



Taller 1: Máquinas de Estados y Protocolos de Comunicación

Miguel Torrealba S.
mtorrealba@usb.ve

Introducción

- Frecuentemente los *Protocolos de Comunicación* se pueden representar con **Máquinas de Estados Finitos (Autómata)** que se pueden automatizar.
- Una máquina cambia de un estado a otro a través de una *transición* y según se reciben *entradas* de un *alfabeto* preacordado.
- Luego esas máquinas se programan y así se obtiene el *Código de Software* asociado al protocolo.
- Si para cada estado no inicial de un autómata, existe una única transición que lo activa, desde otro estado previo, se dice que éste es un *Autómata Finito Determinista (AFD)*.

McClane y Zeus en problemas



McClane y Zeus en problemas

- El problema de las 2 garrafas de agua y la bomba activa



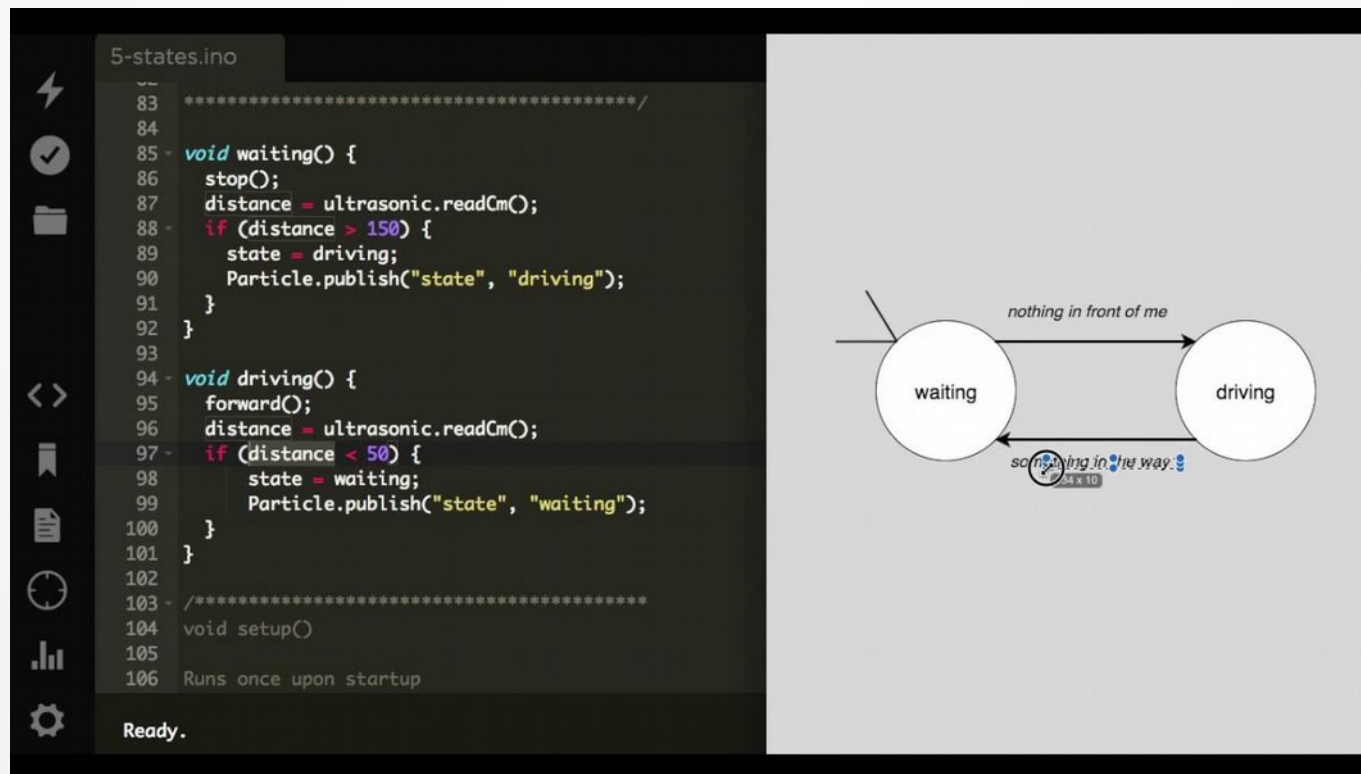
McClane y Zeus en problemas

- Vea el siguiente segmento de la película "Die Hard 3"



McClane y Zeus en problemas

- Simón espera que hayas comprendido el enunciado.
- Simón permitirá que ayudemos a los "los buenos" del filme.
- Simón desea que automaticemos la solución, para que cualquiera pueda resolverlo sin tragedia alguna.

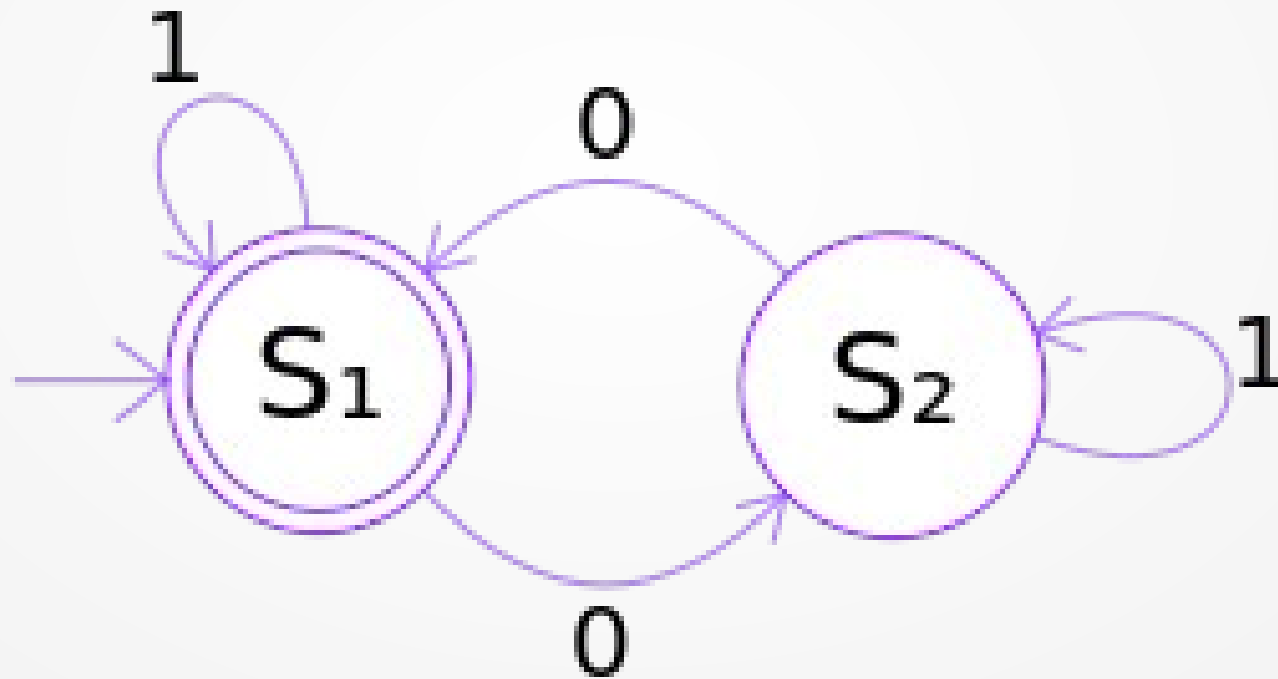


Máquinas de Estado

- Lo primero es determinar si existe solución y en caso de ser afirmativa la respuesta, identificar la que pueda tomar pocas o preferiblemente, la menor cantidad de actividades. El tiempo es un elemento clave del éxito.
- Ahora bien, se deseamos automatizar la solución, es factible construir un Autómata. Este deberá ir desde el estado de inicio, hasta donde se llega a disponer de una jarra con 4 litros y se puede colocar en el peso, a efectos de detener la bomba.
- No se puede usar una pesa para comprobar lo que se ha llenado, por lo que hay que instruir a nuestro sistema para que no desborde los límites, ni calcule mal los 4 litros.

Máquinas de Estado

- Ejemplo de un AFD.



Solución

- Sea **(JG, jp)** la notación de un *estado* cualquiera con dos *variables* sobre nuestro sistema, donde **JG** representa la Jarra Grande y **jp** indica la jarra pequeña.
- Se tiene además las siguientes características:
 - .- $JG \in N \ / \ 0 \leq JG \leq 5$
 - .- $jp \in N \ / \ 0 \leq jp \leq 3$
- Se definen también 4 operaciones de transición: llenar, vaciar, transvasar y añadir.

Solución

- La especificación paramétrica de las operaciones son:

llenar ($variable_i, m$) \rightarrow ($variable_1, variable_2$) donde $m \in \mathbb{Z}$ y el subíndice i es 1 o es 2

vaciar ($variable_i, lim$) \rightarrow ($variable_1, variable_2$) donde lim es 3 o es 5 y el subíndice i es 1 o es 2

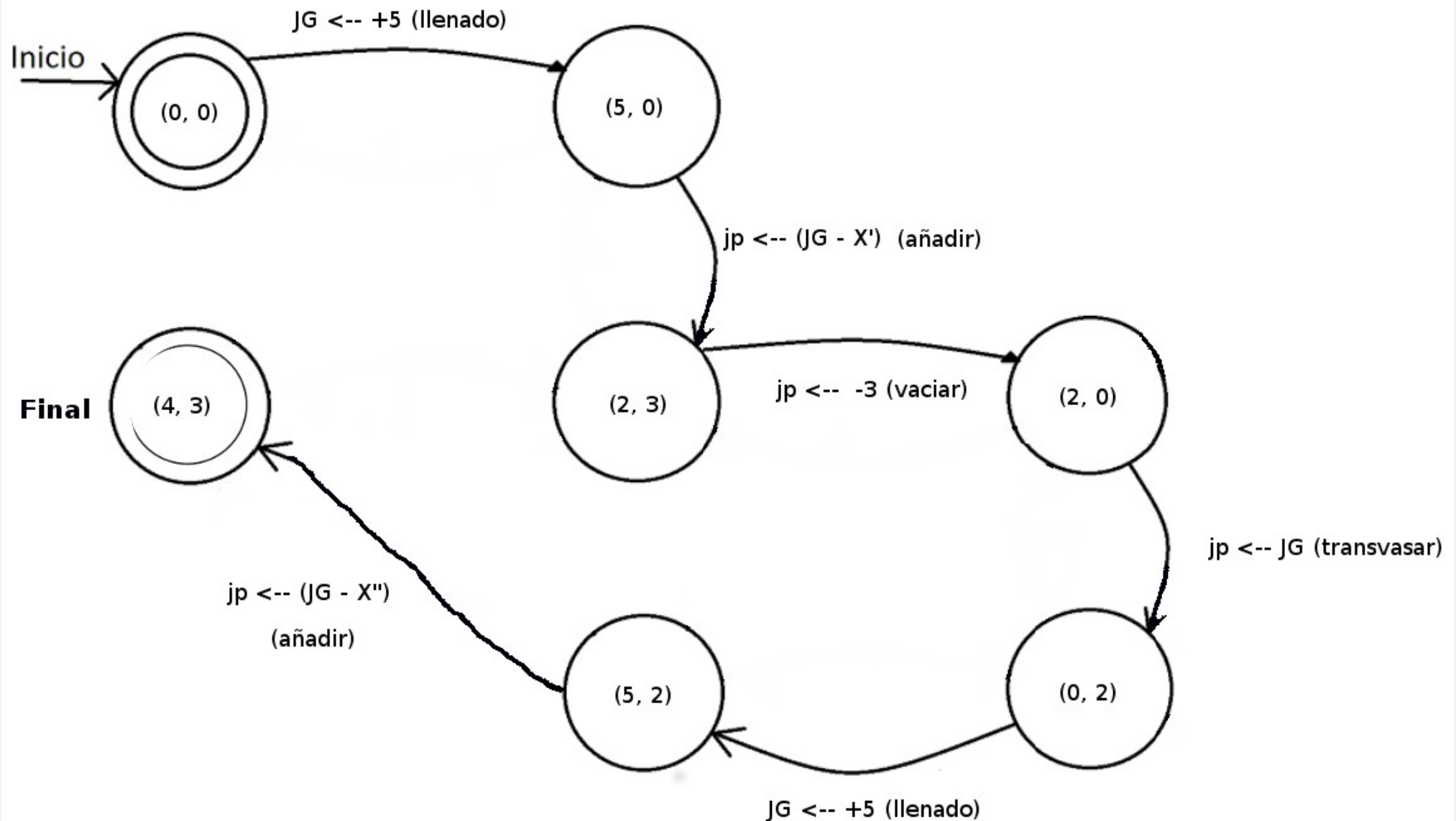
transvasar ($variable_i, variable_j$) \rightarrow ($variable_1, variable_2$) donde los subíndices i, j son 1 o 2 y se cumple además que $i \neq j$

añadir ($variable_i, variable_j, lim$) \rightarrow ($variable_1, variable_2$) donde lim es 3 o es 5 y el subíndice i es 1 o es 2 tal que se cumple además que $i \neq j$

Solución

- Con relación a todo ese diseño, la solución al problema la podemos ilustrar, gráficamente y como un AFD, de esta manera ...

Solución



X' y X'' pertenecen al conjunto R

Solución



Solución

- Hay una solución matemática para este problema, que se puede describir con una *expresión algebraica* y es interesante, ya que también ofrece una respuesta programable, pero en nuestro caso nos interesa el enfoque orientado a los AFD.

Objective: Students will add, subtract, multiply, and divide integers.

	One Solution	Identity (Infinite Solutions)	No Solution
Examples	$x = 2$ $x = 5$ $x = 3$ $x = \frac{1}{4}$ $x = 71$	$8 = 8$ $-13 = -13$ $9 = 9$ $\pi = \pi$ $0 = 0$ $14 = 14$	$25 \neq 3$ $14 \neq 87$ $225 \neq 0$ $-7 \neq 82$ $-1 \neq 1$ $-\frac{2}{3} \neq \frac{1}{4}$

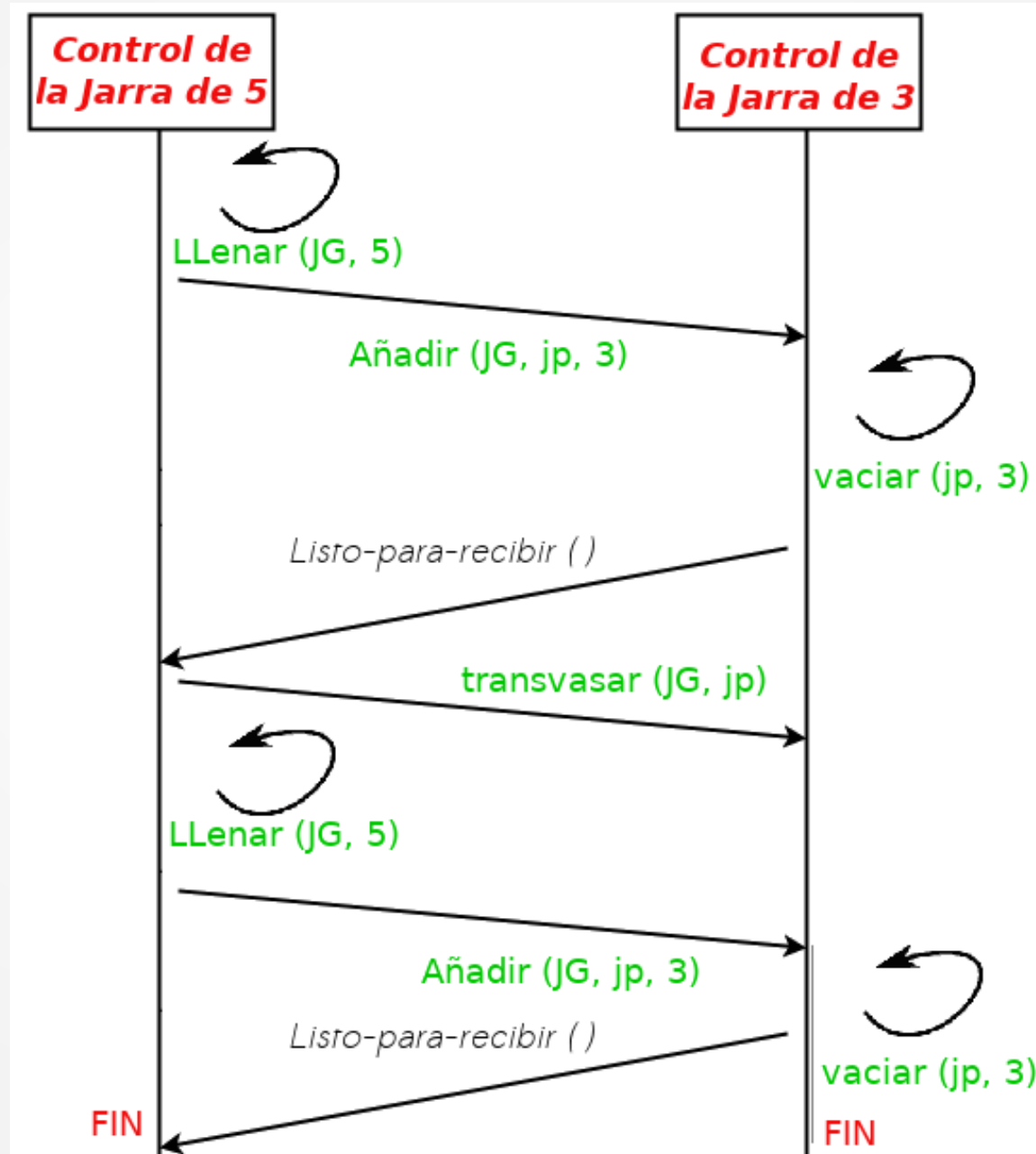
Solución

- **Trabajo práctico:** diseñe un sistema similar, pero que opere con dos computadoras. Es decir, dos autómatas.

Solución



Solución



Solución

- La solución planteada no es óptima y tiene algunos elementos que no cuadran homogéneamente, por ejemplo, ese el caso de la función "*listo-para-recibir*" que constituye más un modo de emitir una señal de control, que uno de funcionabilidad propia de las reglas operativas.
- Aún así, esto funciona y nos trae a la mente la confusión de algunos protocolos de comunicación de décadas atrás, que en su especificación no separaban el *tráfico de datos* del *tráfico de control* y tampoco lo hacían con el *tráfico de gestión*.

Solución

- La solución planteada tampoco contiene elementos que le permitan superar *fallas* y *accidentes*.
- Para ser eficaz y efectivo en el mundo moderno y real, un diseño debe contemplar situaciones problemáticas y restaurar la operatividad al menor coste posible.



Solución

- **Trabajo práctico:** programe los 2 autómatas descritos previamente en *lenguaje C/C++* y entregue el resultado, siguiendo las indicaciones que se le den y dentro del plazo que su profesor le indique.