



DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Ingeniería de software II Trabajo Práctico N°5

Integrante	LU	Correo electrónico
Esteban Luciano Rey	657/10	estebanlucianorey@gmail.com



Facultad de Ciencias Exactas y Naturales Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (++54 +11) 4576-3300

<http://www.exactas.uba.ar>

Índice

1. Respuestas	2
1.1. Ejercicio 1	2
1.2. Ejercicio 2	3
1.3. Ejercicio 3	3
1.4. Ejercicio 4	4
1.5. Ejercicio 5	5
1.6. Ejercicio 6	7

1. Respuestas

1.1. Ejercicio 1

Encuentro

```
ENCUENTRO = (hola->conversar->adios->STOP).
```

Trabajo

```
TRABAJO(N=0) = (llego->trabajo->TRABAJO[1]),  
TRABAJO[1] = (parto->TRABAJO).
```

Movimiento

```
MOVIMIENTO = (adelante-> ( izquierda->STOP | derecha->MOVIMIENTO)).
```

Coin

```
COIN = ( flip->heads->COIN | flip->tails->COIN ).
```

Cuatro ticks

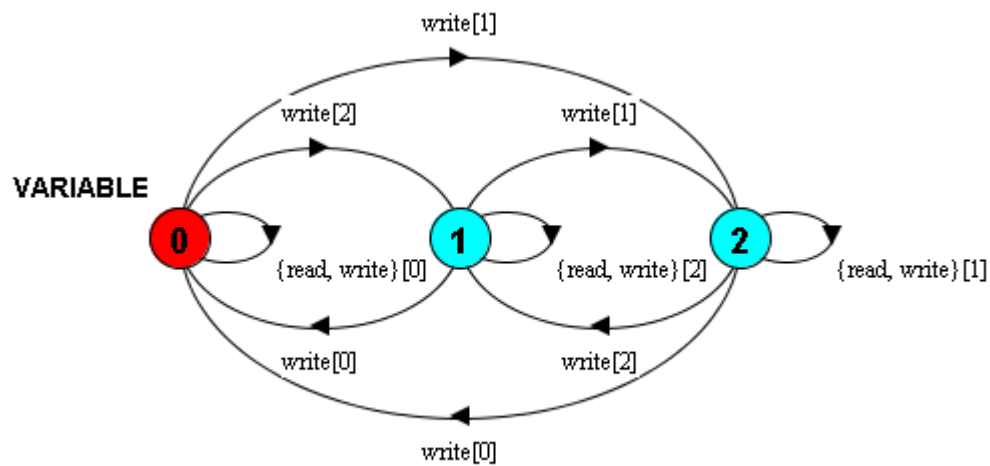
```
CUATROTIKS = (tick-> tick-> tick-> tick->STOP).
```

Double

```
DOUBLE = (in[d:D]->out[d*2]->DOUBLE).
```

1.2. Ejercicio 2

LTS



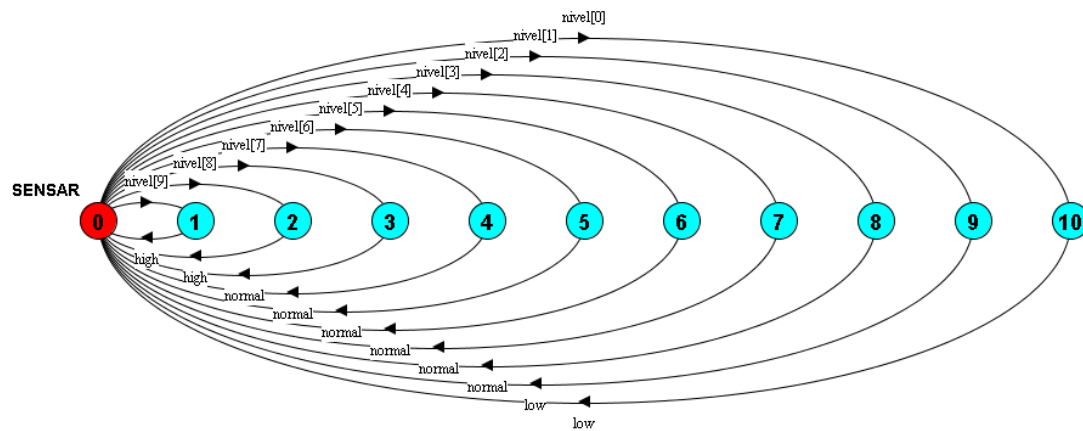
FSP

```

const N = 2
VARIABLE = VARIABLE[0],
VARIABLE[v:0..N] = (write[n:0..N]->VARIABLE[n] | read[v]->VARIABLE[v] ).
  
```

1.3. Ejercicio 3

Solo se muestra el fsp del nivel porque el lts, dada el no-determinismo, genera un grafo muy cargado y poco claro.

LTS**FSP**

```

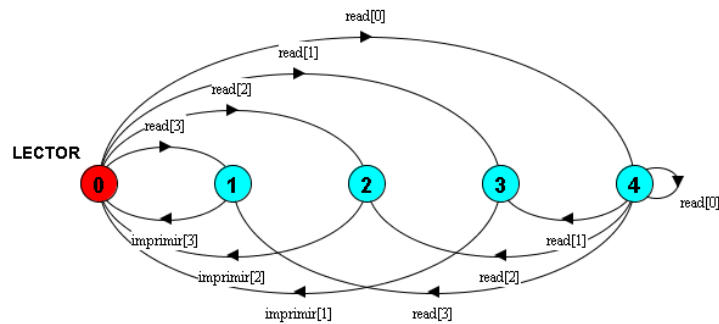
range N = 0..9
NIVEL = NIVEL[5],
NIVEL[k:N] = (nivel[k] ->( when(k<9) variar -> NIVEL[k+1] |when(k>0) variar ->
    NIVEL[k-1] )).

SENSOR(N=5) = (nivel[n:N]->(when (n<2) low->SENSOR | when (n<8 & n>=2) normal->SENSOR
    |when (n>=8) high->SENSOR )).
  
```

1.4. Ejercicio 4**LTS Escriptor****ESCRITOR****FSP Escriptor**

```

ESCRITOR = (write[e:R]->ESCRITOR).
  
```

LTS Lector**FSP Lector**

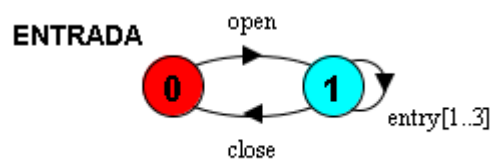
```

LECTOR = (read[e:R]-> LECTOR[e]),
LECTOR[e:R] = (when(e!=0) imprimir[e]->LECTOR | when(e==0) read[i:R]->LECTOR[i] ).
  
```

FSP ESCRITORLECTOR

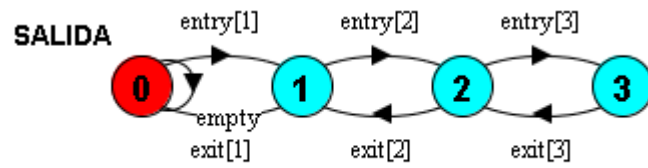
```

|| ESCRITORLECTOR = ( ESCRITOR || LECTOR || VARIABLE ).
  
```

1.5. Ejercicio 5**LTS Entrada****FSP Entrada**

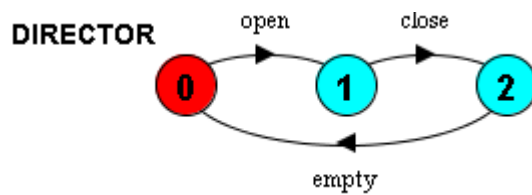
```

ENTRADA = (open -> ENTRADA[0]),
ENTRADA[0] = (entry[i:1..C]->ENTRADA[0] | close -> ENTRADA).
  
```

LTS Salida**FSP Salida**

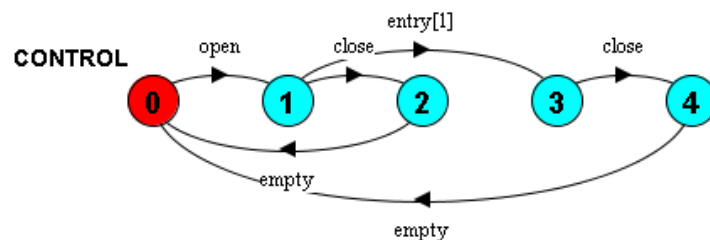
```

SALIDA = (entry[1] -> SALIDA[1] | empty -> SALIDA),
SALIDA[i:1..C] = (when(i==1) exit[1]->SALIDA | when(i>1) exit[i] -> SALIDA[i-1] |
  when(i<C) entry[i+1]->SALIDA[i+1]).
  
```

LTS Director**FSP Director**

```

DIRECTOR = DIRECTOR[0],
DIRECTOR[0] = (open->DIRECTOR[1]),
DIRECTOR[1] = (close->DIRECTOR[2]),
DIRECTOR[2] = (empty ->DIRECTOR)
  
```

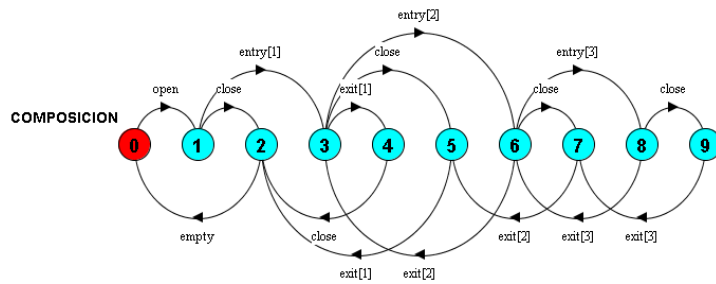
LTS Control**FSP Control**

```

CONTROL = ( open -> (entry[1] -> CONTROL[0] | close -> empty -> CONTROL)),
CONTROL[0] = (close -> CONTROL[1]),
CONTROL[1] = ( empty -> CONTROL ).

```

LTS Composicion



FSP Composicion

```

||COMPOSICION = (ENTRADA || SALIDA || DIRECTOR || CONTROL).

```

1.6. Ejercicio 6

a) FSP de proceso equivalente a S

```

E = ( a->c->b->E | c->a->b->E ).

```

b) Proceso R que restrinja S a trazas a,c,b,a,c,b,...

```

R = (a->c->b->R).

```

c) Proceso T

