

Trabajo Práctico N° 1

Ejercicio 1. Instale LTSA en su directorio local.

Ejercicio 2. Lea los capítulos 1, 2 y 3 de *Concurrency, State Models and Java Programs* (Magee & Kramer 2006).

Ejercicio 3. Para cada uno de los procesos en la Figura 1, dé la descripción en FSP del grafo del sistema de transiciones etiquetadas (LTS). Puede verificar que el proceso que realizó en FSP es correcto generando el correspondiente LTS a través de la herramienta LTSA.

Ejercicio 4. Una variable almacena valores en el rango $0..N$ y soporta las acciones *read* y *write*. Modele la variable como un proceso, denominado *VARIABLE*, usando FSP. Compruebe que, luego de escribir un valor dentro del rango correspondiente, la operación de lectura sólo permita leer el valor escrito. Por ejemplo, para $N = 2$, la siguiente traza es un comportamiento válido del proceso:

write.2 → read.2 → read.2 → write.1 → write.0 → read.0

Ejercicio 5. Una máquina expendedora de bebidas cobra 15c por una lata de gaseosas. La máquina acepta monedas de 5c, 10c, y 20c, y da vuelto. Modele la máquina como un proceso FSP. *Ayuda: La máquina expendedora debería aceptar monedas hasta que alcance el costo de la lata y luego dar la lata de gaseosa y el cambio.*

Ejercicio 6. Implemente en Java el clásico programa que imprime el mensaje “Hello World” mediante la creación de un thread.

Ejercicio 7. Implemente en Java un programa compuesto por dos threads (independientes), los cuales imprimen los mensajes “ping” y “pong” k veces (k es un parámetro del programa), cada uno con un retraso (delay) diferente entre un mensaje y el siguiente.

Ejercicio 8. Implemente en Java un programa que cuente cuántos números primos hay desde 1 hasta un número dado. Puede utilizar la siguiente función que determina si un número es primo:

```
private static boolean isPrime(int n) {  
    int k = 2;  
    while (k * k <= n && n % k != 0)  
        k++;  
    return n >= 2 && k * k > n;  
}
```

Realice una implementación sin threads, una utilizando 2 threads, es decir, dividiendo el rango en dos y contando concurrentemente en ambos tramos. Finalmente realice una implementación utilizando 5 threads.

Experimente la performance ejecutando el proceso con rangos grandes en las diferentes versiones.

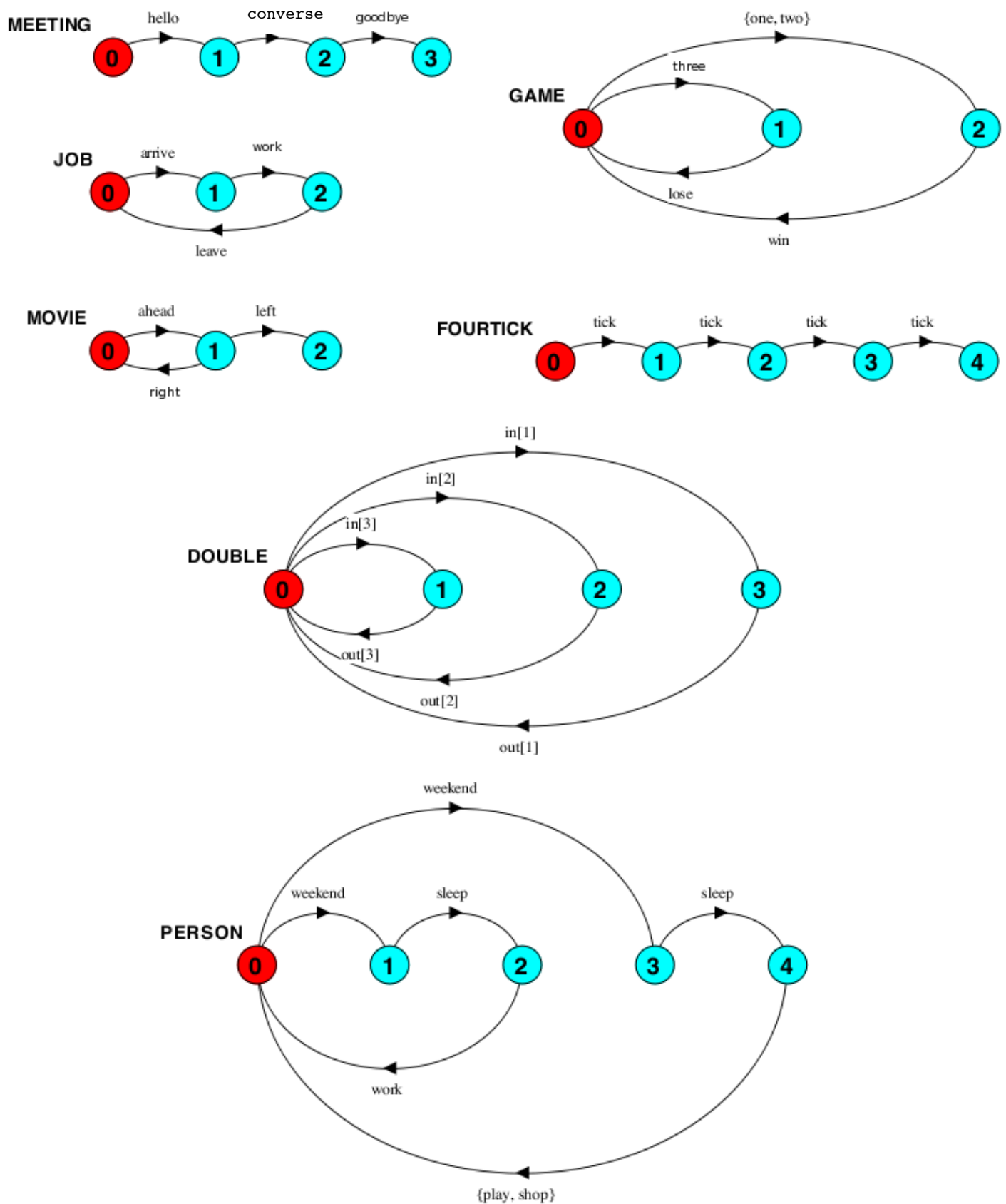


Figura 1: Modelar con FSP