

Varios ejercicios presentados aquí fueron extraídos de *Concurrency, State Models and Java Programs*, 2nd ed. (Magee & Kramer 2006).

✓ Ejercicio 1. Lea los capítulos 4, 5 y 10 de *Concurrency, State Models and Java Programs*, 2nd ed. (Magee & Kramer 2006).

Ejercicio 2. Considere el problema de lectores y escritores que desean acceder a un recurso común (por ej. una variable) donde los lectores sólo pueden leer de ese recurso mientras los escritores pueden leer y escribir. Además de las consideraciones usuales del problema de lectores-escritores, por una restricción de la administración del recurso, sólo se permite que  $K$  procesos lo estén accediendo simultáneamente. Suponga que en el sistema hay  $N$  lectores y  $M$  escritores (con  $N$  y  $M$  mayores a  $K$ ). Modele el problema descrito en FSP, proveyendo además de las definiciones de los procesos correspondientes, el diagrama de estructura de su modelo.

*Nota.* Ejercicio 3. Una computadora central conectada a terminales remotas a través de enlaces de comunicación se utiliza para realizar reservas de asientos en un teatro. Los empleados del teatro que realizan las reservas pueden mostrar el estado actual de reservas en las pantallas de las terminales. Para poder reservar un asiento, el cliente elige el número, y un empleado accede a través de la terminal para saber si el mismo está reservado o no. Si no lo está, realiza la reserva correspondiente. Construya un modelo FSP para este problema, asegurando que no puede reservarse más de una vez el mismo asiento en el teatro.

✓ Ejercicio 4. *The Dining Savages*: Una tribu de salvajes disfruta de una cena comunitaria. Cada uno se sirve de una gran olla que contiene hasta  $M$  porciones de estofado de misionero. Cuando un salvaje quiere comer se sirve una porción de la olla al menos que esta este vacía en cuyo caso espera hasta que esta se vuelva a llenar. Si la olla se vacía, el cocinero de la tribu la vuelve a llenar con  $M$  porciones. Modele los salvajes, el cocinero y la olla.

✓ Ejercicio 5. Diseñe y modele en FSP un protocolo de comunicación que permita a un proceso productor y un proceso consumidor comunicarse a través de un canal asíncrono con un número máximo de mensajes, de manera que el productor se bloquee cuando el buffer del canal está lleno.

*Difícilísimo* Ejercicio 6. Modele en FSP el protocolo de Reconocimiento Positivo con Retransmisión: PAR ("Positive Acknowledgement with Retransmission")<sup>1</sup>. Para ello modele el transmisor, el receptor y los canales de ida y de vuelta. Observe que estos canales corresponden a un medio con pérdida y duplicación.

Ejercicio 7. Si se siente con ganas modele también el *Go Back N*<sup>1</sup> con tamaño de ventana  $N = 1$ . (Si logró hacer el PAR, no debería ser difícil.)

Ejercicio 8. Si se siente eufórico modele el *Go Back N* con tamaño de ventana  $N > 1$ . Note que en este caso, y para que tenga sentido el modelo, los canales deben poder encontrarse transportando a la vez una cantidad mayor a  $N$  de mensajes (eso sí, siempre en orden, aunque se puedan duplicar o perder).

<sup>1</sup>Para los detalles de estos protocolos busque en el moodle los apuntes correspondientes (sección Material Adicional).



Ejercicio 9. La empresa *nomemuevodemicasa.com* se dedica a la venta de libros a través de internet. Para efectuar la venta de un libro, en respuesta a la solicitud de compra de un cliente, la empresa debe realizar dos tareas:

- comprobar la disponibilidad del libro por parte de la editorial correspondiente, y
- comprobar que la tarjeta de crédito del cliente (las ventas se realizan sólo mediante tarjeta de crédito) cuenta con fondos disponibles.

En caso de no contar la editorial con ejemplares disponibles o la tarjeta del cliente con fondos, la compra se rechaza; caso contrario, la misma se acepta y se efectúa. Para simplificar el problema, consideremos que la empresa interactúa sólo con una editorial y una tarjeta de crédito, *Nerd's Press* y *Plasti-Card*, respectivamente.

Modele el problema descrito en FSP, proveyendo además de las definiciones de los procesos correspondientes, el diagrama de estructura de su modelo.

Ejercicio 10. En la peluquería de Fígaro y don Mateo hay dos peluqueros (Fígaro y don Mateo) y  $N$  sillas para que los clientes esperen sentados. Si un peluquero no tiene cliente que atender, entonces se echa a dormir una siesta. Cuando un cliente llega a la peluquería si algún peluquero está durmiendo, lo despierta y se hace atender. Si ambos peluqueros están atendiendo entonces espera si hay asientos libres donde sentarse, o se va si todos los asientos están ocupados. Cabe destacar que en esta peluquería los pelados tienen prioridad (pero no pueden interrumpir un corte de pelo).

Modele el sistema utilizando FSP, proveyendo el diagrama de estructura del modelo. (Sugerencia: Trate a los clientes como *jobs*, no como procesos independientes).

Ejercicio 11. Muestre que los siguientes procesos son <sup>fuerte</sup> (fuertemente) bisimilares

$$\begin{aligned} P &= ( a \rightarrow ( b \rightarrow c \rightarrow P \\ &\quad | b \rightarrow c \rightarrow P ) \\ &\quad ). \\ Q &= ( a \rightarrow b \rightarrow R \\ &\quad | a \rightarrow b \rightarrow R ), \\ R &= ( c \rightarrow Q ). \end{aligned}$$

Ejercicio 12. Considere el siguiente proceso FSP:

$$\begin{aligned} P &= ( a \rightarrow ( t \rightarrow b \rightarrow \text{STOP} \\ &\quad | t \rightarrow c \rightarrow \text{STOP} ) \\ &\quad ). \\ ||Q &= P \setminus \{t\}. \end{aligned}$$

En LTSA minimice el proceso  $Q$  y observe el LTS obtenido. Luego, valla a "Option" y desmarque la opción "Enable tau redaction". Vuelva a minimizar  $Q$  y observe el nuevo LTS. ¿Cuál de los dos es la minimización correcta de acuerdo a la bisimulación débil? Justifique su respuesta.

Ejercicio 13. Muestre que los siguientes procesos son débilmente bisimilares

$$\begin{aligned} P &= ( a \rightarrow ( b \rightarrow \text{STOP} \\ &\quad | \tau \rightarrow c \rightarrow \text{STOP} ) \\ &\quad ). \\ Q &= ( a \rightarrow ( b \rightarrow \text{STOP} \\ &\quad | \tau \rightarrow c \rightarrow \text{STOP} ) \\ &\quad | a \rightarrow c \rightarrow \text{STOP} \\ &\quad ). \end{aligned}$$

Ejercicio 14. Muestre que  $P$  simula a  $Q$  (i.e. que hay una relación de simulación  $S$  tal que  $(0_Q, 0_P) \in S$ ) pero no al revés.

$$\begin{aligned} P &= ( a \rightarrow ( b \rightarrow \text{STOP} \\ &\quad | c \rightarrow \text{STOP} ) \\ &\quad ). \\ Q &= ( a \rightarrow b \rightarrow \text{STOP} \\ &\quad | a \rightarrow c \rightarrow \text{STOP} ). \end{aligned}$$

Este ejercicio es distinto al 11 porque en el 11 a pesar de una indeterminación los estados de  $Q$  a los que se llega son equivalentes. En cambio en este la indeterminación define estados distintos.



$$Q = \{(1, A), (2, B), (3, C), (4, D), (4, A)\}$$

## GUIA NP2

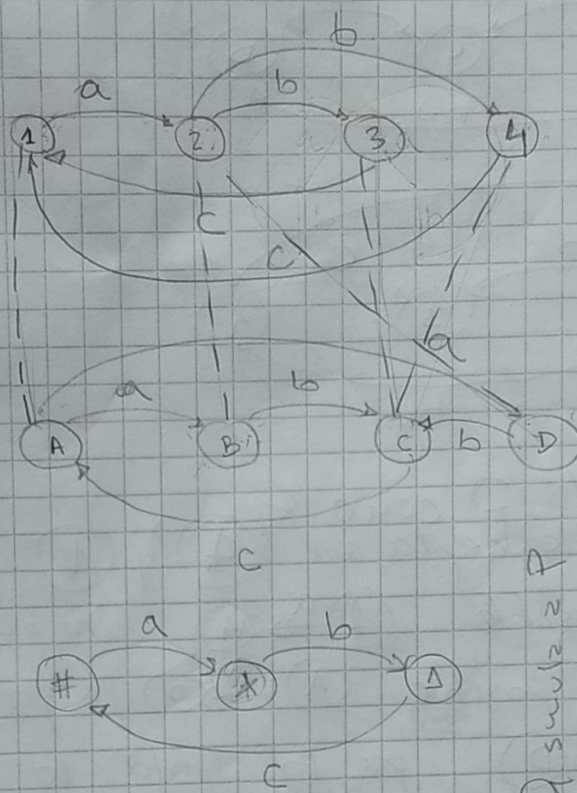
### EJERCICIO 11

$A \xrightarrow{a} B$   
 $A \xrightarrow{b} D$   
 $B \xrightarrow{b} C$   
 $D \xrightarrow{b} C$   
 $C \xrightarrow{c} A$

$1 \xrightarrow{a} 2$  ✓  
 $1 \xrightarrow{b} 4$  ✓  
 $2 \xrightarrow{b} 3$  ✓  
 $2 \xrightarrow{b} 4$  ✓  
 $3 \xrightarrow{c} 1$  ✓

P simula a Q

P  
 1  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67  
 68  
 69  
 70  
 71  
 72  
 73  
 74  
 75  
 76  
 77  
 78  
 79  
 80  
 81  
 82  
 83  
 84  
 85  
 86  
 87  
 88  
 89  
 90  
 91  
 92  
 93  
 94  
 95  
 96  
 97  
 98  
 99  
 100



$f: Q \rightarrow P$

$f(1) = 1$   $P \rightarrow Q$

$f(2) = 2$   $1 = A$

$f(3) = 3$   $2 = B$

$f(4) = 4$   $3 = C$

$f(1) = 1$

$1 \xrightarrow{a} 2$   $A \xrightarrow{a} B$  ✓

$2 \xrightarrow{b} 3$   $B \xrightarrow{b} C$  ✓

$2 \xrightarrow{b} 4$   $B \xrightarrow{b} C$  ✓

$3 \xrightarrow{c} 1$   $C \xrightarrow{c} A$  ✓

$4 \xrightarrow{c} 1$   $C \xrightarrow{c} A$  ✓

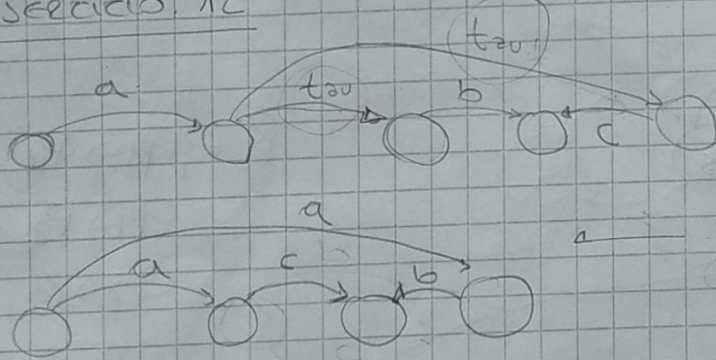
$Q(1) = A$

$Q(2) = B$

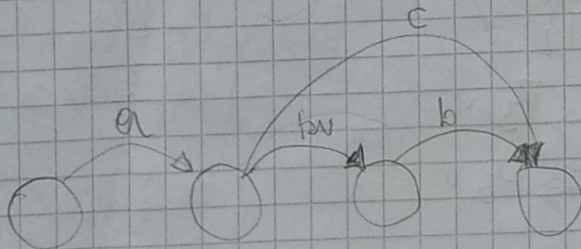
$Q(3) = C$

$Q(4) = A$

### EJERCICIO 12



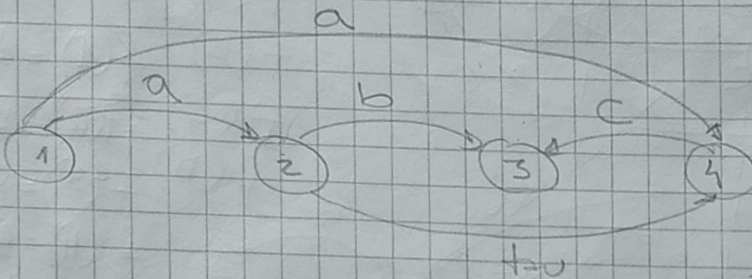
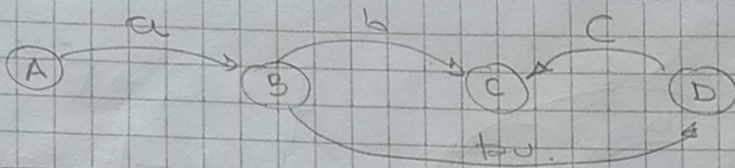
Este es una minimización  
 correcta ya que  
 tanto para b como para  
 c se necesita una  
 transición tau para  
 llegar a ellos.



Ace en cambio hay stop  
 que cambia entre elegir b y c  
 y es dicha transición tau.  
 Como para c no es nec.  
 ninguna transición entonces  
 sería injusto ponerla en  
 el mismo lugar que b.



### EXERCICIO 13



$$f: P \rightarrow Q$$

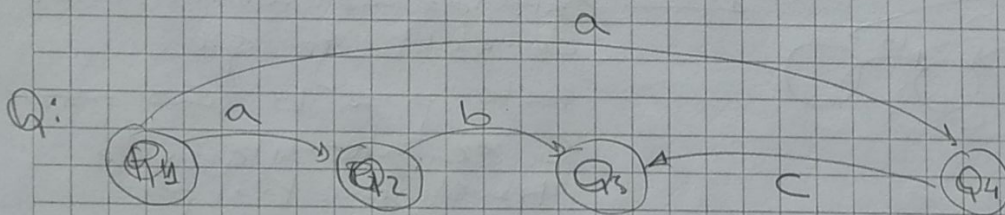
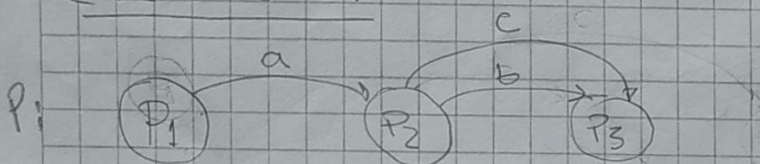
$$f(A) = 1$$

$$f(D) = 4$$

Nota que tanto P como Q son iguales excepto en A y 1.

En 1 es posible  $1 \xrightarrow{a} 4$  mientras que en A no es posible  $A \xrightarrow{a} D$  pero si existe  $A \xrightarrow{a} (\tau)^+ \rightarrow D$  por lo tanto lo consideramos DEBILMENTE BISIMILARES.

### EXERCICIO 14



$$R = \{ Q1 \rightarrow P1, Q4 \rightarrow P2, Q2 \rightarrow P2, Q3 \rightarrow P3 \}$$

✓

$$Q1 \xrightarrow{a} Q2$$

$$Q1 \xrightarrow{a} Q4$$

$$Q2 \xrightarrow{b} Q3$$

$$Q4 \xrightarrow{c} Q3$$

∃

$$P1 \xrightarrow{a} P2 \checkmark$$

$$P1 \xrightarrow{a} P2 \checkmark$$

$$P2 \xrightarrow{b} P3 \checkmark$$

$$P2 \xrightarrow{c} P3 \checkmark$$

⇒ R ⇒

$$\begin{cases} Q1 \xrightarrow{a} Q2 \checkmark \\ Q2 \xrightarrow{b} Q3 \checkmark \\ Q2 \xrightarrow{c} Q3 \times \end{cases}$$

$$\begin{cases} Q1 \xrightarrow{a} Q4 \checkmark \\ Q4 \xrightarrow{c} Q3 \checkmark \\ Q4 \xrightarrow{b} Q3 \times \end{cases}$$

Se ve que P y Q son equivalentes en el sentido de la bisimilitud débil.



ejerc 4

the dining. savages.

RANGE T = 0..4

OLLA = OLLA(4)

OLLA(m:T) = ( when m > 0 porción → OLLA(m-1)  
when m = 0 carga → OLLA(4) )

COCINEO = (carga → COCINEO.)

SALVAJE = (2C-cucharon → porción → rel-cucharon)

CUCHARON = (2C-cucharon → rel-cucharon  
→ CUCHARON)

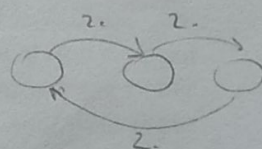
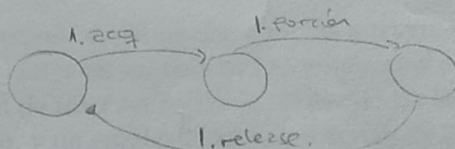
Eslo sine solo  
p22 in salvaje.

Para 2 salvajes

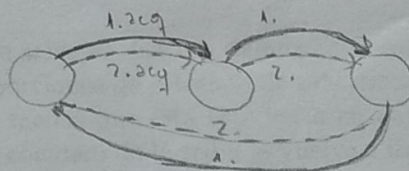
|| SALVAJE\_ORD = (SALVAJE || CUCHARON)

|| DINING = (COCINEO || SALVAJE\_ORD || OLLA)

|| SALVAJES\_ORD = ( {1,2} :: SALVAJE || CUCHARON )



|| DINING = ( {1,2} :: SALVAJES\_ORD || OLLA  
|| COCINEO )



## EXERCICIO 5

PROD [ ] [ ] [ ] [ ] CONSUM

RANGE T = 0..4

BUFFER = BUFFER[0]

BUFFER(n:T) = ( when n > 0 get → BUFFER[n-1]  
when n < T put → BUFFER[n+1]  
)

PRODUTOR = put → PRODUTOR

CONSUMIDOR = get → CONSUMIDOR

|| ASINC = (PRODUTOR || CONSUMIDOR || BUFFER)



# EXERCICIO 9

VENTA  $[s: \{0,1\}] = (\text{when } (s=0) \text{ error} \rightarrow \text{VENTA} \mid \text{when } (s=1) \text{ OK} \rightarrow \text{VENTA})$

VENTA = (petición  $\rightarrow$  CHECK).

CHECK = (ed-check  $\rightarrow$  (ed-ok  $\rightarrow$  card-check  $\rightarrow$  (card-ok  $\rightarrow$  VENTA[1])  
 $\mid$  ed-nok  $\rightarrow$  VENTA[0])  
 $\mid$  card-nok  $\rightarrow$  VENTA[0])

EDITORIAL = (ed-ok  $\rightarrow$  EDITORIAL  $\mid$  ed-nok  $\rightarrow$  EDITORIAL).

CARD = (card-ok  $\rightarrow$  CARD  $\mid$  card-nok  $\rightarrow$  CARD).

