Ingeniería del Software 2

Taller 6 – Procesos Secuenciales

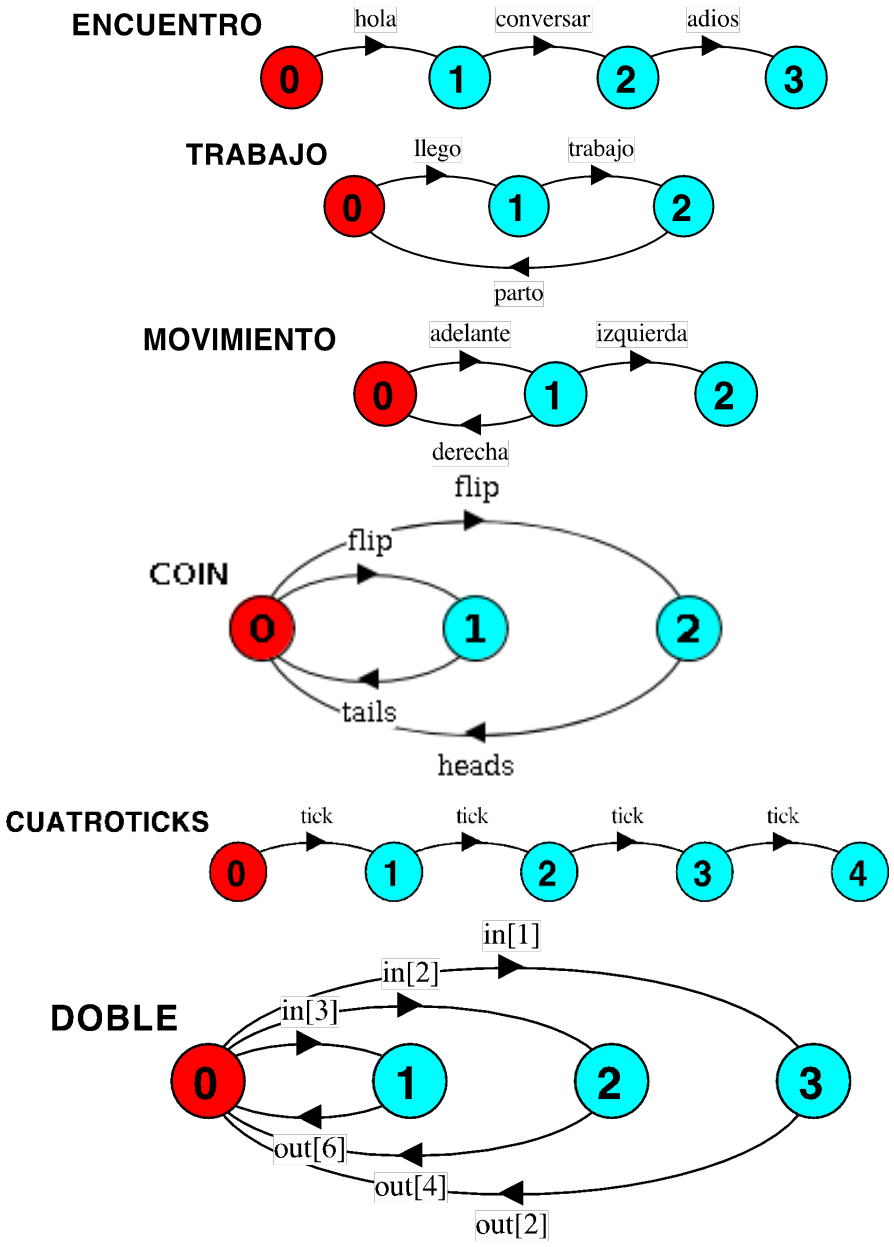
DEADLINE: 16 de julio de 2020

La herramienta LTSA se encuentra en el siguiente link. La herramienta se levanta corriendo $ java -jar ltsa.jar

La sintaxis de FSP puede encontrarse en acá[1]

Ejercicio 1

Para cada uno de los siguientes LTS, dar una expresión en FSP (lo más compacta posible) cuya semántica representa al LTS dibujado.



[1] https://www.doc.ic.ac.uk/~jnm/LTSdocumention/FSP-Syntax.html

Ejercicio 2

Una variable guarda valores entre 0..N y permite ser leída y escrita a través de eventos *read* y *write*. Modelar la variable como un proceso, **VARIABLE**, usando FSP. Considere que la variable empieza inicializada en 0.

Para N=2, el modelo debería exhibir la siguiente traza:

$$\text{write}[2] \rightarrow \text{read}[2] \rightarrow \text{read}[2] \rightarrow \text{write}[1] \rightarrow \text{write}[0] \rightarrow \text{read}[0] \rightarrow \ldots$$

Para N=2, el modelo **NO** debería exhibir la siguiente traza:

$$\text{write}[2] \rightarrow \text{read}[2] \rightarrow \text{read}[1] \rightarrow \ldots$$

Ejercicio 3

Un sensor mide el nivel de agua de un tanque mediante el evento *sense*. El nivel es inicialmente 5 y puede variar (no-deterministicamente) entre 0 y 9 de un paso a la vez. El sensor emite una señal *low* cuando el nivel de agua pasa a ser menor que 2 y emite una señal *high* cuando el nivel pasa a ser mayor que 8. Cuando el nivel vuelve al rango entre 2 y 8, el sensor emite una señal *normal*. Modelar el sensor como un solo proceso FSP.

Ejercicio 4

Dado el siguiente proceso que describe el comportamiento de una variable:

```
range 0..3

VARIABLE = VARIABLE[0],
VARIABLE[i:R] = ( read[i] -> VARIABLE[i] | write[j:R] -> VARIABLE[j] ).
```

Modelar los siguientes procesos:

a) Un proceso **ESCRITOR** que escribe un valor entre 0 y N en la variable.
b) Un proceso **LECTOR** que lee un valor y si es distinto de cero lo imprime con *imprimir[i]*.
c) La composición paralela de todos los anteriores.

Ejercicio 5

Un museo que puede albergar hasta N turistas, permite entrar a los turistas por la entrada oriental y salir por la salida occidental. Arribos y partidas se señalan al controlador del museo mediante señales *entry* y *exit* emitidas por molinetes. Cuando el museo debe abrir, el director del mismo le da la señal *open* al controlador y el controlador permite el ingreso y egreso de visitantes. A la hora de cerrar, el director da la señal *close* al controlador que a partir de ese momento solo permite egresos. Cuando el museo está vacío, el director recibe la señal *empty* del controlador. El director no reabre el museo hasta que el museo se encuentra sin turistas dentro.

a) Modele un proceso **ENTRADA** que represente la entrada al museo aceptando eventos *entry*.
b) Modele un proceso **SALIDA** que represente la salida del museo aceptando eventos *exit*.
c) Modele un proceso **DIRECTOR** que abra, cierre y espere a que se vacie el museo.
d) Modele un proceso **CONTROL** que se encargará de que todo el sincronizado funcione correctamente: primero espera a que el director abra el museo, luego deja entrar y salir gente siempre que sea permitido hasta que el museo cierra. En este momento solo deja salir a los turistas y eventualmente confirma estar vacío.
e) La composición de todos los anteriores.

Ejercicio 6

Considere los procesos **P** y **Q** descritos en FSP y su composición paralela **S**:

```
P = (a -> b -> P).
Q = (c -> b -> Q).
||S = (P || Q).
```

a) Describa en FSP un proceso secuencial equivalente a **S**.
b) Muestre un proceso **R**, tal que compuesto con **S** restrinja el conjunto de trazas a las secuencias donde por cada evento *a* sucede un solo evento *b*. Luego *b* no sucede hasta que no suceda primero una *a*.
c) Muestre un proceso **T**, tal que compuesto con **S** permita la ocurrencia opcional de un evento *d* por cada evento *c* observado.

Formato de Entrega

El taller debe ser entregado a través del campus en la fecha de entrega indicada en el documento. Deben entregar un archivo `entrega.lts` con el código implementado en FSP. Además, deberán entregar un informe con los LTS resultantes. Justifique las decisiones de modelado a la hora de definir los LTS. Sugerencia: primero modele en LTS y luego escribir el FSP a partir de eso.