

Agustín Rodríguez Esteva

Introducción al Control industrial – Curso 2020

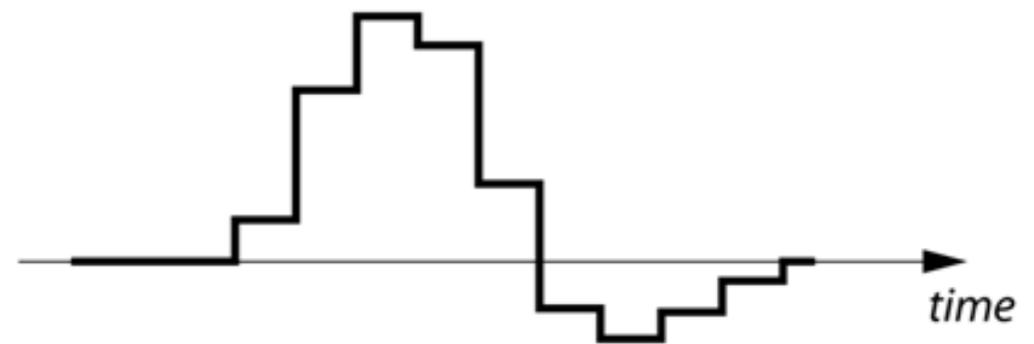
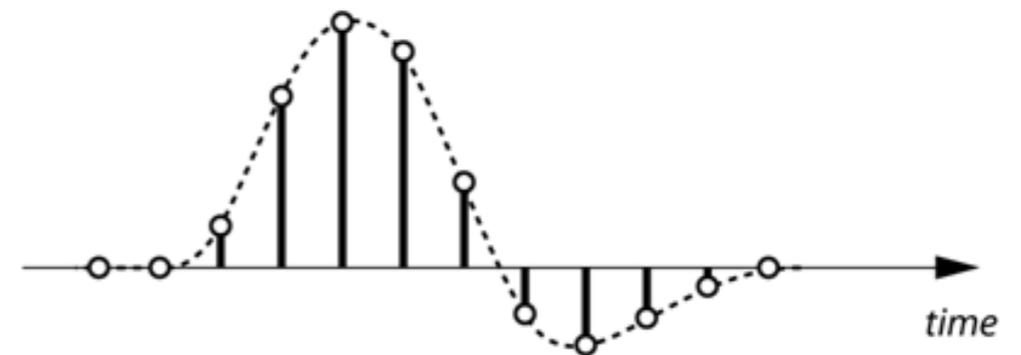
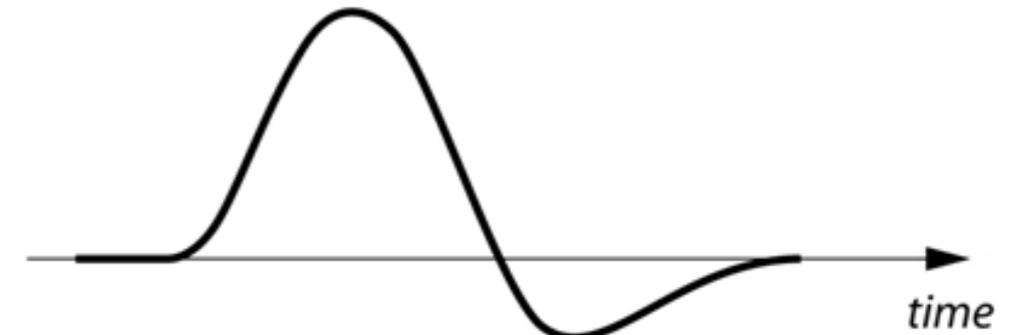
INTRODUCCIÓN A PROCESOS DISCRETOS

▼ Definición

- Sistemas dinámicos
- Los estados que los representan toman valores discretos
- La evolución en el tiempo es también discreta y está determinada por eventos

➤ Elementos

➤ Herramientas de análisis

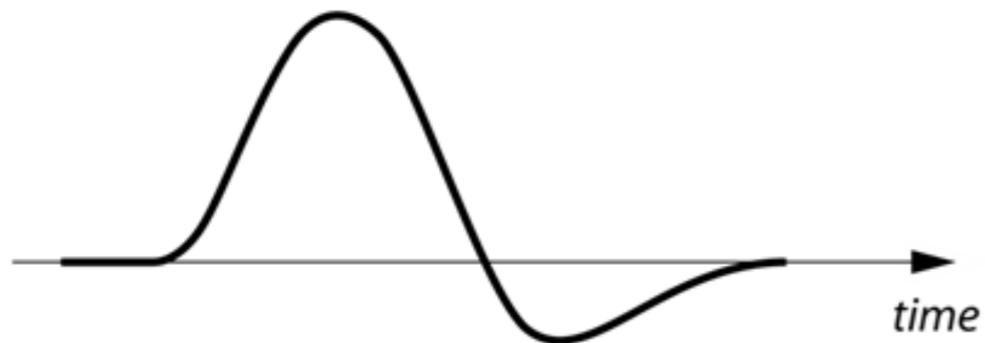


COMPARACIÓN SISTEMA CONTINUO/DISCRETO

Sistema continuo
(lineal)

$$X' = A.X(t) + B.U(t)$$

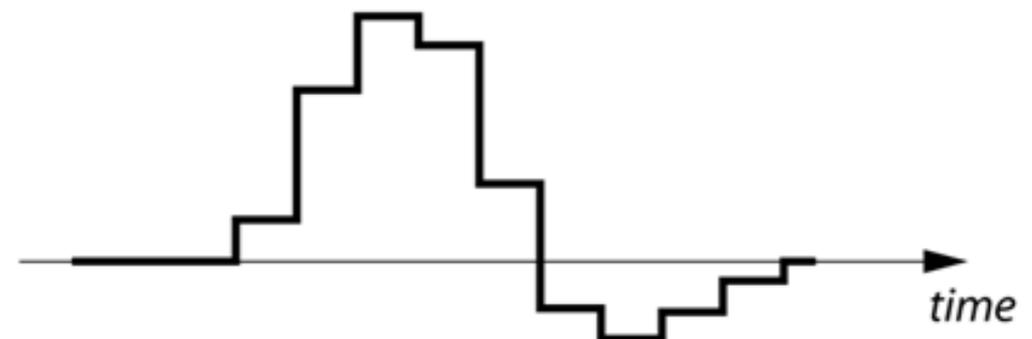
$$Y = C.X(t) + D.U(t)$$



Sistema discreto

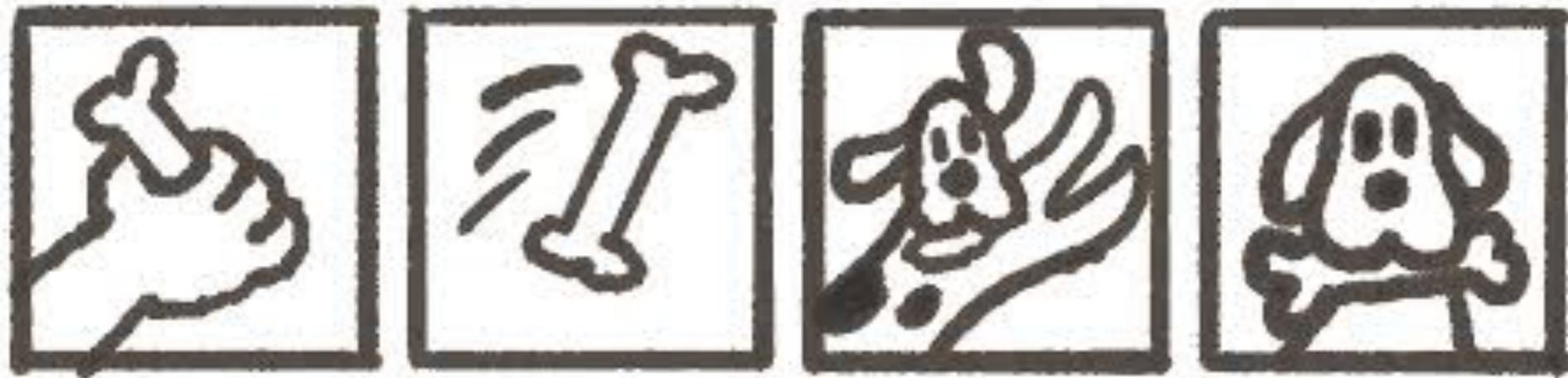
$$X(k+1) = P(X(k), U(k))$$

$$Y(k) = C(X(k)) + D(X(k))$$



¿UN PROCESO FÍSICO ES DISCRETO?

“Un proceso físico es esencialmente continuo, pero según sea el propósito del modelo, puede representarse tanto como un sistema continuo o como un sistema discreto”



EJEMPLO FÍSICO

Variables discretas

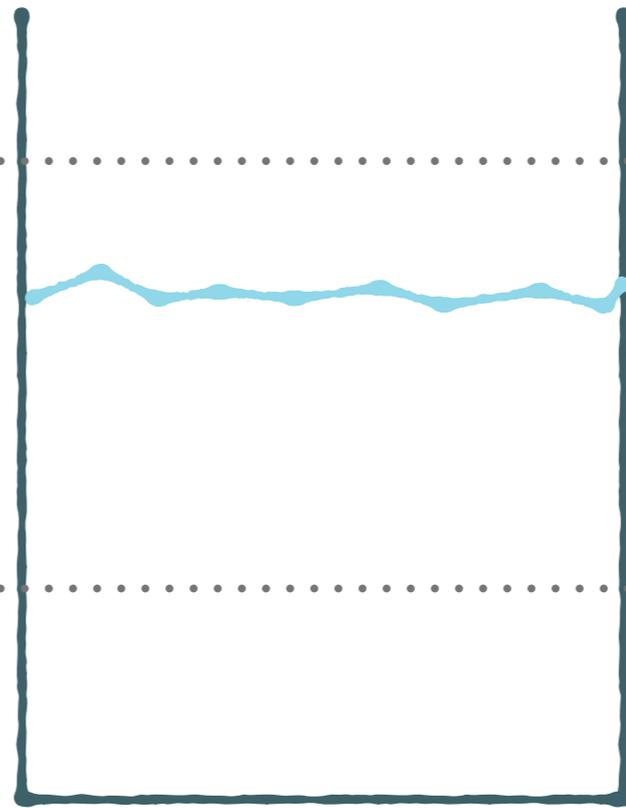
Proceso

Variables continuas

Lleno

Normal

Vacío



H

CONTROLADORES LÓGICOS

- Son en sí mismos procesos discretos
- Sus características fundamentales:
 - Tienen como entradas señales lógicas (señales discretas de 2 estados posibles, 0 y 1)
 - Generan salidas de señales lógicas
 - Los estados internos son lógicos
- Operan sobre variables según las reglas del álgebra de Boole

ÁLGEBRA DE BOOLE

El álgebra de Boole está basada en 3 operadores básicos:

- NOT ($\text{NOT } a = \tilde{a}$)
- AND ($a \text{ AND } b = a.b$)
- OR ($a \text{ OR } b = a+b$)

Se utiliza este orden como orden de precedencia al operar.

ÁLGEBRA DE BOOLE

$$S = \text{NOT } a = \bar{a}$$

a	S
0	1
1	0

$$S = a \text{ AND } b = a.b$$

a	b	S
1	1	1
1	0	0
0	1	0
0	0	0

$$S = a \text{ OR } b = a + b$$

a	b	S
1	1	1
1	0	1
0	1	1
0	0	0

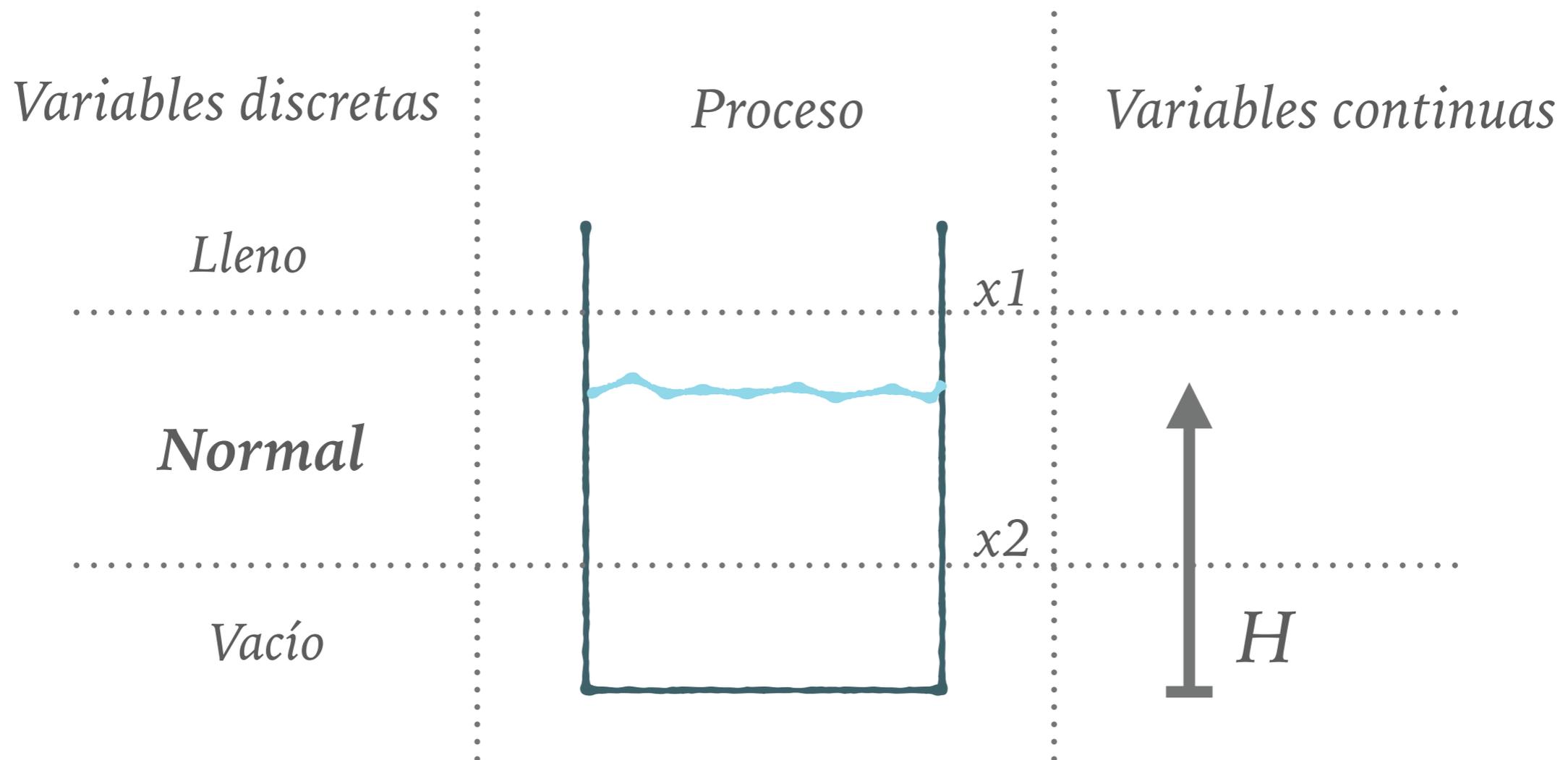


CODIFICACIÓN

¿Cómo representar los estados discretos con variables lógicas cuando tenemos más de 2 estados discretos ?

Se utilizan tantas variables lógicas como se necesiten tal que agrupadas en un número binario, este cubra el rango de estados discretos posibles del sistema.

EJEMPLO FÍSICO: CODIFICACIÓN



Consideramos las variables lógicas $X1$ y $X2$

$X1 = 1$ (verdadero) cuando L está por encima de la cota $x1$

$X2 = 1$ (verdadero) cuando L está por encima de la cota $x2$

EJEMPLO: CODIFICACIÓN

El número binario codificado que representa los distintos estados discretos posibles de la variable nivel (N) es :

$$N = X1X2$$

X1	X2	Estado
1	1	LLENO
0	1	NORMAL
1	0	ERROR
0	0	VACÍO

Variables lógicas:

$$\text{LLENO} = X1.X2$$

$$\text{NORMAL} = \hat{X}1.X2$$

$$\text{VACÍO} = \hat{X}1.\hat{X}2$$

ALGEBRA DE BOOLE: PROPIEDADES

Propiedad conmutativa ➤ $A \cdot B = B \cdot A$

➤ $A + B = B + A$

Propiedad asociativa * $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

* $(A + B) + C = A + (B + C)$

Propiedad distributiva ❖ $A \cdot (B + C) = A \cdot B + (A \cdot C)$

❖ $A + B \cdot C = (A + B) \cdot (A \cdot C)$

Teorema de Morgan ★ $\overline{A \cdot B} = \overline{A} + \overline{B}$

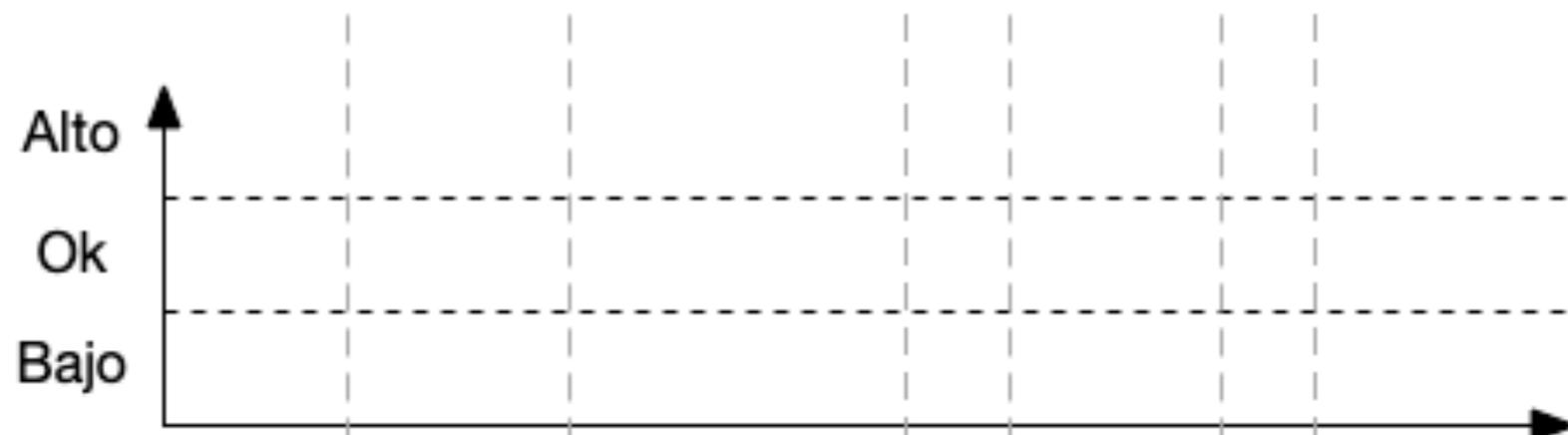
★ $\overline{A + B} = \overline{A} \cdot \overline{B}$

CONDICIÓN Y EVENTO

- Condición se refiere al estado de una variable lógica que se extiende durante cierto tiempo
- Evento se refiere al momento del cambio de estado de una condición y se distingue el sentido del cambio
- Un evento puede ser el cambio de una entrada o de cualquier función lógica de variables internas o externas

CONDICIÓN Y EVENTO

- Condición se refiere al estado de una variable lógica que se extiende durante cierto tiempo
- Evento se refiere al momento del cambio de estado de una condición y se distingue el sentido del cambio
- Un evento puede ser el cambio de una entrada o de cualquier función lógica de variables internas o externas



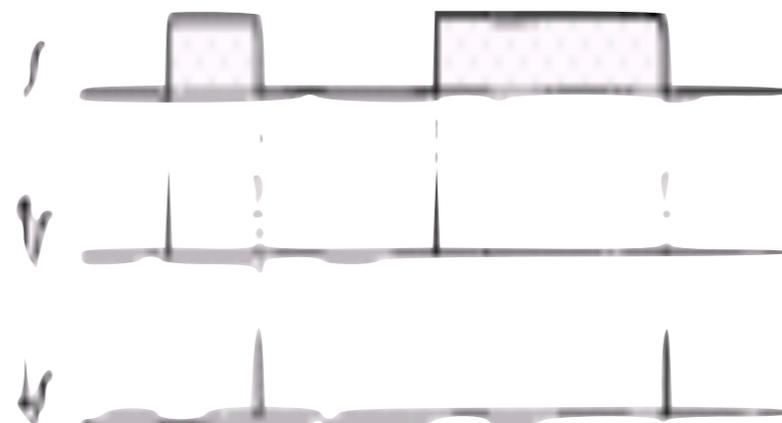
ALGEBRA DE EVENTOS

Notación

$f(a_1, a_2, \dots, a_N)$ - función booleana

$\uparrow f$ = cambio de 0 a 1

$\downarrow f$ = cambio de 1 a 0



Precedencia

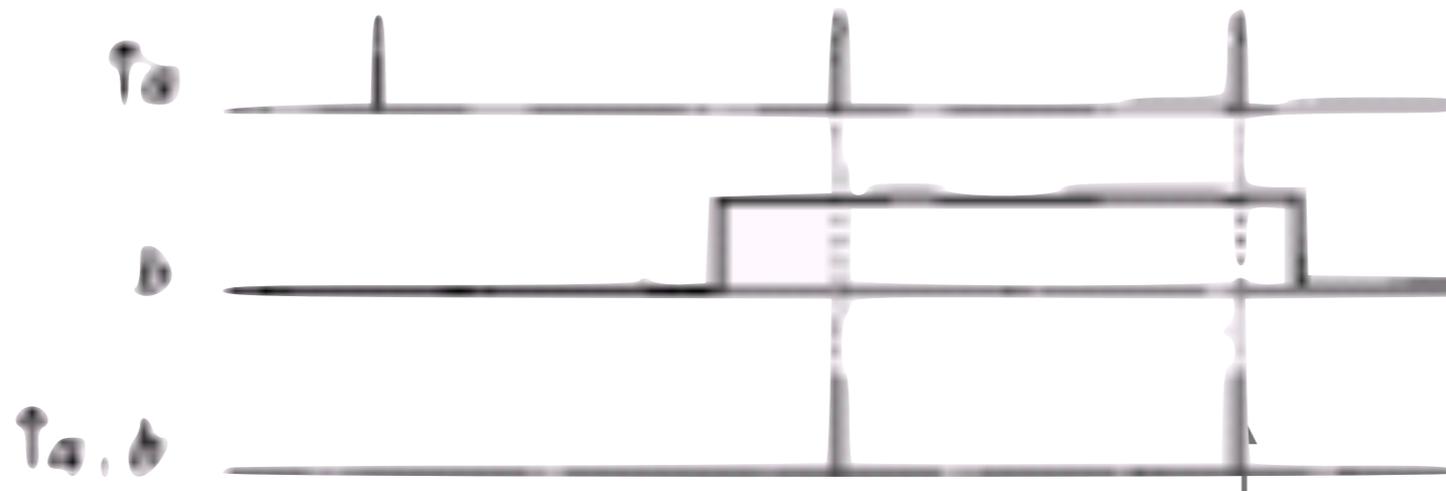
El símbolo “ \uparrow ” tiene precedencia sobre los operadores “ \cdot ” (AND) y “ $+$ ” (OR).

$$\uparrow a \cdot b = (\uparrow a) \cdot b$$

$$\uparrow a + b = (\uparrow a) + b$$

ALGEBRA DE EVENTOS

$\uparrow a.b$ = evento en el mismo instante que $\uparrow a$ cuando $b=1$



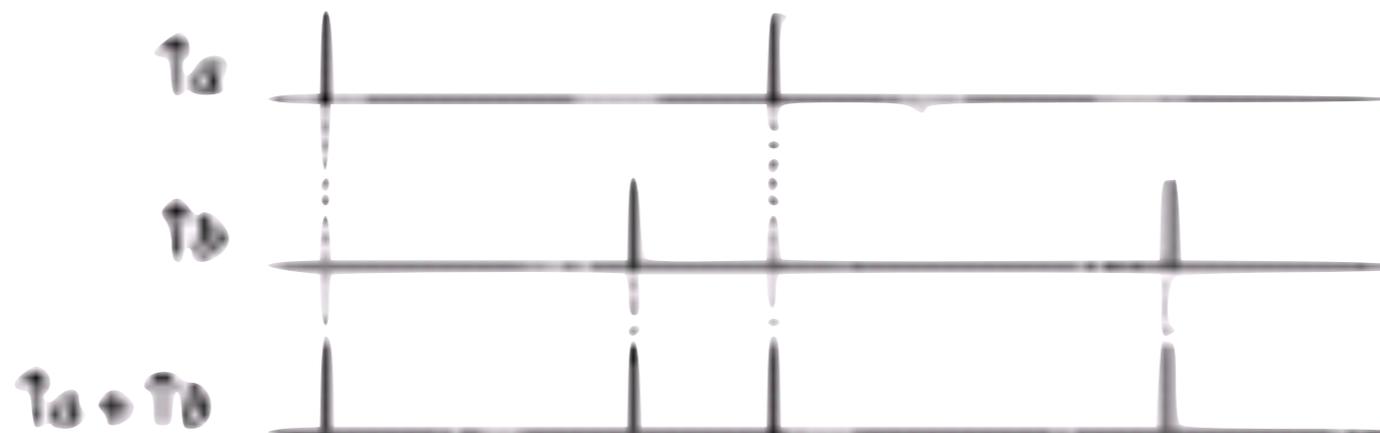
$\uparrow a.\uparrow b$ = evento que ocurre cuando $\uparrow a$ y $\uparrow b$ se dan al mismo tiempo



ALGEBRA DE EVENTOS

Definiciones

$\uparrow a + \uparrow b =$ evento que ocurre cada vez que se da $\uparrow a$ o $\uparrow b$



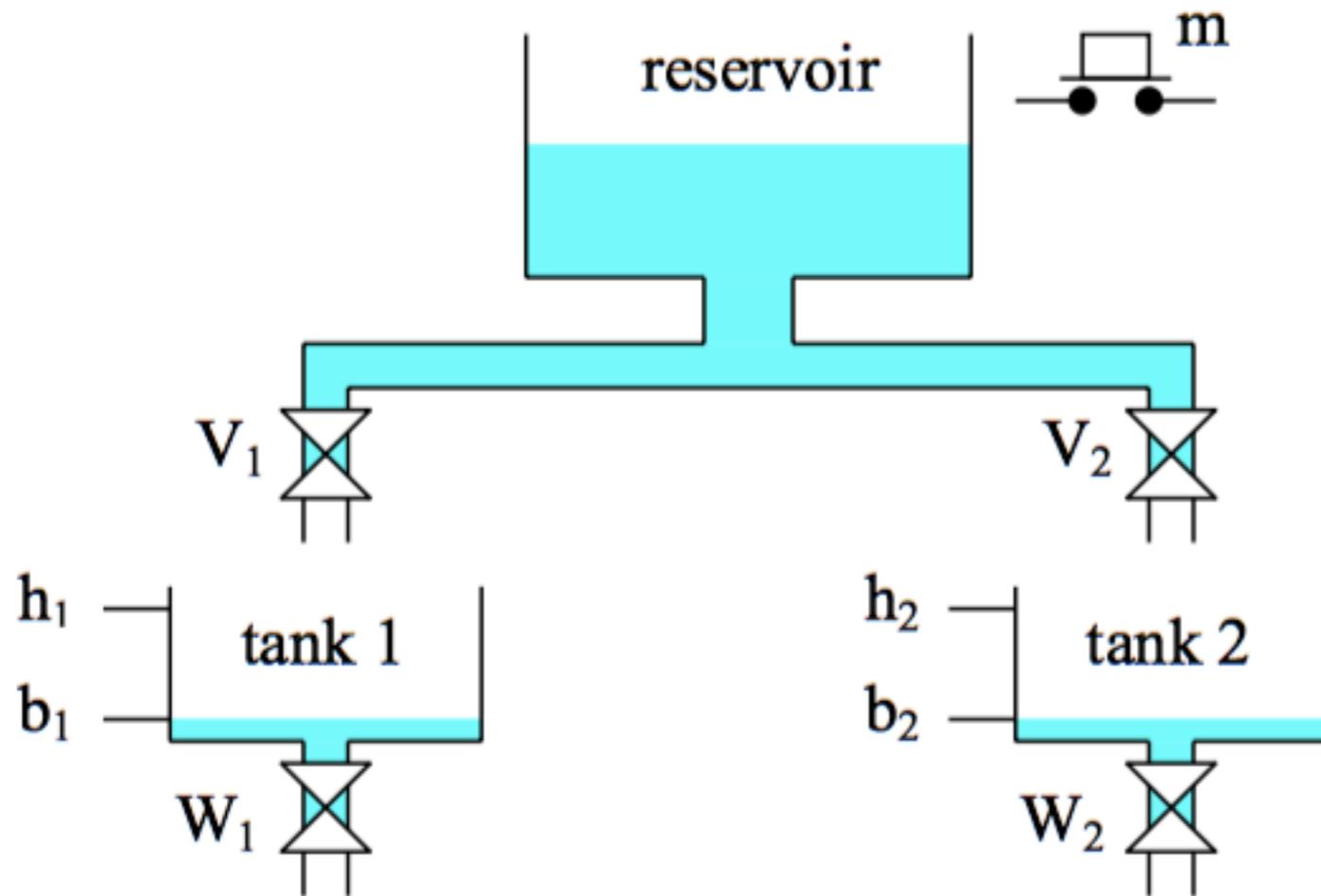
HERRAMIENTAS DE ANÁLISIS

Se requiera que estas herramientas permitan:

- Describir la evolución de los estados
- Representar al sistema en forma compacta
- Entender claramente el comportamiento entrada-salida del controlador lógico
- Tener en cuenta la concurrencia (sistemas parcialmente independientes)

HERRAMIENTAS DE ANÁLISIS

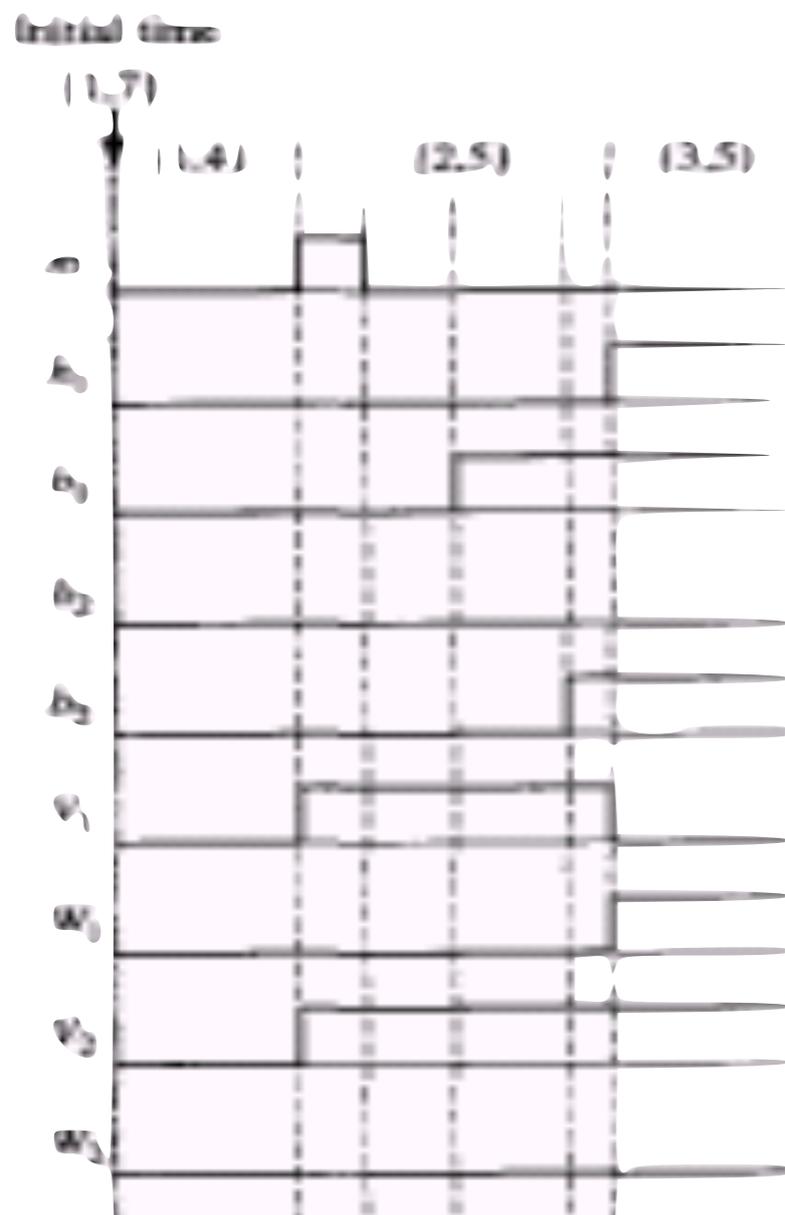
Ejemplo de sistema con concurrencia



En el estado inicial, ambos tanques están vacíos. Al presionar m ambos tanques comienzan a llenarse a través de V_1 y V_2 . Cuando se llena un tanque se cierra la válvula correspondiente y se comienza a vaciar. Una vez vacíos si se presiona m se comienza nuevamente el ciclo

HERRAMIENTAS DE ANÁLISIS

Process Timing Diagram



Este diagrama muestra la evolución en el tiempo de las salidas del controlador para una cierta combinación de las entradas

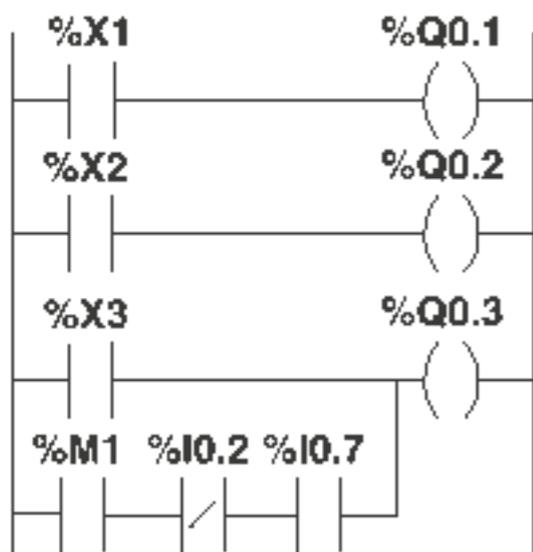
HERRAMIENTAS DE ANÁLISIS

Diagrama de estados

- Cada estado se representa con un número o referencia dentro de un círculo.
- Cada estado tiene “salidas” encendidas/apagadas durante la permanencia en él.
- Cada transición se indica con una combinación de entradas.

POR QUÉ HOY HABLAMOS DE GRAFCET?

- Lenguajes de programación según norma IEC para PLC:
 - Ladder LD
 - Diagrama de bloque de funciones FBD
 - Texto estructurado ST
 - Lista de instrucción IL
 - **Bloques de funciones secuenciales SFC**



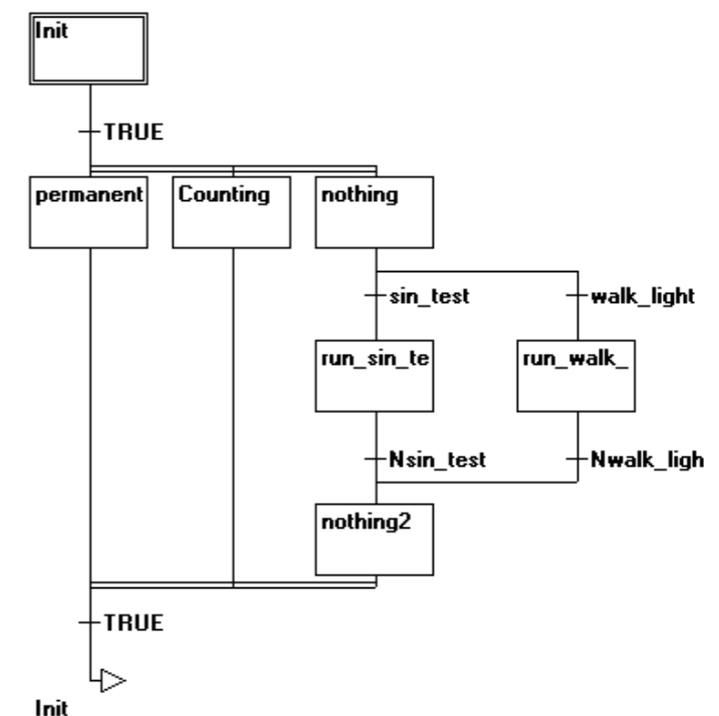
```
Program Main
  Var T : TagWriteValue; End_Var
  Var DoReq : Bool; Index : INT; End_Var

  if Index < 11 Then

    if T.DONE Then
      (* Move to the next request *)
      Index := Index + 1;
      DoReq := False;
    else
      (* Start the next request *)
      DoReq := True;
    end_if;

    T( REQ:=DoReq,
      TAGNAME:=TagNames[ Index ],
      Value:=TagValues[ Index ] );
  end_if

end_Program
```



SEQUENTIAL FUNCTION CHART (SFC, GRAFCET)

Definiciones

Pasos: se representan con un cuadrado numerado en el diagrama

Estados: están determinados por la combinación de pasos activos en el mismo momento

Salidas: se representan con un rectángulo asociado a un paso con el nombre de la salida activada

SEQUENTIAL FUNCTION CHART (SFC, GRAFCET)

Transiciones

- Entre 2 pasos siempre existe una transición
- Se nota con una barra horizontal
- Se numera del lado izquierdo y se hace referencia a la receptividad asociada del lado derecho

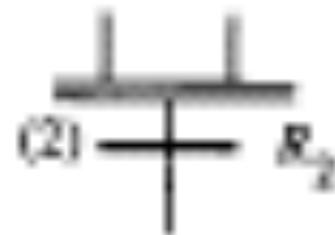
SEQUENTIAL FUNCTION CHART (SFC, GRAFCET)

- Entre dos pasos siempre existe una transición

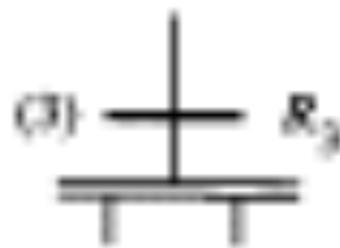
Simple



Junction AND



Distribution AND



Junction and distribution AND



Simple



Junction OR



Distribution OR



Are oriented from bottom to top



RECEPTIVIDAD

Es el evento que permite el “disparo” de la transición (evolución de un paso a otro) y puede ser cualquier función lógica formada por eventos y condiciones.

- Una transición es “disparable” cuando se cumple que:
 - Todos los Pasos precedentes están activos
 - La receptividad de la transición es verdadera

SEQUENTIAL FUNCTION CHART (SFC, GRAFCET)

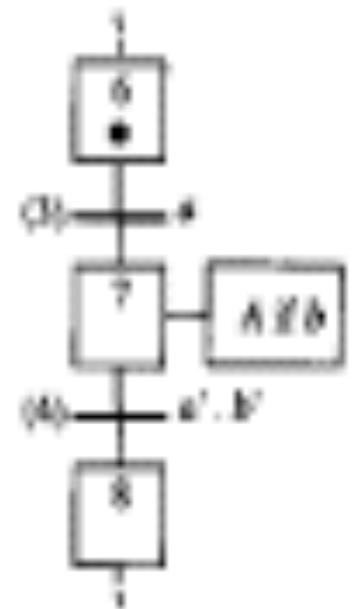
Reglas

- Todas las transiciones disparables son disparadas inmediatamente
- Todas las transiciones disparables al mismo tiempo se disparan en forma simultánea
- Cuando un Paso es activado y desactivado en forma simultánea, permanece activo

SEQUENTIAL FUNCTION CHART (SFC, GRAFCET)

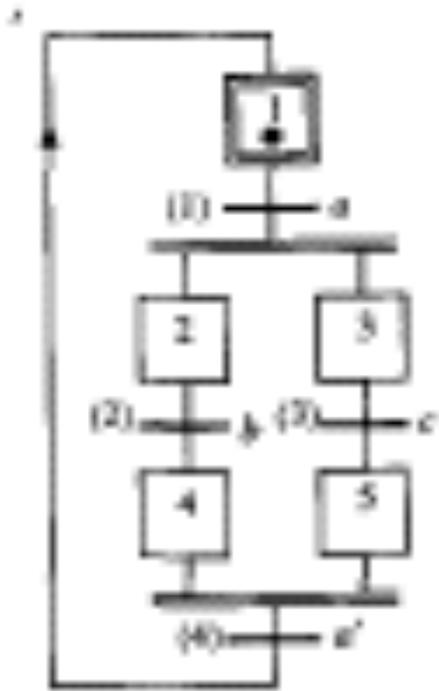
Salidas

- Cada Paso puede tener asociado salidas que son activadas cuando el Paso esta activado
- Se distinguen 3 tipos de salida:
 - De nivel
 - De pulso (se marcan con un asterisco)
 - Condicionales



SEQUENTIAL FUNCTION CHART (SFC, GRAFCET)

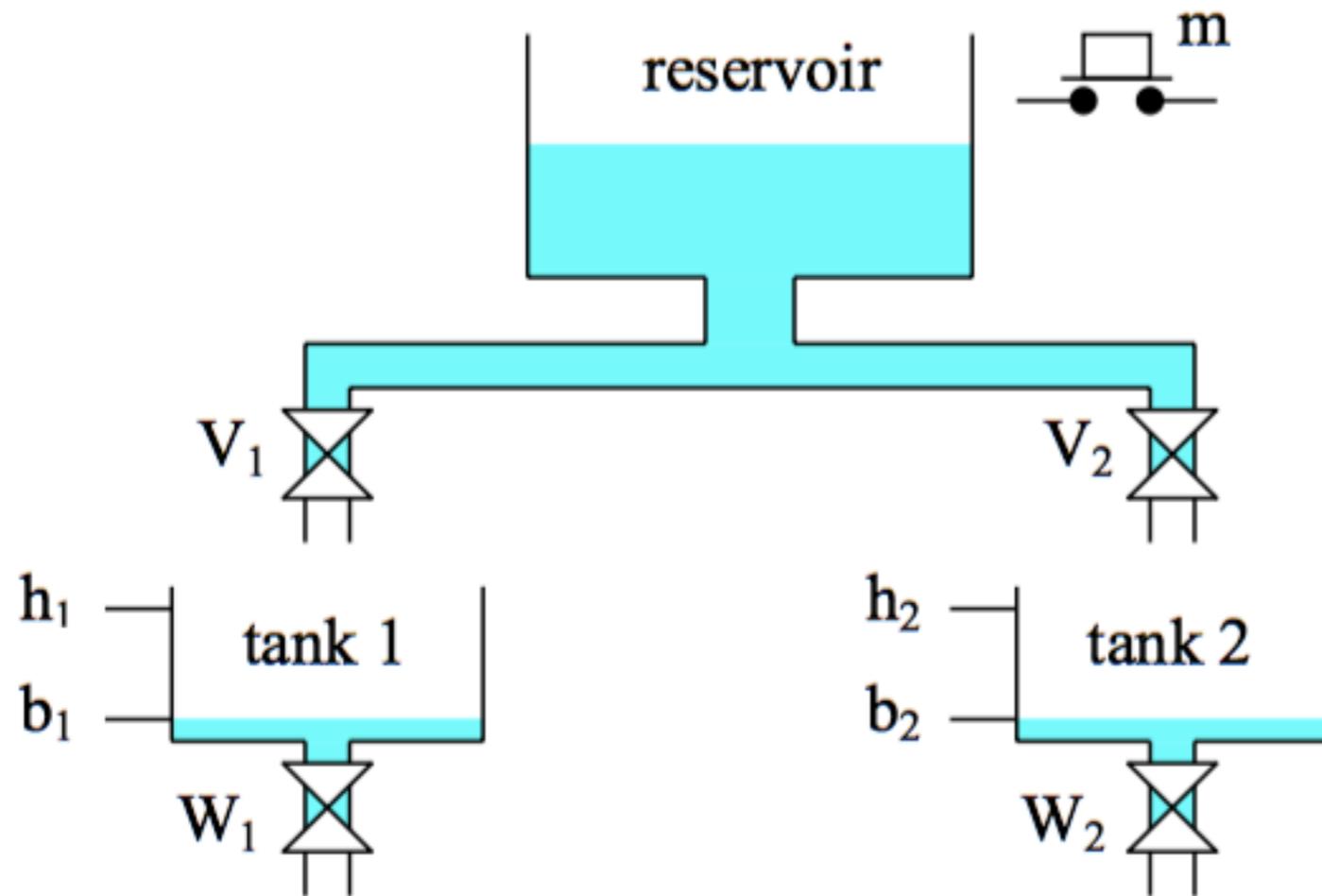
Concurrencia y sincronización



- Situación (Estado) inicial {1}
- Estado {2,3} se alcanza al disparar (1)
- Desde este momento existe una concurrencia entre las ramas (P2,Tr(2),P4) y (P3,Tr(3),P5)
- La transición (4) corresponde a una sincronización porque los pasos 4 y 5 deben estar activos para que pueda ser disparada

HERRAMIENTAS DE ANÁLISIS

Ejemplo de sistema con concurrencia



En el estado inicial, ambos tanques están vacíos. Al presionar m ambos tanques comienzan a llenarse a través de V_1 y V_2 . Cuando se llena un tanque se cierra la válvula correspondiente y se comienza a vaciar. Una vez vacíos si se presiona m se comienza nuevamente el ciclo

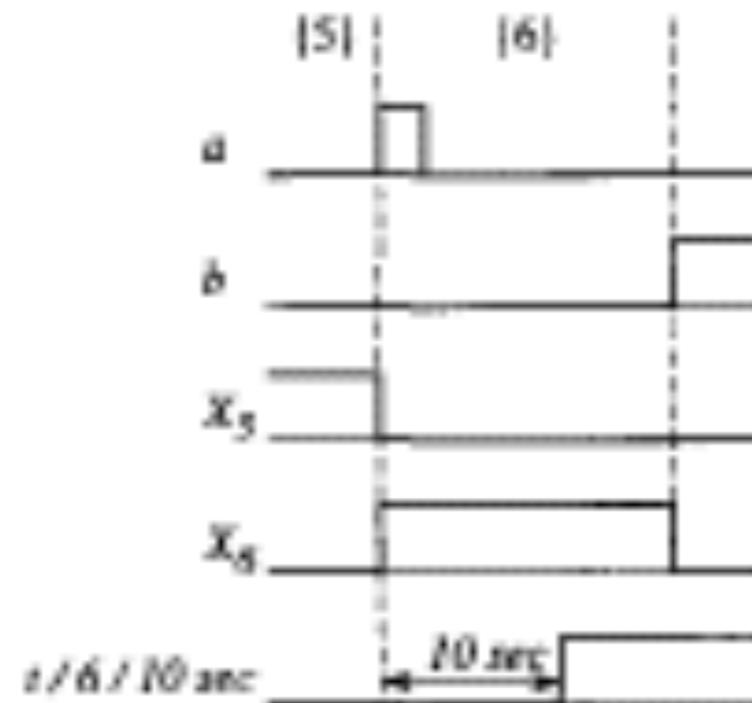
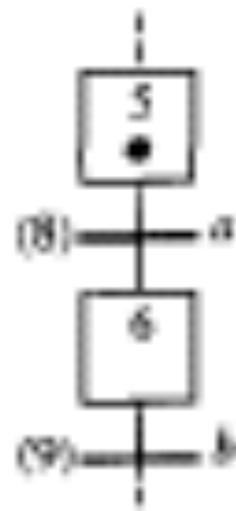
ESTADOS INTERNOS Y TIEMPOS

► Definición:

La variable X_i es una variable lógica la cual es 1 cuando el Paso i está activo

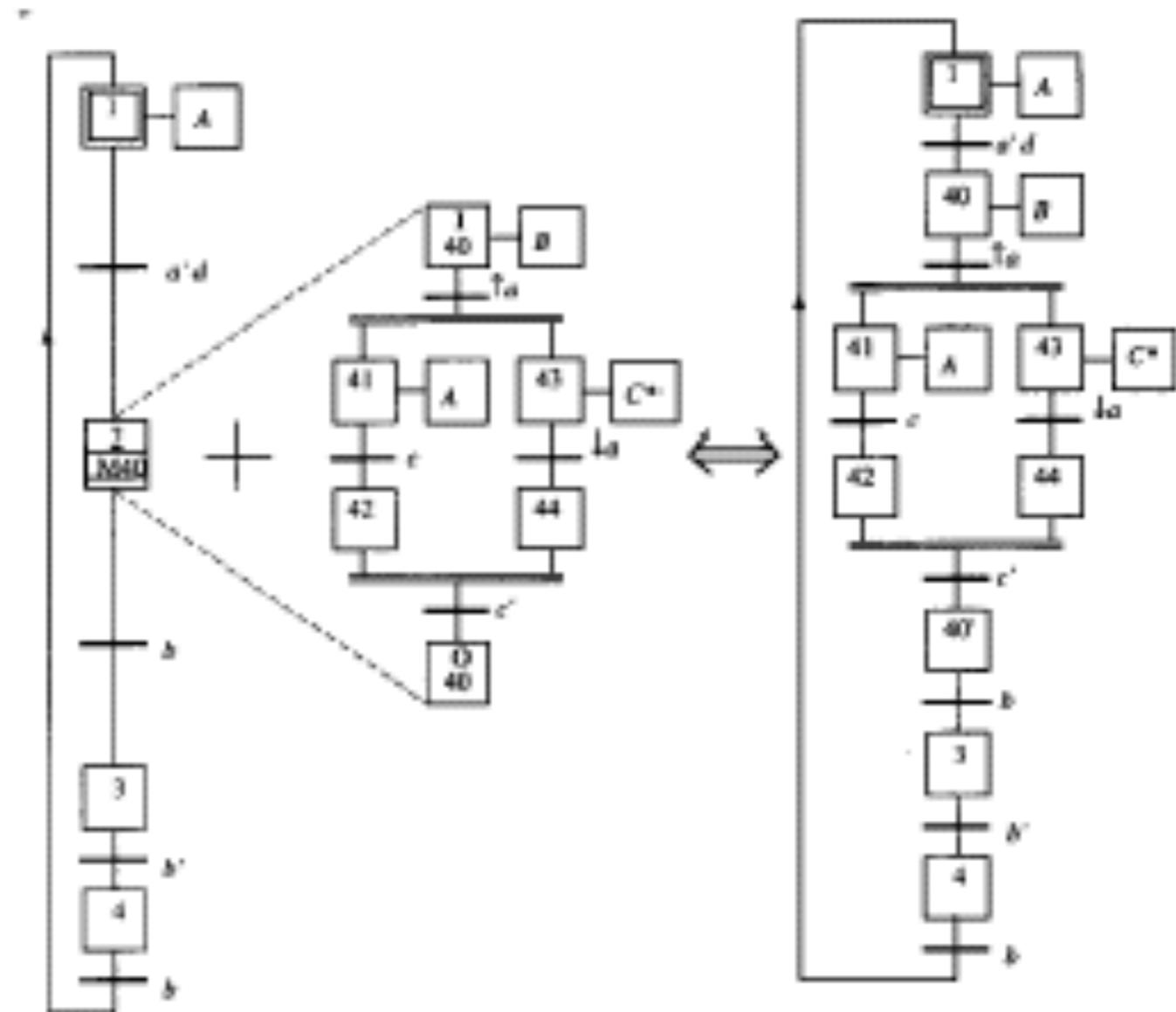
► Definición:

La variable $t / i / \tau$ es una variable lógica que es igual a 1 si un tiempo τ ha transcurrido desde la última vez que el Paso i pasó de inactivo a activo

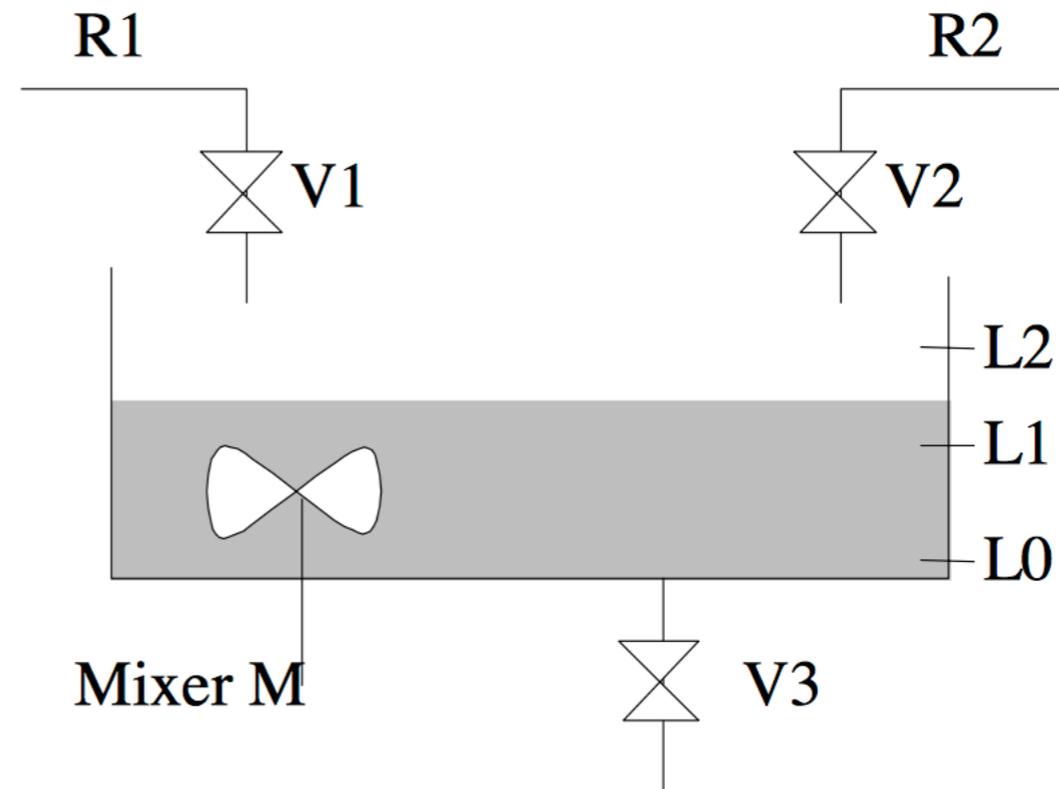


MACRO-STEP (O MACRO-PASO)

- Un Macro Paso tiene siempre un solo Paso de entrada (nominado I) y un solo Paso de salida (nominado O)
- El disparo de la transición aguas arriba del Macro Paso activa el Paso I
- El Paso O habilita las transiciones aguas abajo del Macro Paso
- Ninguna otra conexión intermedia entra o sale del Macro Paso



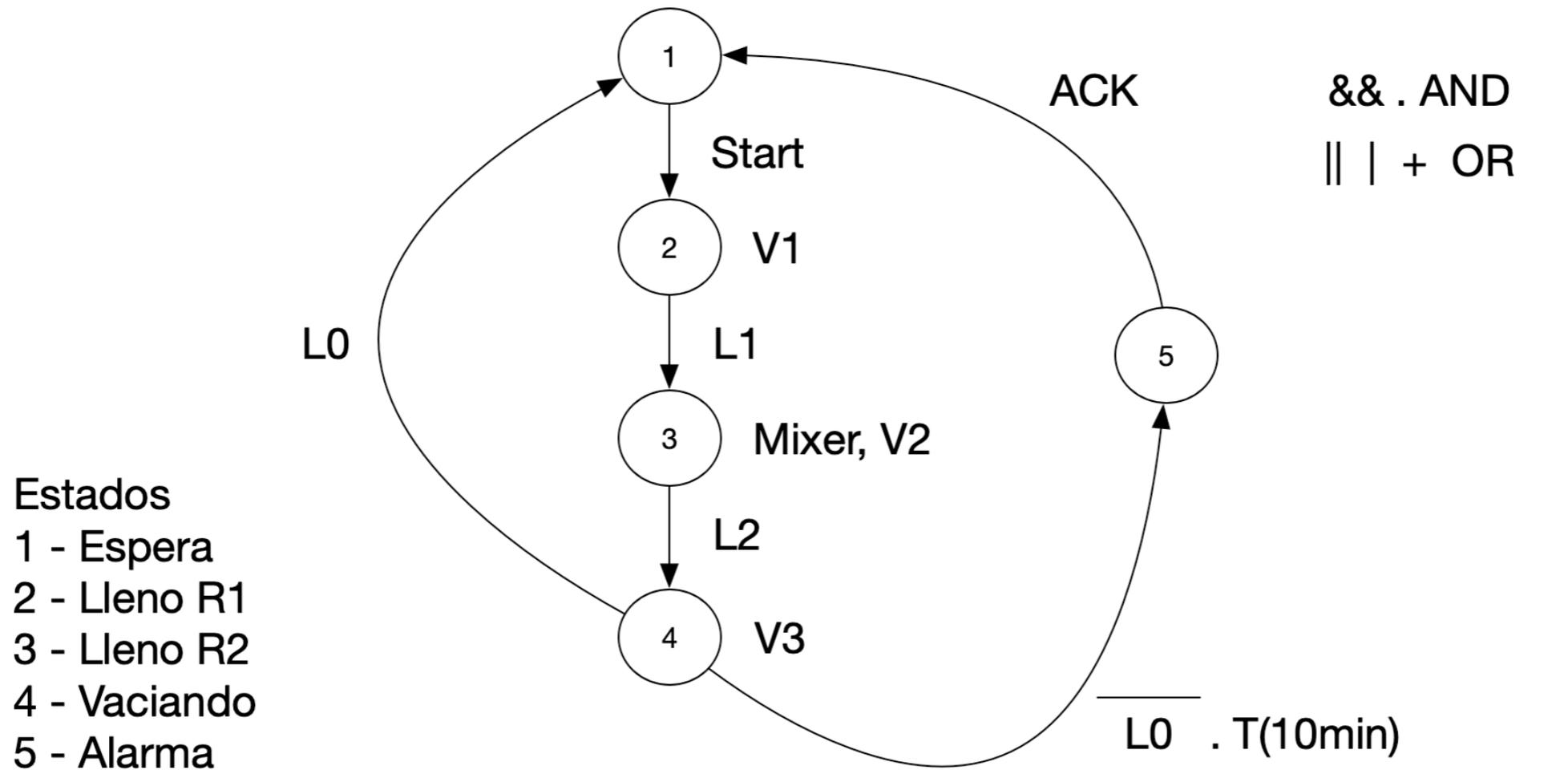
EJEMPLO DE PROGRAMACIÓN CON GRAFCET (SFC)



Pedido del cliente

- 1) Cuando presione Start, abrir V1 hasta el nivel L1.
- 2) Cuando alcanzo L1, enciendo el mixer, cierro V1 y abro V2.
- 3) Cuando alcanzo L2, apago el mixer, cierro V2 y abro V3.
- 4*) Si luego de 10 minutos no se alcanza el nivel L0, lanzo alarma hasta que se reconozca la misma (acknowledge: ACK). El proceso no se puede reiniciar hasta que la misma fue reconocida.

SOLUCIÓN DE EJEMPLO DE PROGRAMACIÓN EN DIAGRAMA DE ESTADOS



SEQUENTIAL FUNCTION CHART (SFC, GRAFCET)

Referencias

Grafcet : A powerful tool for specification of Logic controllers

René David

Grafcet: an Analysis Pattern for Event Driven Real-time Systems

Nathalie Gaertner & Bernard Thirion