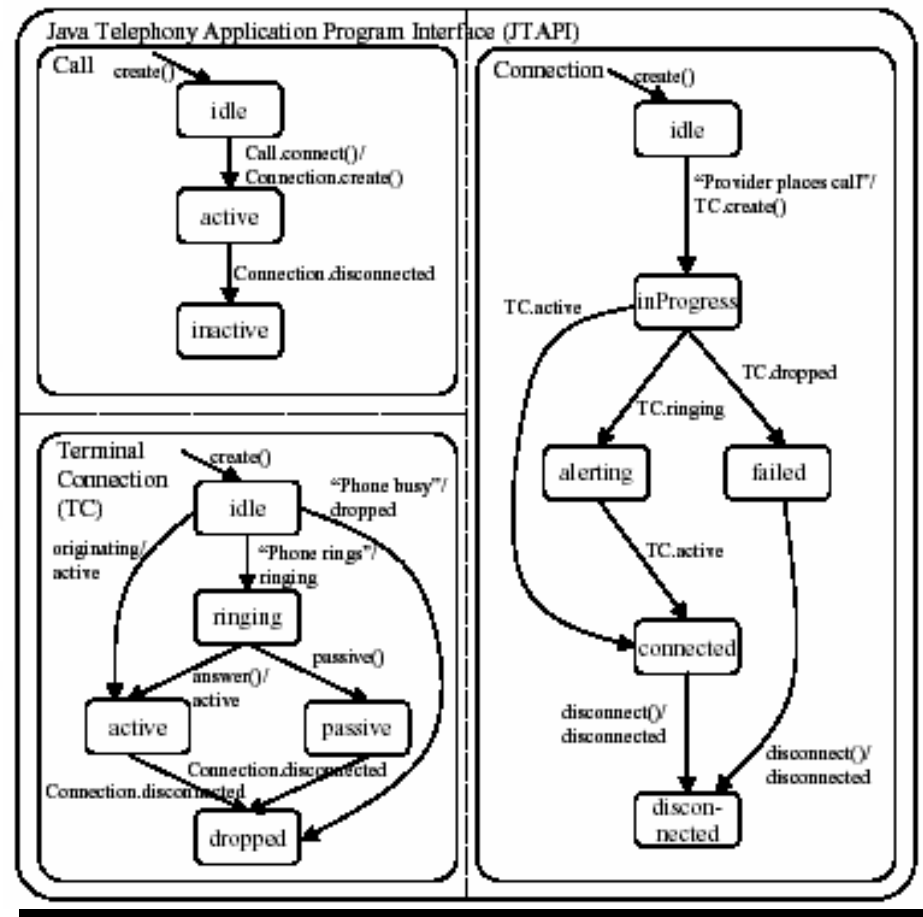


Figure 1. Event processing example

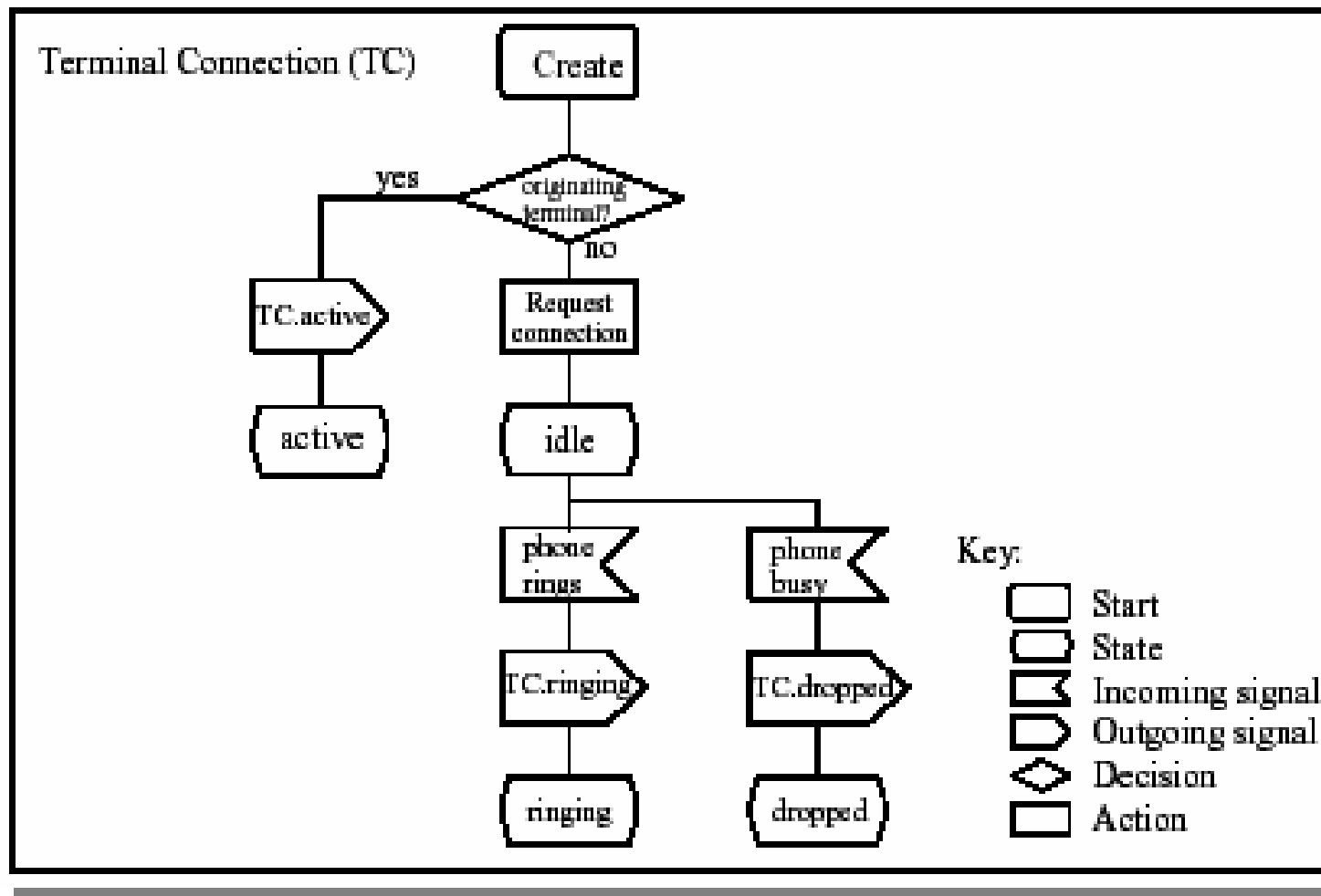
Problema con el Interlevel transtion



Ejemplo real de Statechart



SDL Flow Chart



Maquinas de Estado

- Relación con Otros Modelos -

- Diagrama de Contexto
 - Cuál es la relación temporal entre los fenómenos en la interfaz de agentes
 - Variantes de LTS distinguen monitoreado y controlado
- Modelos de objetivos
 - Las trazas o árbol de ejecución satisface las expectativas o requerimientos asociados al agente?
 - El comportamiento de la composición satisface los objetivos
- Modelo de clases
 - Los estados son abstracciones de valuaciones de atributos?
- Modelo de operaciones
 - Los cambios de estado de la ME se corresponden a pre/post condiciones de operaciones.