

# AUTÓMATAS PROGRAMABLES

## LENGUAJES DE PROGRAMACIÓN DE AUTÓMATAS

# ÍNDICE

- 
- Introducción a la programación del autómeta
    - Etapas en la programación del autómeta.
    - Definición del sistema de control
    - Definición de las variables del modelo de control
  - Lenguajes de programación

# Introducción a la programación del autómeta

Podríamos dividir la programación del autómeta en varias etapas:

1. Definir el sistema de control (qué debe hacer, en qué orden, etc.):

- una descripción literal
- o un diagrama GRAFCET.

Consiste en representar el sistema de control mediante un modelo, indicando todas las funciones que intervienen, las relaciones entre ellas, y la secuencia que deben seguir.

3. Identificar las señales de entrada y salida del autómeta.

- Tipo digital, analógicas, número de E/S

3. Asignar las direcciones de entrada/salida o internas del autómeta a las correspondientes del modelo.

4. Codificar la representación del modelo. Lenguaje de programación.

5. Cargar el programa en la memoria del autómeta desde la unidad de programación.

6. Probar y depurar el programa.

# ÍNDICE

- Introducción a la programación del autómeta
  - Etapas en la programación del autómeta.
  - Definición del sistema de control
  - Definición de las variables del modelo de control
- Lenguajes de programación



# Definición del Sistema de Control

- Sistemas sin complejidad  $\Rightarrow$  lenguaje (descripción literal)
- Sistemas más complejos  $\Rightarrow$  Herramientas de representación basadas en símbolos
- Clasificación según los símbolos utilizados:
  - Descripciones literales:
    - Enumerando literalmente las acciones a desarrollar por el mismo, expuestas secuencialmente y con indicación de las condiciones de habilitación o validación en cada caso.
    - Puede resultar de difícil comprensión.
    - Normalmente se complementa con otras formas de representación
  - *Algebraicas*:
    - Funciones booleanas y aritméticas
  - *Gráfica*:
    - Diagramas Lógicos
    - Esquemas de Relés
    - Esquemas de contactos KOP
    - Diagramas GRAFCET

# Definición del sistema de control: Funciones algebraicas

- Funciones algebraicas
  - Las funciones algebraicas de cada una de las salidas se obtienen:
    - o bien directamente de la descripción literal del proceso a controlar
    - o bien se aplican métodos de síntesis basados en el álgebra de Boole (tablas de verdad, Karnaugh, etc.).

$$Y = (B\bar{C} + \bar{B}C) \cdot A$$

# Definición del sistema de control: Funciones algebraicas

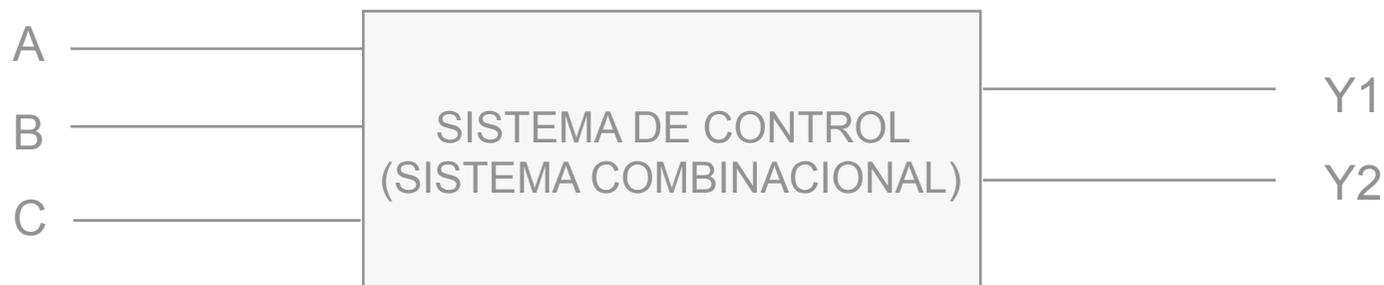
- Ejemplo:

- P.e. modelar un sistema de control que realice la siguiente función:

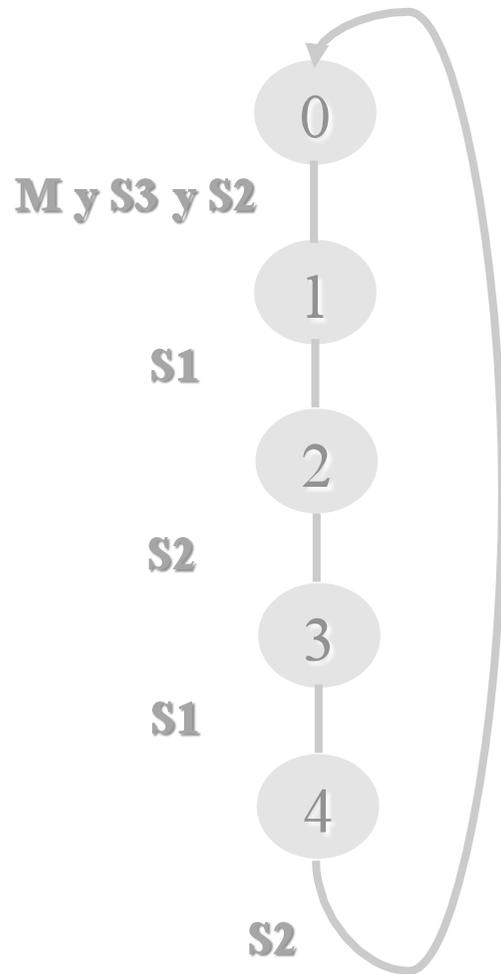
- Luz verde (Y1) tiene que estar encendida cuando las entradas A y B estén activadas o la C esté desactivada.
- La luz roja (Y2) se activa cuando la entrada A está activada y la B desactivada.

$$Y1 = (AB + \bar{C})$$

$$Y2 = (A \cdot \bar{B})$$



- Pero pensemos ahora en implementar un programa de control del estilo del visto anteriormente.



- 0: Estado inicial.
- 1: Limpieza I → Motor MP izq, XV, MC
- 2: Limpieza II → Motor MP der, XV, MC
- 3: Secado I → Motor MP izq, MV
- 4: Secado II → Motor MP der, MV

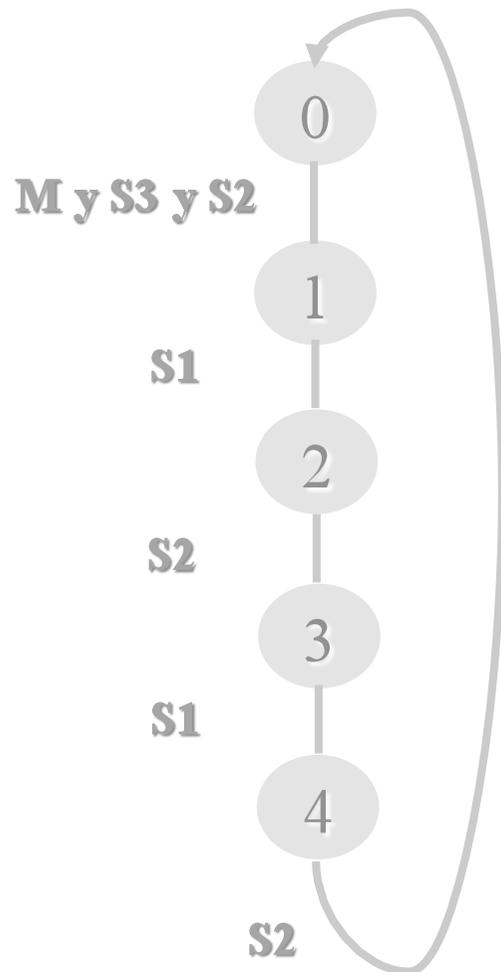
- Vamos a escribir una función algebraica para cada salida...

- $MP1 = M \cdot S3 \cdot S2$
- $MP2 = S1$
- $XV = M \cdot S2 \cdot S3$
- ...

- ¡Mal! ¿Qué falla?

- Las salidas está únicamente activas cuando el sensor cambia de estado. P.e. S2 se activa al llegar al final de carrera, pero se desactiva cuando movemos
- En realidad las salidas dependen de las entradas y del ESTADO en el que se encuentre el proceso.

- 0: Estado inicial.
- 1: Limpieza I → Motor MP izq, XV, MC
- 2: Limpieza II → Motor MP der, XV, MC
- 3: Secado I → Motor MP izq, MV
- 4: Secado II → Motor MP der, MV



- Hay que codificar también los estados.

- Posible solución:

**Inicio:**

$E0=1, E1=0, E2=0....$

**Transiciones:**

$E1 = E0 \cdot M \cdot S2 \cdot S3$

$E2 = E1 \cdot S1$

$E3 = E2 \cdot S2$

$E4 = E3 \cdot S1$

...

**Ojo: Únicamente un estado activo al tiempo**

Si  $E1==1$ , entonces  $E0 = 0$

Si  $E2 ==1$ , entonces  $E1=0$

**Salidas asociadas a los estados:**

$MP1=E1, XV=E1, MC=E1$

$MP2=E2, XV=E2, MC=E2$

$MP1=E3, MV=E3$

...

## Definición del sistema de control: funciones algebraicas

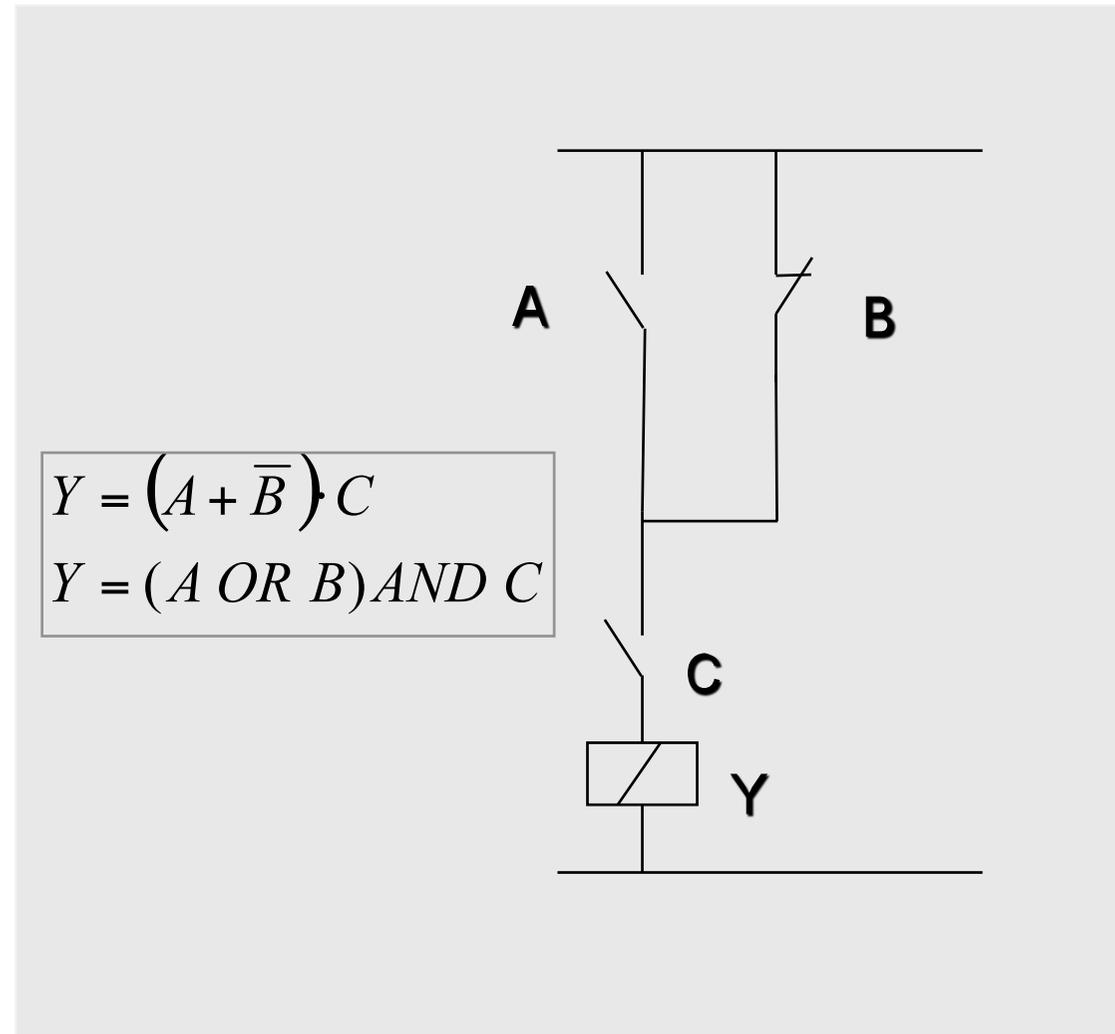
- En resumen:
  - Las funciones algebraicas modelan correctamente condiciones de transición (transición entre estados o etapas).
  - La representación de sistemas secuenciales es complicada → Es difícil de analizar y sintetizar sistemas secuenciales.
  - Limitado a la representación de combinaciones de variables.

# Definición del Sistema de Control: Esquemas de relés

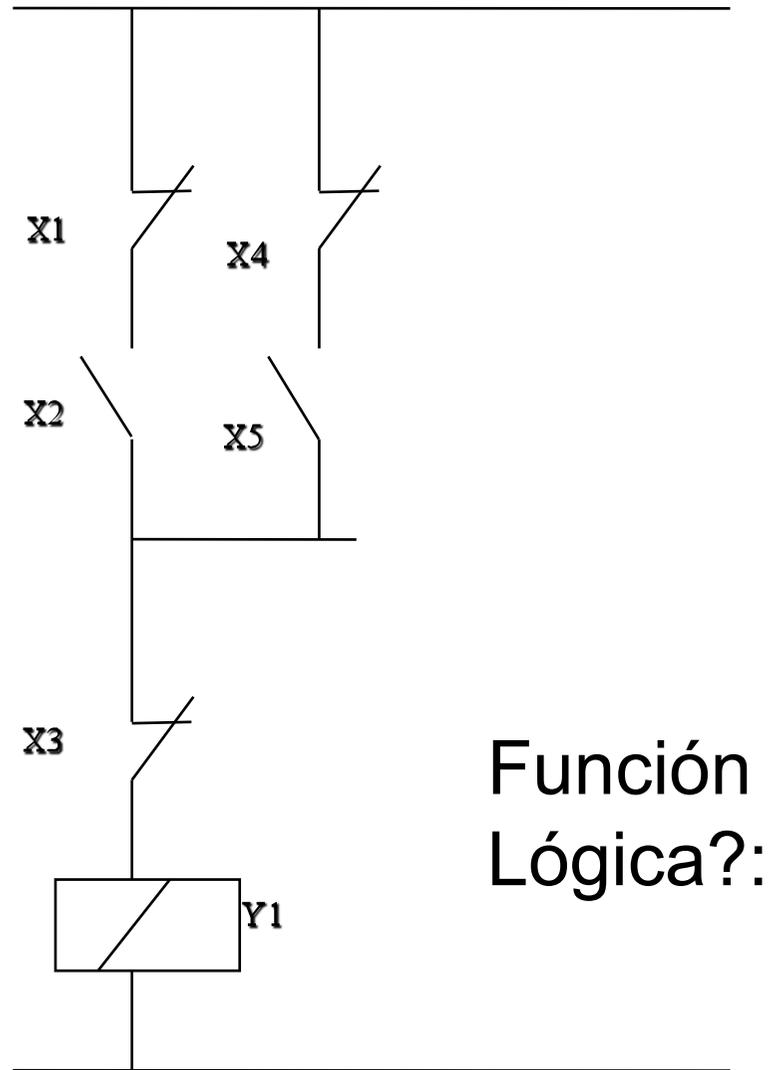
- Esquemas de relés
  - Origen: en las representaciones electromecánicas de sistemas de mando.
  - Representa la circulación de corriente eléctrica y conmutadores o interruptores asociados a sensores.
  - Representación de sistemas sencillos: señales lógicas binarias todo-nada (digitales), o bloques secuenciales sencillos como temporizadores y contadores.
  - Igual que con las funciones lógicas, es difícil representar sistemas secuenciales.

# Definición del Sistema de Control: Esquemas de relés

- Ejemplo:
  - Contacto NO (normalmente abierto, se cierra si  $A=1$ ).
  - Contacto NC (normalmente cerrado, se abre si  $B=1$ )
  - Líneas en paralelo, OR lógico.
  - Serie, AND lógico.
  - Salida Y



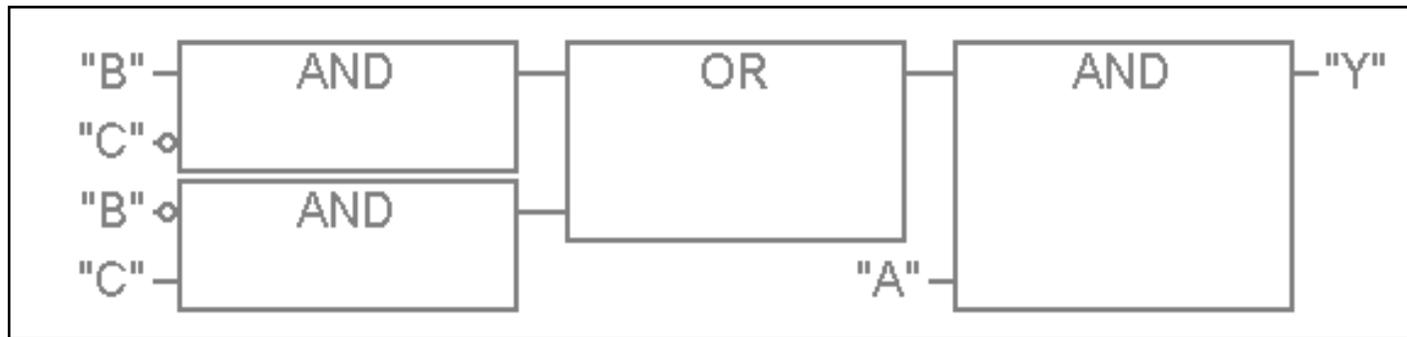
# Definición del Sistema de Control: Esquemas de relés



# Definición del sistema de control: Diagramas lógicos

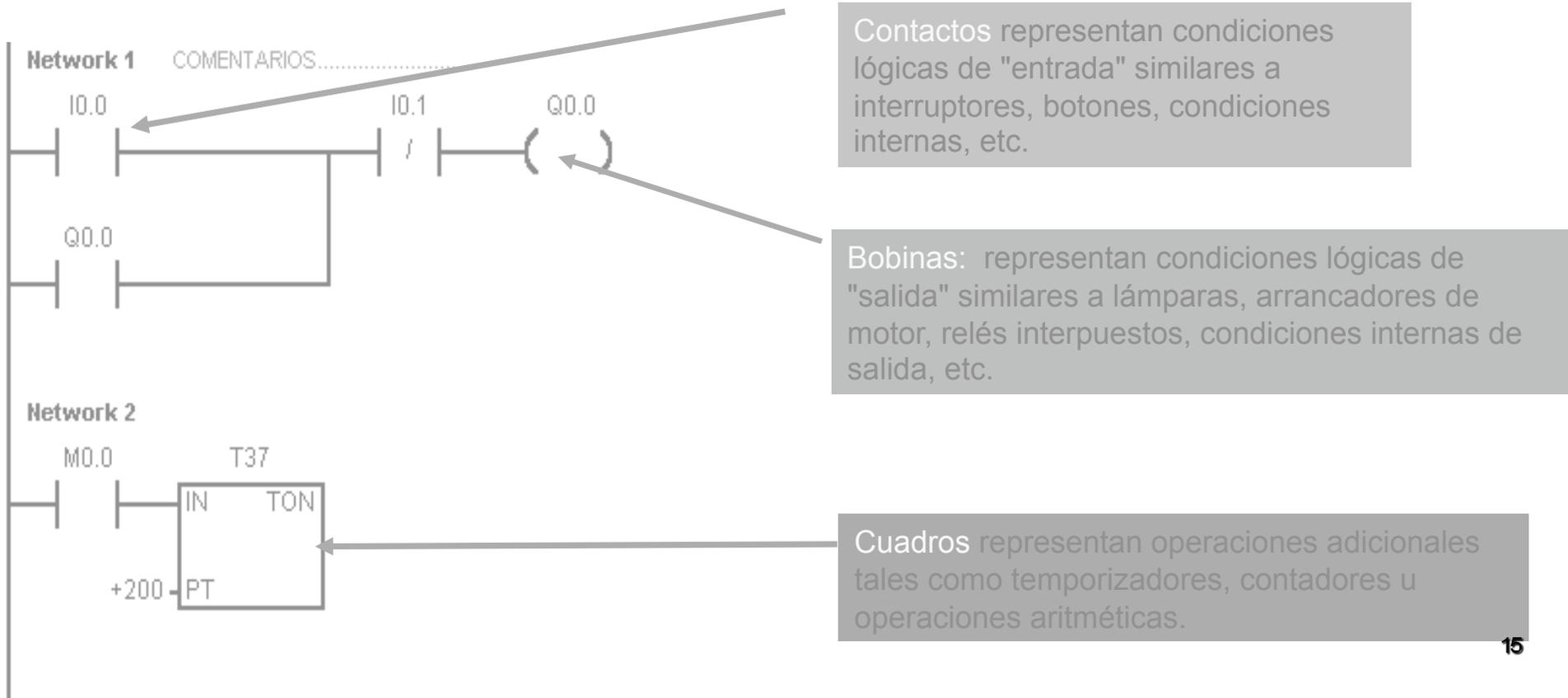
- Diagramas lógicos

- Algunos lenguajes están basados en el uso de bloques lógicos como estos.
- Cada bloque representa una operación.
- Diagrama de funciones (FUP)



# Definición del Sistema de Control: Esquemas de contactos (KOP)

- Es un lenguaje gráfico similar al lenguaje de relés.
- Usa símbolos que representan contactos y bloques funcionales.
- Está basado en contactos y en bobinas y permite codificar de forma muy visual secuencias de control.
- Permite incluir fácilmente otros bloques como contadores o temporizadores.

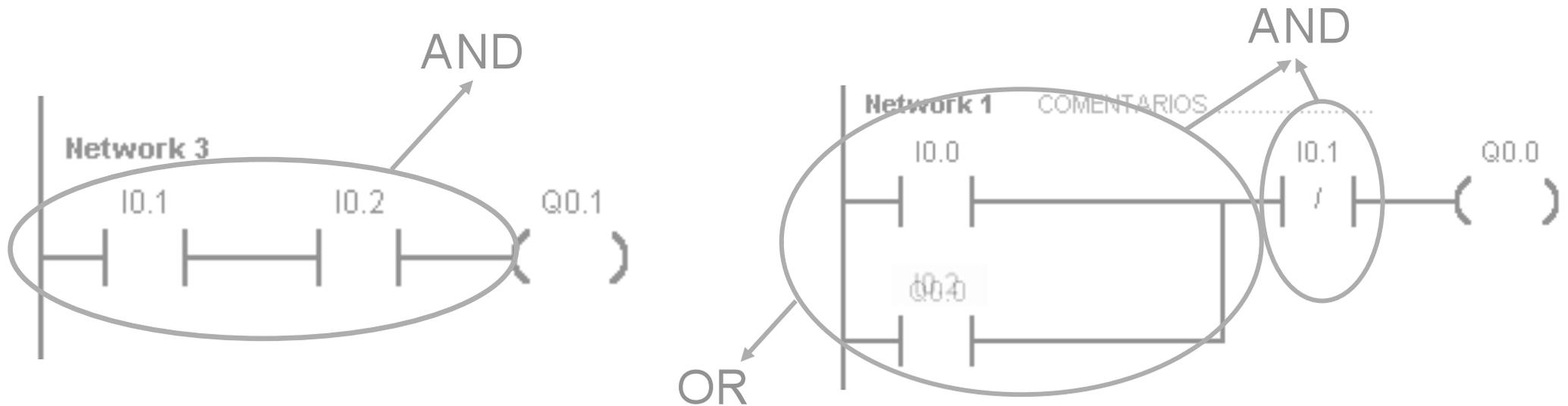


# Definición del Sistema de Control: Esquemas de contactos (KOP)

- Los programas que escribamos en los autómatas estarán basados en lenguaje KOP
- Se agrupan en segmentos (Networks)
- El programa se ejecuta segmento por segmento, de izquierda a derecha y luego de arriba a abajo.
- Tras finalizar el último segmento la CPU comienza a ejecutar desde el principio.
- Permiten hacer referencia directamente a las entradas y salidas del autómata
  - I0.0, I0.1, I0.2, I0.3... entradas digitales binarias del autómata.
  - Q0.0, Q0.1, Q0.2... salidas digitales binarias
- KOP se estudia en profundidad en el tema siguiente.

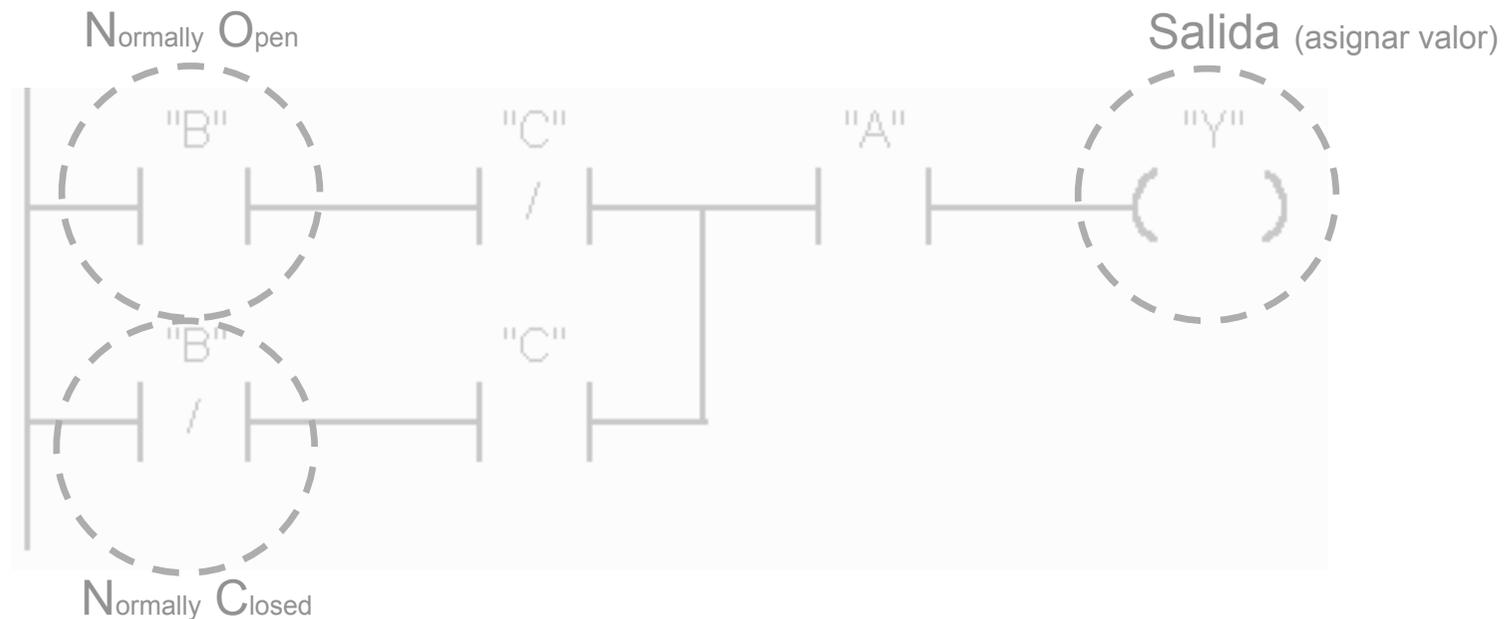
# Definición del Sistema de Control: Esquemas de contactos (KOP)

- La operación AND se implementa mediante contactos en serie
- La operación OR se implementa mediante contactos en paralelo



# Definición del Sistema de Control: Esquemas de contactos (KOP)

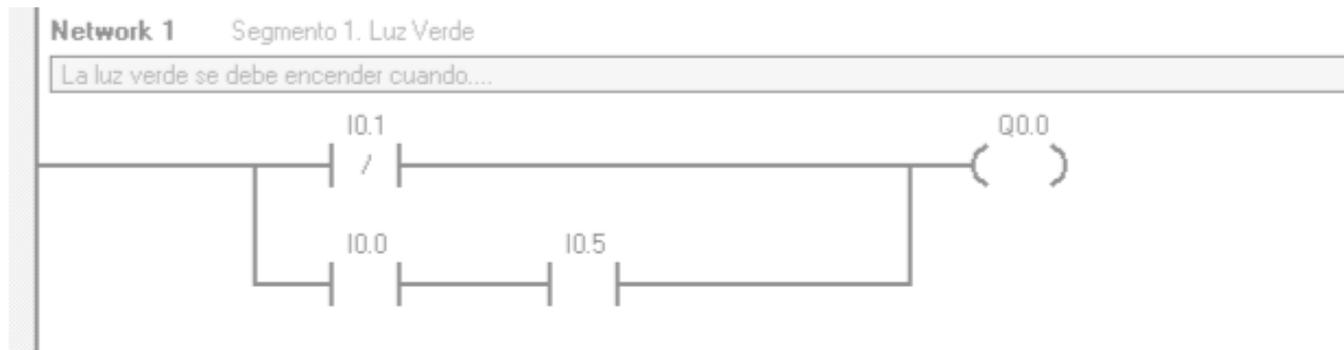
- Para entenderlo debemos pensar en la circulación de la corriente (de izquierda a derecha).



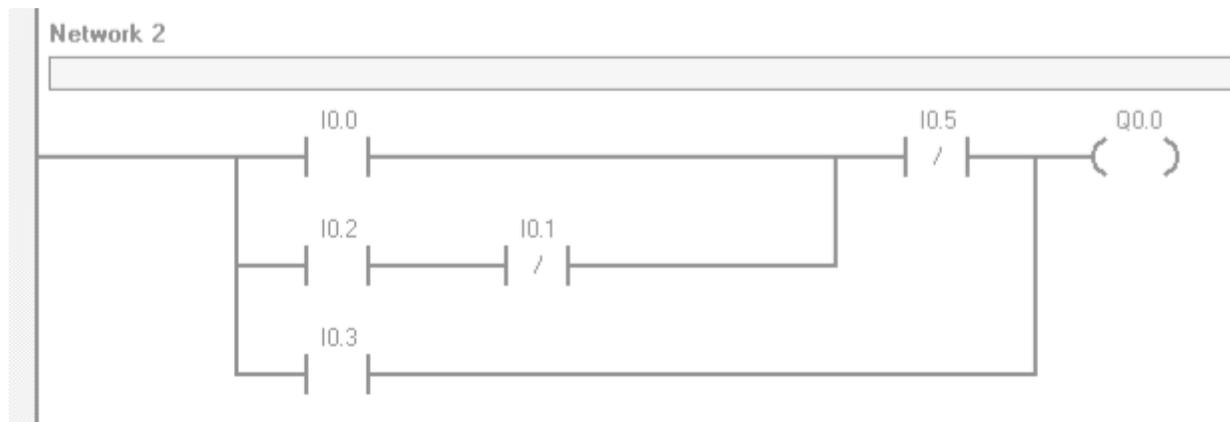
- La función lógica es: ¿?

# Definición del Sistema de Control: Esquemas de contactos (KOP)

- Y al revés:
  - La luz verde (Q0.0) deberá estar encendida siempre que las entradas I0.0 y I0.5 estén activas o la entrada I0.1 se encuentre a nivel bajo.¿?

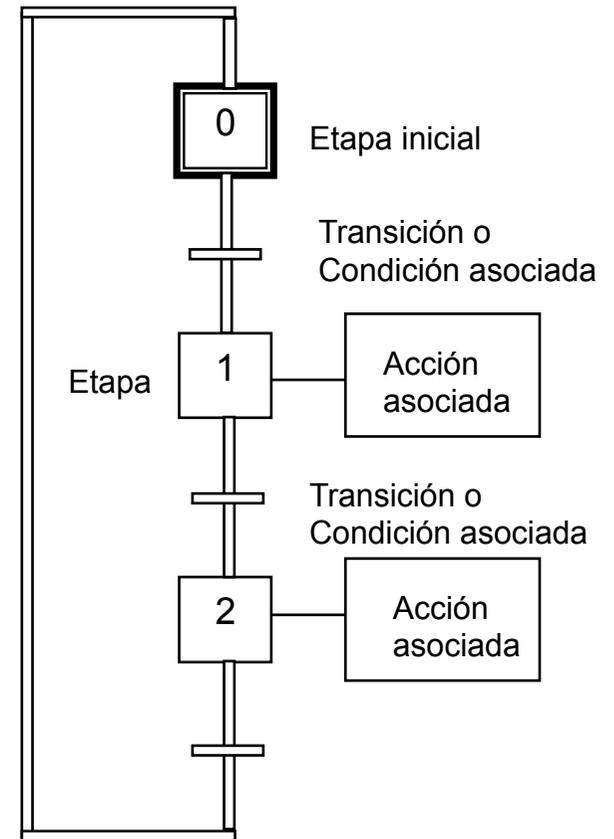


- usando una función algebraica ¿?     
  - $Q0.0 = ((I0.0 \text{ OR } (I0.2 \text{ AND } \overline{I0.1})) \text{ AND } I0.5) \text{ OR } I0.3$



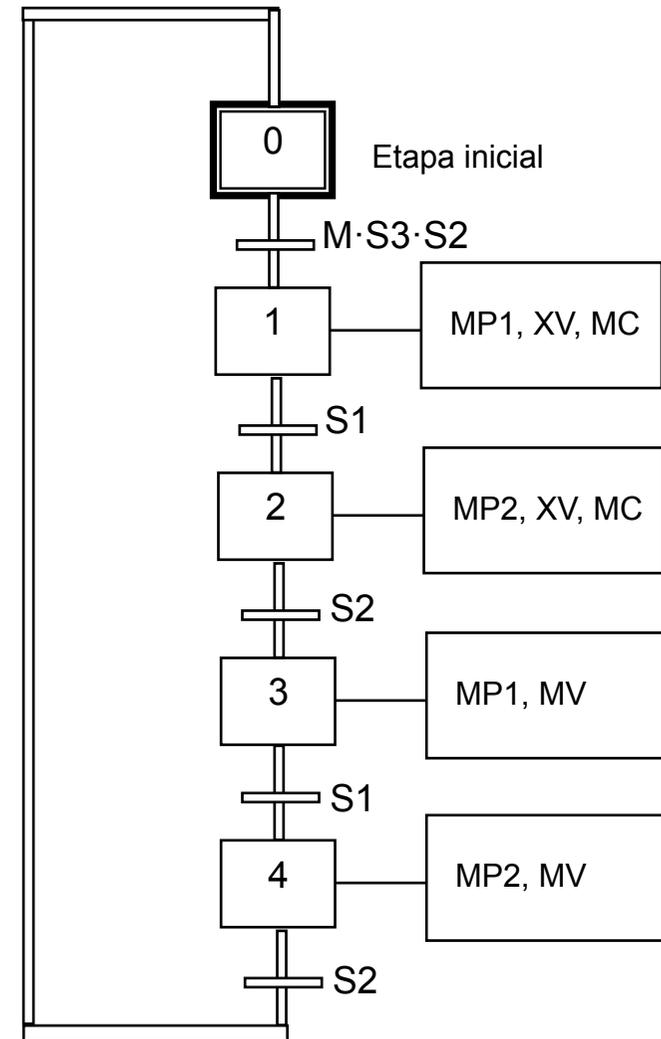
# Definición del Sistema de Control: GRAFCET

- **GRAFCET** (*“Grappe de Comande Etape Transition”*, Gráfico de Comando Etapa–Transición)
  - Es un diagrama donde se indican de forma gráfica las etapas que tienen asociadas unas determinadas acciones a realizar sobre el proceso y las condiciones o transiciones que provocan que se produzca el paso de una etapa a otra.
  - Está especialmente indicado para representar procesos discretos secuenciales.
  - Normalizado: *International Electrotechnical Commision IEC 848*
  - Una de las mejores herramientas para representar automatismos secuenciales



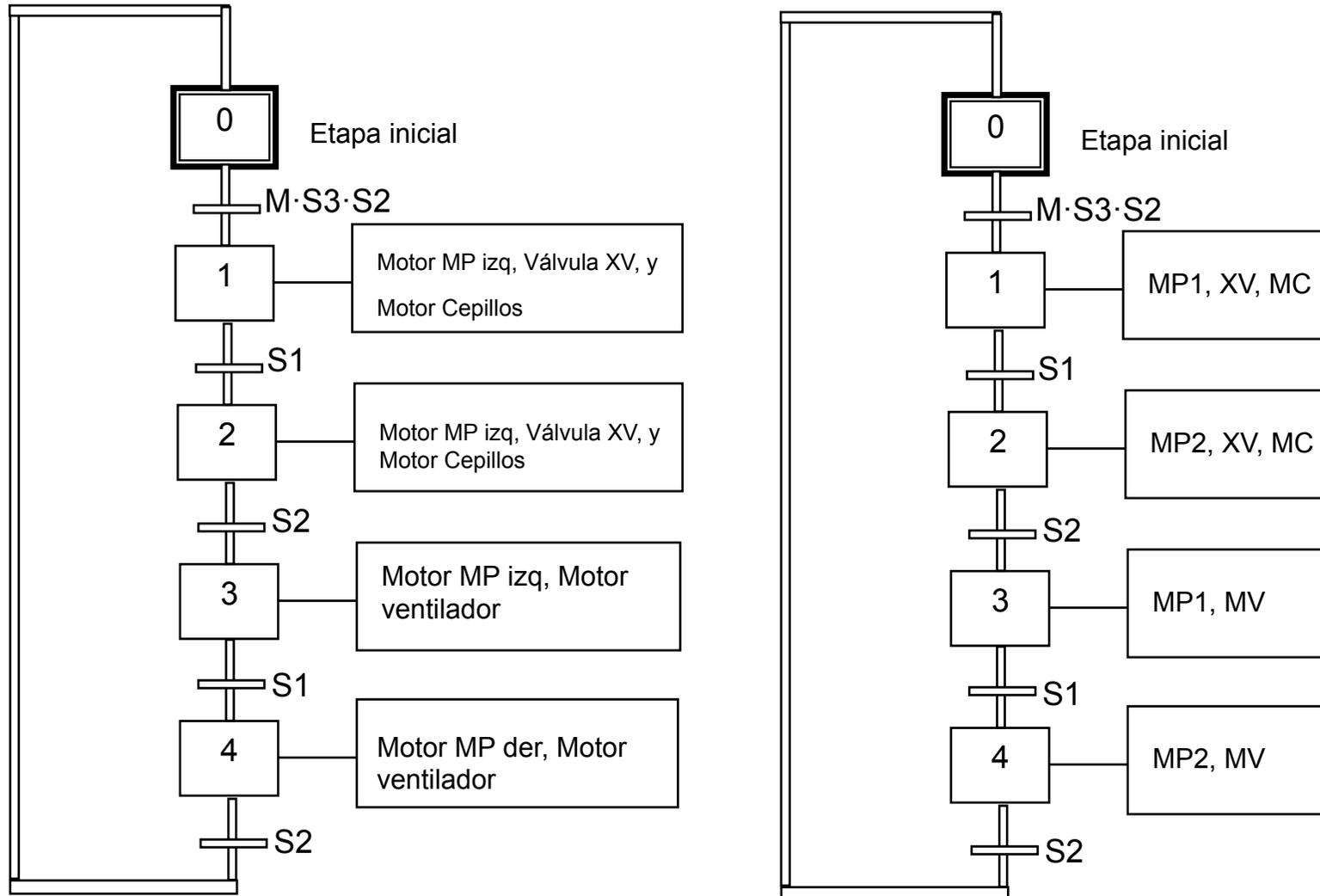
# GRAFSET: Ejemplo

- Para la máquina de lavado.
- La idea es similar al grafo de estados que ya realizamos.
- Pero se asocian las salidas de forma más clara.
- Está normalizado. P.e.: etapa inicial con doble cuadro.
- Permite representar bucles, acciones en paralelo, temporizaciones de forma fácilmente comprensible.



# GRAFCET: Ejemplo

- Se permite la descripción con variables o lenguaje literal (niveles GRAFCET)



# ÍNDICE

- Introducción a la programación del autómeta
  - Etapas
  - Definición del sistema de control
  - Definición de las variables del modelo de control
- Lenguajes de programación

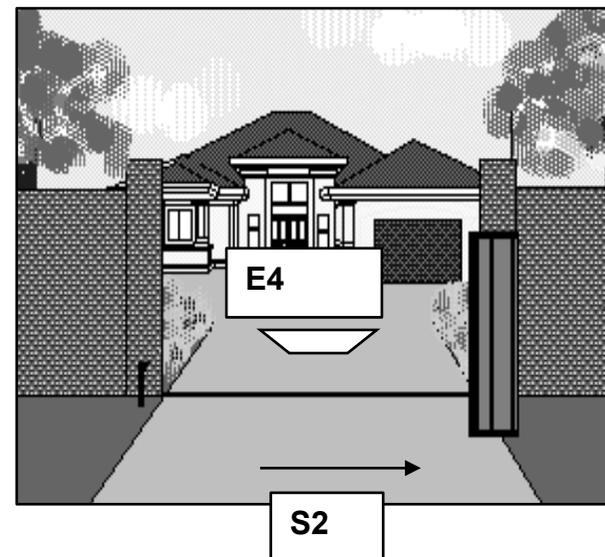
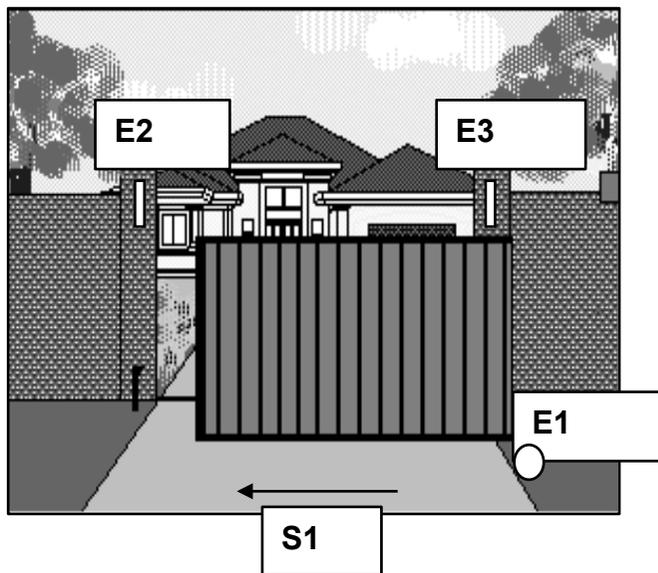


# Variables del Modelo de Control

- Definir las variables que intervienen y asignarles direcciones de memoria
  - Especificar las entradas y salidas que tendrán nombres simbólicos
  - Asignar a las E/S del Autómata
  - Especificar variables internas

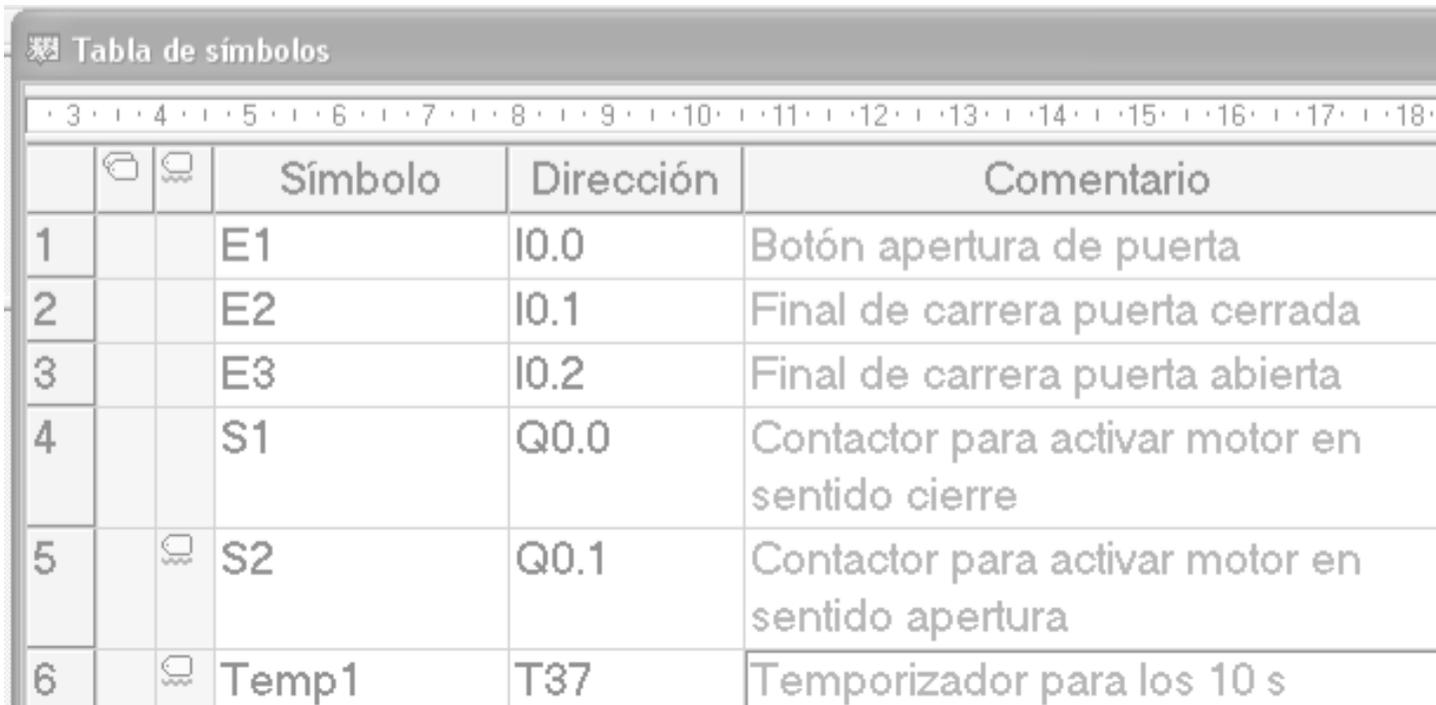
# Ejemplo

- **Ejemplo:** “Control de una puerta corredera accionada por medio de un motor”
  - La puerta se abre al presionar el botón *E1* situado enfrente de la puerta.
    - Si se activa *E1*, se cierra el contactor *S2* (activar motor sentido apertura) y se mantiene cerrado hasta que se active el interruptor *E3* de final de carrera.
    - Una vez abierta la puerta, se activa el temporizador *T1*, y transcurridos 10 segundos, la puerta se cierra mediante el contactor *S1* (activa motor en sentido de cierre).
    - La acción de cerrar se produce hasta que se detecta fin de carrera *E2* y si no se detecta un vehículo con el sensor de paso *E4*.
  - Las lámparas *LED1* y *LED2* indican cuándo se está cerrando o abriendo la puerta respectivamente.



# Ejemplo

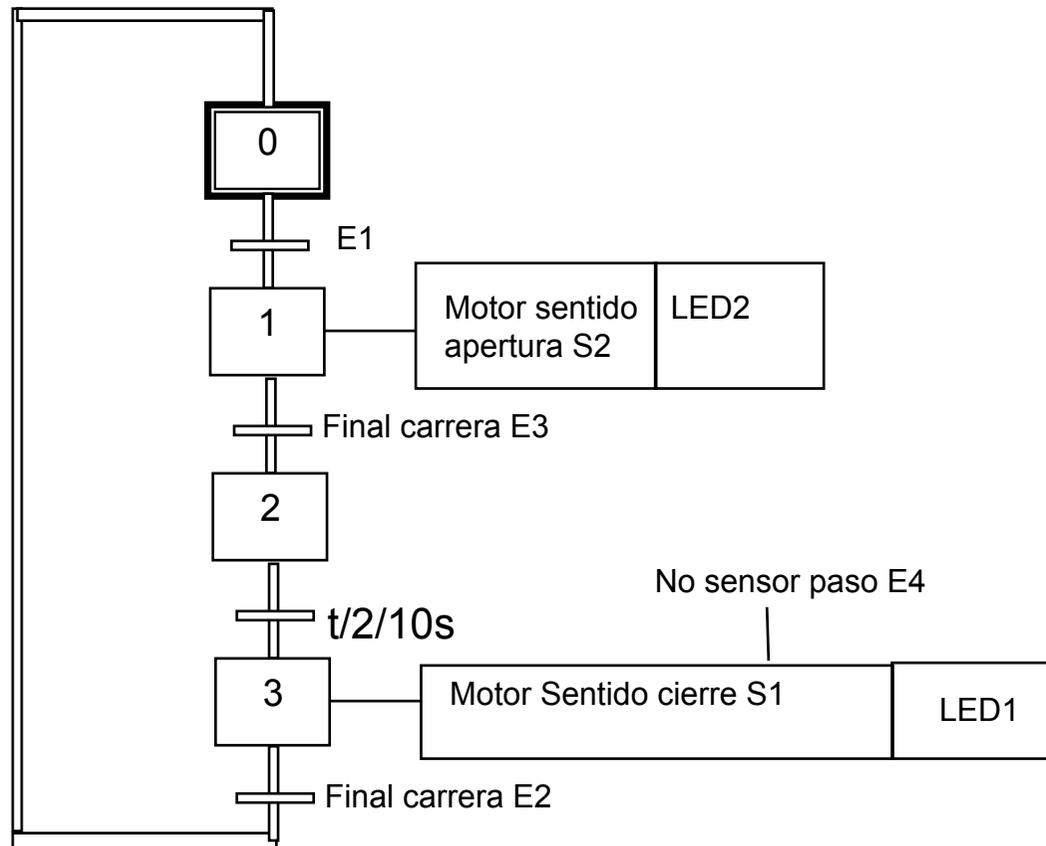
- Asignamos los símbolos a las direcciones físicas de las salidas del autómeta. Facilita la creación del programa, evitando errores.
- Añadimos la descripción de cada salida/entrada (Q/I)
- Se consideran también otras variables como temporizadores y contadores.



			Símbolo	Dirección	Comentario
1			E1	I0.0	Botón apertura de puerta
2			E2	I0.1	Final de carrera puerta cerrada
3			E3	I0.2	Final de carrera puerta abierta
4			S1	Q0.0	Contactador para activar motor en sentido cierre
5			S2	Q0.1	Contactador para activar motor en sentido apertura
6			Temp1	T37	Temporizador para los 10 s

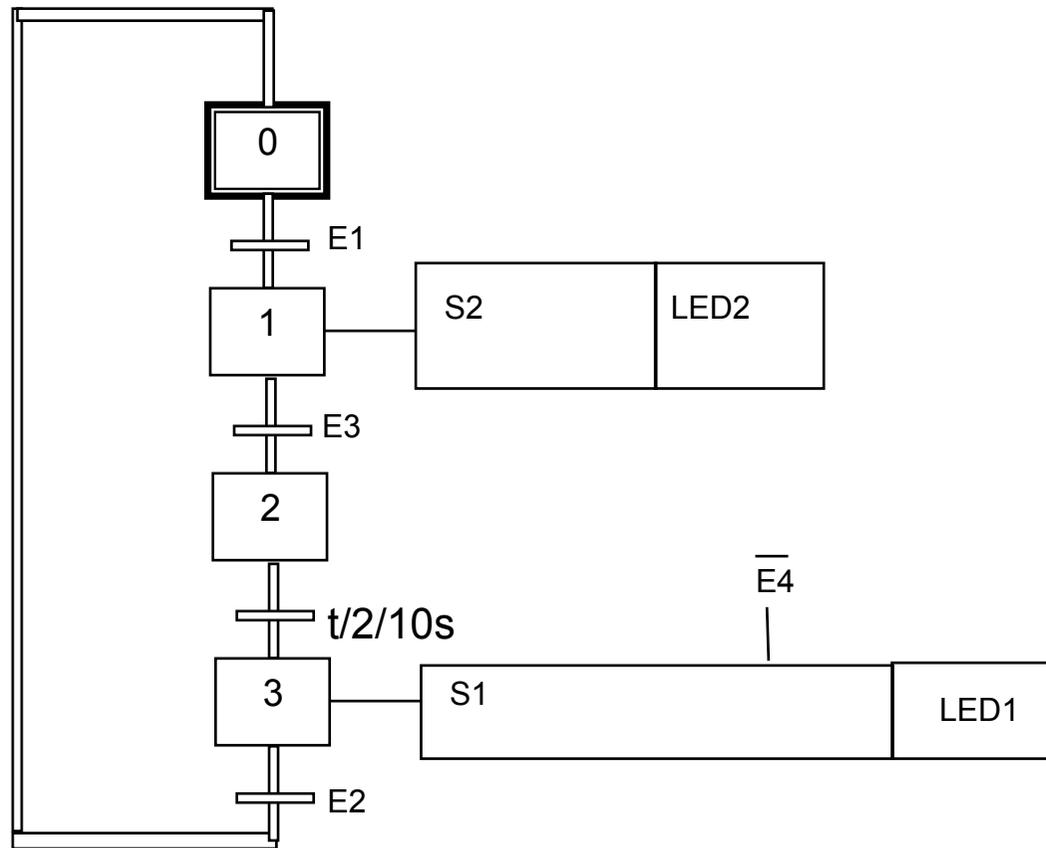
# Ejemplo

- Dada la descripción literal, vamos a asociarle un diagrama GRAFCET



# Ejemplo

- Dada la descripción literal, vamos a asociarle un diagrama GRAFCET



# ÍNDICE

- Introducción a la programación del autómeta
  - Etapas
  - Definición del sistema de control
  - Definición de las variables del modelo de control
-  Lenguajes de programación

# Lenguajes de Programación

- **Lenguajes Literales**

- **Siemens STEP7:**

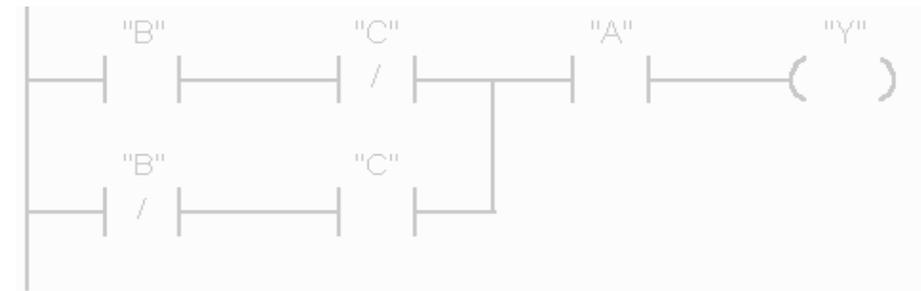
- Lista de Instrucciones: AWL.

```
LD  "B"  
AN  "C"  
LDN "B"  
A   "C"  
OLD  
A   "A"  
=   "Y"
```

- **Lenguajes Gráficos**

- **Siemens STEP 7:**

- Esquema de contactos: KOP
    - Diagrama de funciones: FUP



- **GRAFCET: Como tal no es un lenguaje sino una representación que debe ser traducida a un lenguaje específico.**

- Lista de instrucciones (AWL), diagramas de contactos (KOP) pueden ser insuficientes para programar aplicaciones complejas para autómatas de gama media y alta → Posibilidad de otros lenguajes C/C++

# Lenguajes de programación

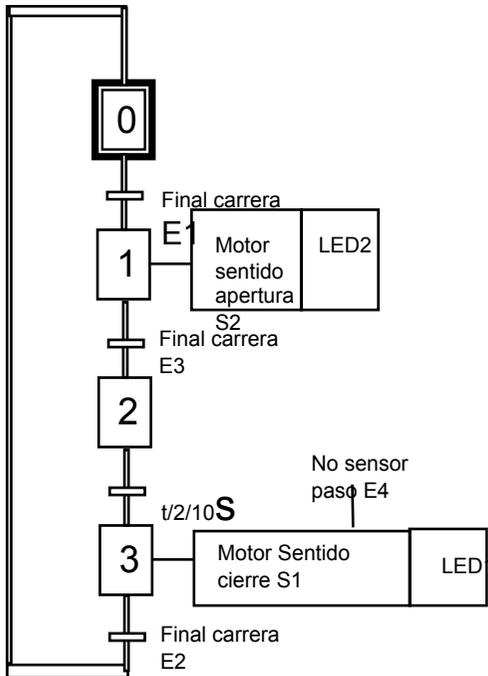
- Lista de instrucciones (AWL)
  - Los programas se escriben mediante una secuencia de instrucciones.
  - La lista de instrucciones es muy cercana a las instrucciones del microprocesador.
  - El tiempo de desarrollo de una aplicación es alto.

$$Y = (B\bar{C} + \bar{B}C) \cdot A$$

```
LD  "B"  
AN  "C"  
LDN "B"  
A   "C"  
OLD  
A   "A"  
=   "Y"
```

# Estrategia que seguiremos

1.- Modelado del sistema de control: GRAFCET

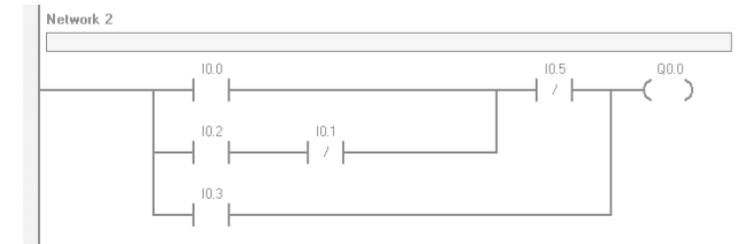


2.- Identificar las señales de entrada y salida del autómeta.

3.- Asignar las direcciones de entrada/salida

Tabla de símbolos				
		Símbolo	Dirección	Comentario
1		E1	I0.0	Botón apertura de puerta
2		E2	I0.1	Final de carrera puerta cerrada
3		E3	I0.2	Final de carrera puerta abierta
4		S1	Q0.0	Contactora para activar motor en sentido cierre
5		S2	Q0.1	Contactora para activar motor en sentido apertura
6		Temp1	T37	Temporizador para los 10 s

4.- Codificar mediante KOP



- Si el proceso es sencillo, con práctica es posible pasar directamente a la programación.
- En el caso general (examen) será necesario seguir estas etapas.