COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI ICADE CIHS

MÁSTER EN INGENIERÍA INDUSTRIAL
ASIGNATURA
AUTOMATIZACIÓN INDUSTRIAL

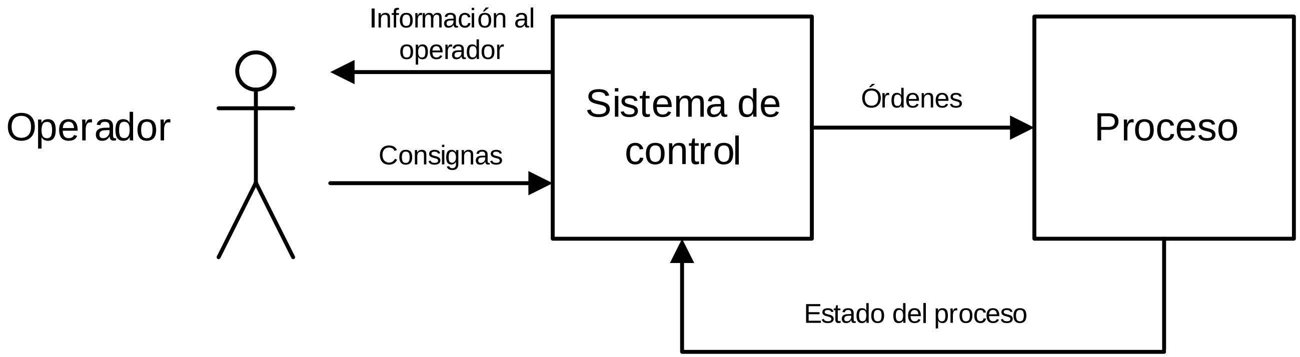
Matemáticas de la Automatización

Prof. Dr. José Antonio Rodríguez Mondéjar
mondejar@comillas.edu

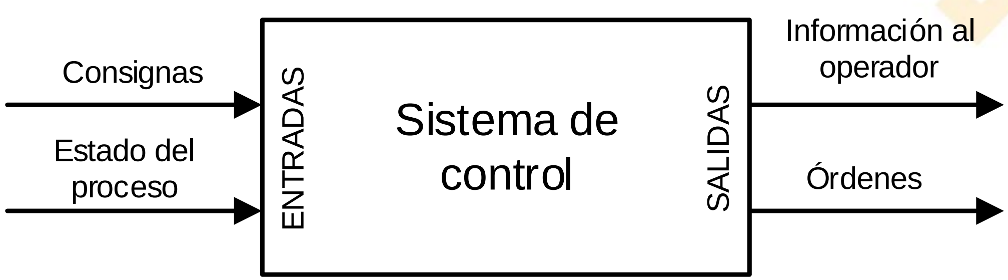
Escuela Técnica Superior de Ingeniería ICAI
Departamento de Electrónica, Automática y Comunicaciones

Enero 2022

¿Qué es automatizar según las matemáticas?



Automatizar es determinar las funciones matemáticas que calculan el valor de las salidas a partir del valor actual de las entradas y de la evolución histórica de estos valores



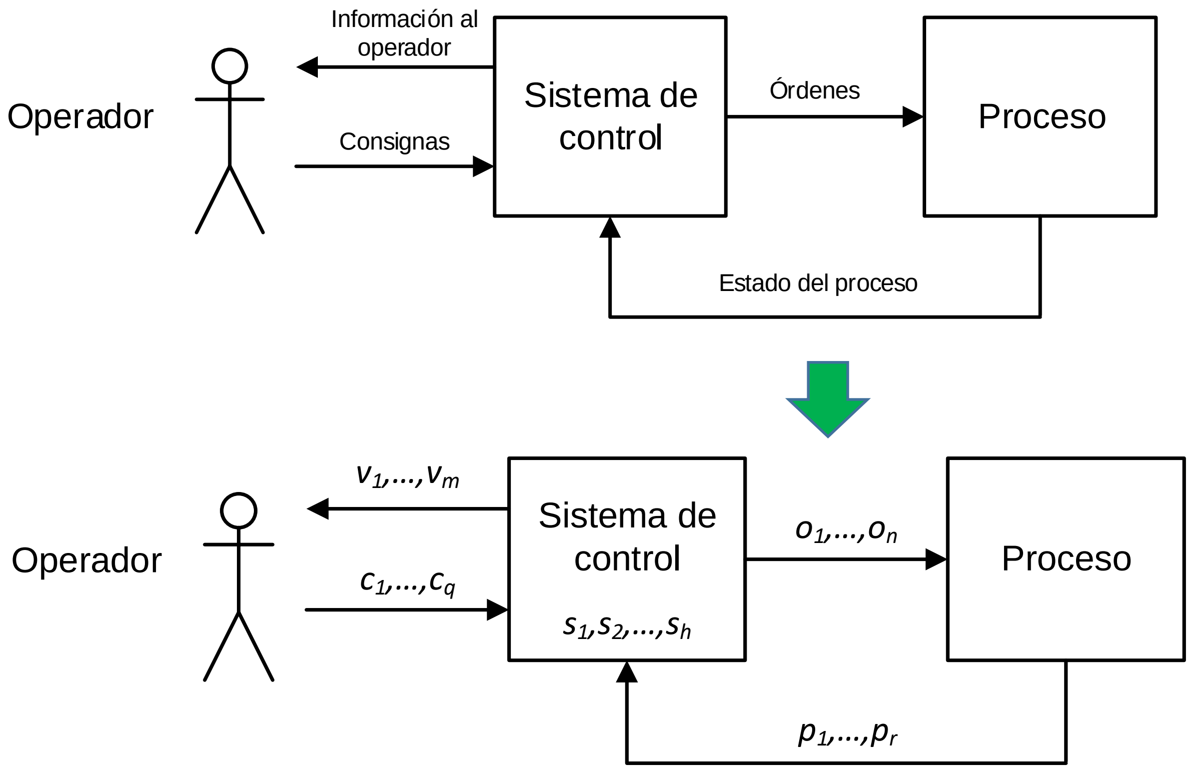
¿Por qué estudiar las matemáticas de la automatización?

COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

- **Las matemáticas permiten:**
  - Generalizar las soluciones: abaratar las soluciones
  - Simplificar las soluciones: abaratar las soluciones
  - Garantizar la efectividad de las soluciones: aumentar la robustez
  - Adecuar las soluciones a las posibles implementaciones
- **Gran parte las matemáticas necesarias ya se han visto**
  - Digital: “Álgebra de Boole”
    - Transparencias con título en azul: temas que debe conocer el alumno
  - Regulación: ecuaciones diferenciales, transformada de Laplace…
- **¿Qué se aporta aquí?**
  - Visión unificada y sistemática: patrones bien definidos y acotados
    - Ir más allá de las soluciones basadas en pura receta
  - Aplicación al entorno industrial
    - Múltiples variables, criterios de seguridad, …

¿Cómo ve las matemáticas un sistema automatizado?

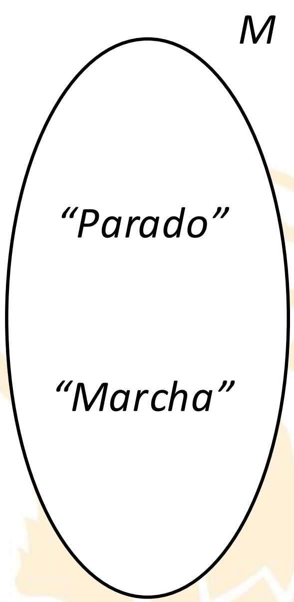
- **Un sistema automatizado son variables ligadas mediante funciones**



comillas.edu

Revisión matemáticas: conjunto

- **No se puede entender el concepto de variable sin entender el concepto de conjunto**
- **Conjunto**
  - Grupo de objetos o elementos que comparten una propiedad común de tal manera que se pueda indicar claramente si un elemento pertenece al conjunto o no.
  - Los conjuntos se denotan mediante un símbolo
    - Típico símbolo: concatenación de una o varias letras
      - Primera en mayúscula
  - Dos formas típicas de definir un conjunto
    - Extensión: $M = \{"Parado", "Marcha"\}$
      - Modos de funcionamiento de un sistema
    - Comprensión o mediante una propiedad
      - $Piso = \{n \in \mathbb{N} / n \leq 41\}$
      - Pisos a los que puede dar servicio un ascensor



comillas.edu

Revisión matemáticas: variable

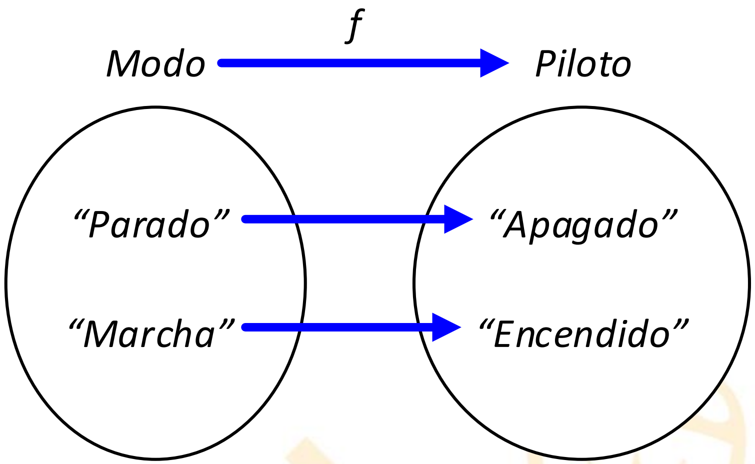
- **Símbolo que representa un elemento cualquiera de un conjunto (primera letra en minúscula)**
  - Ejemplo: m.
- **Se indica que una variable representa a un conjunto mediante** $\in$
  - $m \in M$
- **Dos típicos usos**
  - Variable que representa un valor del conjunto pero no se sabe cuál es
  - Variable que tiene un valor concreto de ese conjunto en un instante dado
    - En un momento $m = "Parado"$
    - En otro momento $m = "Marcha"$

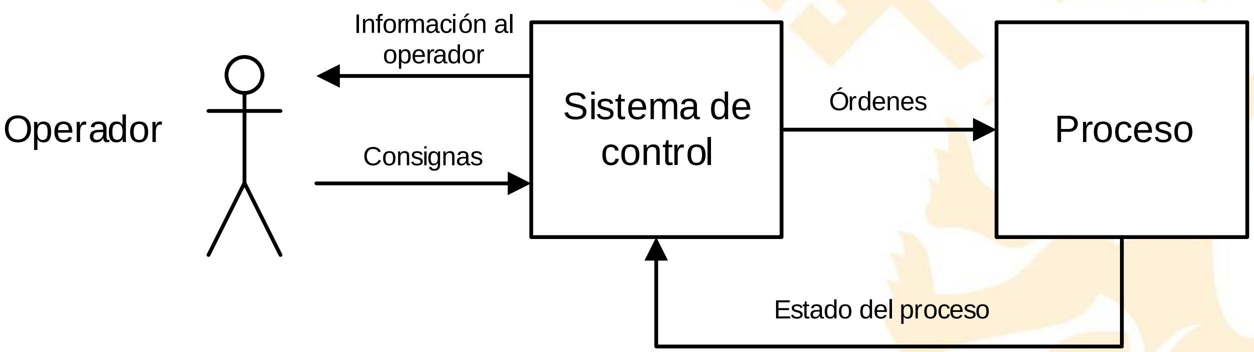
Revisión matemáticas: función

- **Correspondencia entre dos conjuntos** A **y** B **donde a cada elemento de** A **se le asigna un único elemento de** B
  - Se representa mediante un símbolo:
    - Ejemplo: f
- **Notación:**
  - Basada en conjuntos: $f: A \rightarrow B$
    - Para cada elemento del conjunto A, f calcula su correspondiente en B
    - A: conjunto origen o dominio de la función
    - B: conjunto final
    - Valores utilizados de B: conjunto imagen o recorrido de f
  - Basada en variables: $x \in A$, $y \in B$, $y = f(x)$
    - Para cada valor de la variable x, f calcula el valor de la variable y
      - x es la variable independiente
      - y es la variable dependiente: su valor depende de x

Ejemplo de función en automatización: una variable

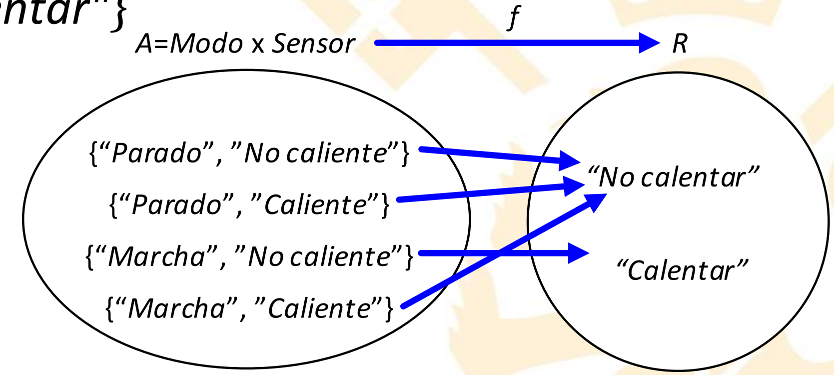
- **Ejemplo: función para visualizar el modo de funcionamiento**
- **Conjunto origen**
  - $Modo = \{"Parado", "Marcha"\}$
  - Variable $modo \in Modo$
- **Conjunto imagen**
  - $Piloto = \{"Apagado", "Encendido"\}$
  - Variable $piloto \in Piloto$
- **Función f: $\boldsymbol{Modo \rightarrow Piloto}$**
  - $piloto = f(modo)$





Ejemplo de función: varias variables

- **Ejemplo: función qué indica cuándo calentar según el modo de funcionamiento del sistema de control y de la información recibida de un sensor**
- **Conjunto 1:** $Modo = \{"Parado", "Marcha"\}$, variable $modo \in Modo$
- **Conjunto 2:** $Sensor = \{"No\ caliente", "Caliente"\}$, variable $sensor \in Sensor$
- **Conjunto origen: *A***
  - Los elementos son parejas donde el primer elemento pertenece a *Modo* y el segundo a *Sensor*
  - $A = \{\{"Parado", "No\ caliente"\}, \{"Parado", "Caliente"\}, \{"Marcha", "No\ caliente"\}, \{"Marcha", "Caliente"\}\}$
  - Definición matemática: $A = Modo \times Sensor$
    - Producto cartesiano de *Modo* por *Sensor*
- **Conjunto imagen:** $R = \{"No\ calentar", "Calentar"\}$
  - Variable $r \in R$
- **Función** $\boldsymbol{f: Modo \times Sensor \rightarrow R}$
  - $r = f(modo, sensor)$



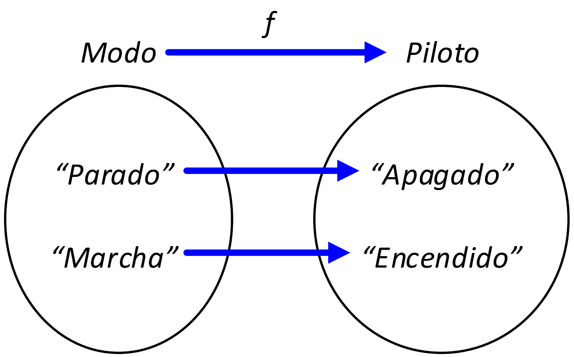
comillas.edu

¿Cómo determinar la función?

- **Representación gráfica**
  - Intuitiva pero difícil de implementar
- **Mediante tabla**
  - Para cada valor de la variable del conjunto origen se indica el valor que toma la variable del conjunto imagen
  - Intuitiva y fácil de implementar en casos sencillos
- **Mediante expresión que combina variables utilizando operadores**
  - Operadores aritméticos: Regulación
  - Operadores lógicos: se verá más adelante
    - Se basa en la abstracción del concepto de tabla
    - Menos intuitiva pero más fácil de implementar

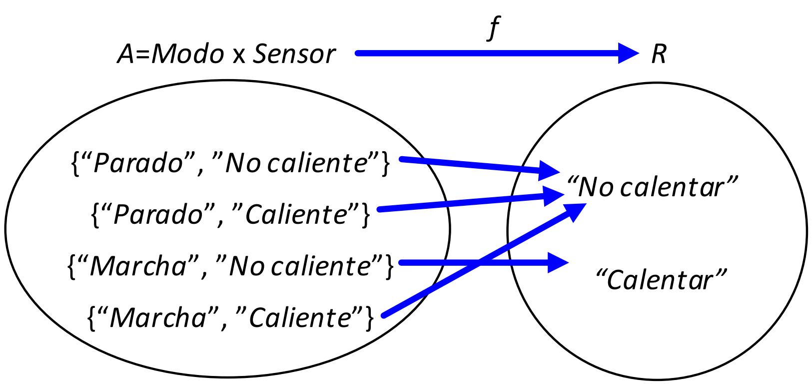
Ejemplos de cálculo con tabla

- **Visualizar el modo de funcionamiento**



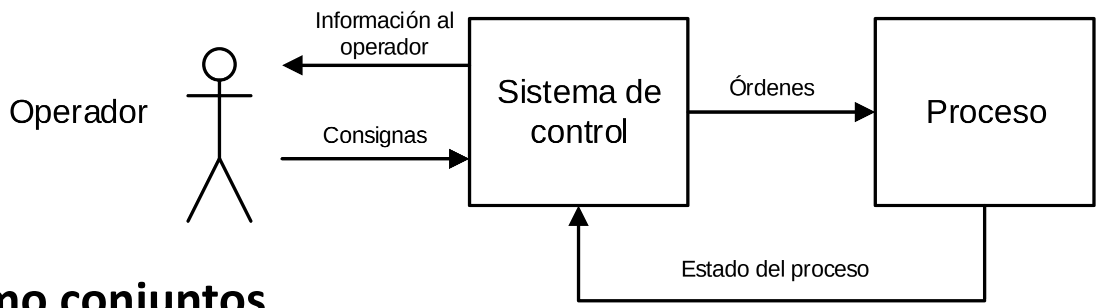
modo	piloto
"Parado"	"Apagado"
"Marcha"	"Encendido"

- **Función que indica cuándo calentar**



modo	sensor	r
"Parado"	"No caliente"	"No calentar"
"Parado"	"Caliente"	"No calentar"
"Marcha"	"No caliente"	"Calentar"
"Marcha"	"Caliente"	"No calentar"

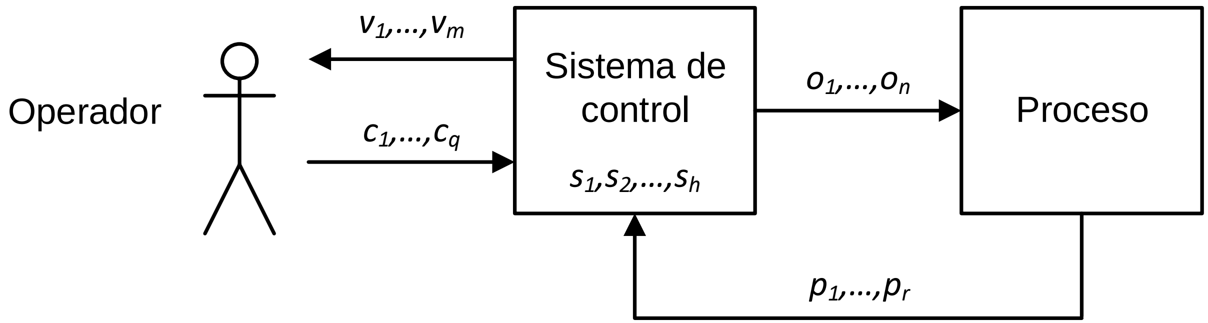
¿Cuántas variables hay en un sistema automatizado? (I)



- **Tantas como conjuntos**
- **Cada conjunto modela los posibles valores de**
  - Cada fuente de consigna: conjuntos C_i
    - $C_1 = \{$"No calentar", "Calentar"$\}$, $C_2 = \{$"No aspirar", "Aspirar"$\}$
  - Cada fuente de información sobre el proceso (sensores): conjuntos P_i
    - $P_1 = \{$"No cerrado", "Cerrado"$\}$, $P_2 = \{$"Frio", "Templado", "Caliente"$\}$
  - Cada receptor de órdenes (accionamientos o actuadores): conjuntos O_i
    - $O_1 = \{$"No girar", "Girar"$\}$, $O_2 = \{$"No Bombear", "Bombear"$\}$
  - Cada medio para informar al operador: conjuntos V_i
    - $V_1 = \{$"No subiendo", "Subiendo"$\}$, $V_2 = \{$"Normal", "Alarma"$\}$
  - Conjuntos que modelan los posibles estados de cada agrupación de estados internos: conjuntos S_i
    - $S_1 = \{$"Modo automático", "Modo manual", "Modo semiautomático"$\}$
    - $S_2 = \{$"Parado, "Subiendo", "Subido", "Bajando", "Bajado"$\}$

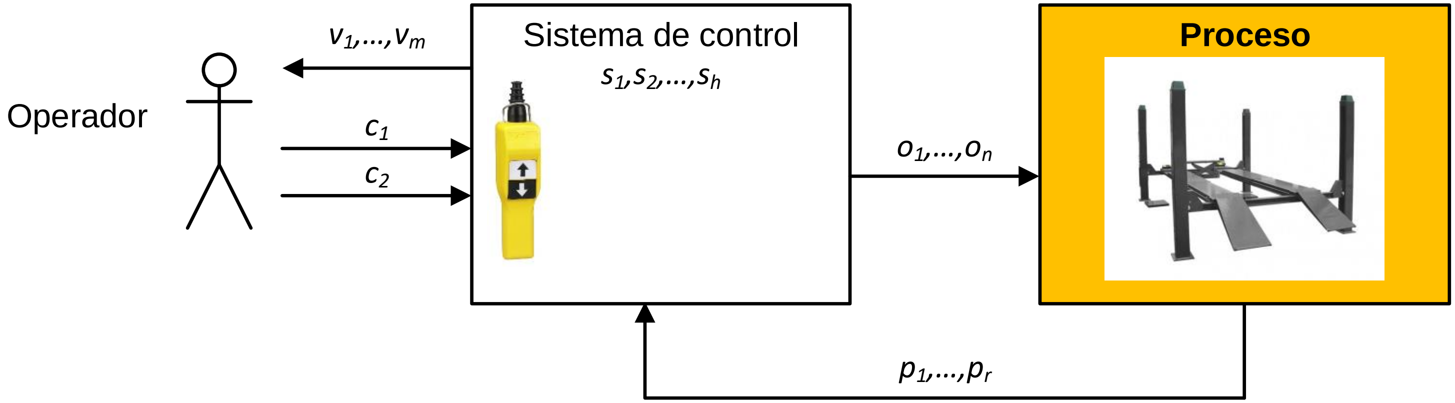
No todos tienen dos valores (conjuntos binarios)

¿Cuántas variables hay en un sistema automatizado? (II)



- q **variables independientes de tipo consigna:** c_i
  - Cada variable representa una fuente de consigna del operador: $c_i \in C_i$
- r **variables independientes de tipo proceso:** p_i
  - Cada variable representa una fuente de información sobre el proceso: $p_i \in P_i$
- n **variables dependientes de tipo orden:** o_i
  - Cada variable representa un receptor de órdenes del proceso: $o_i \in O_i$
- m **variables dependientes de tipo información:** v_i
  - Cada variable representa un medio de informar al operador: $v_i \in V_i$
- h **variables de tipo estado:** s_i
  - Cada variable representa a un grupo de posibles estados internos del sistema de control: $s_i \in S_i$

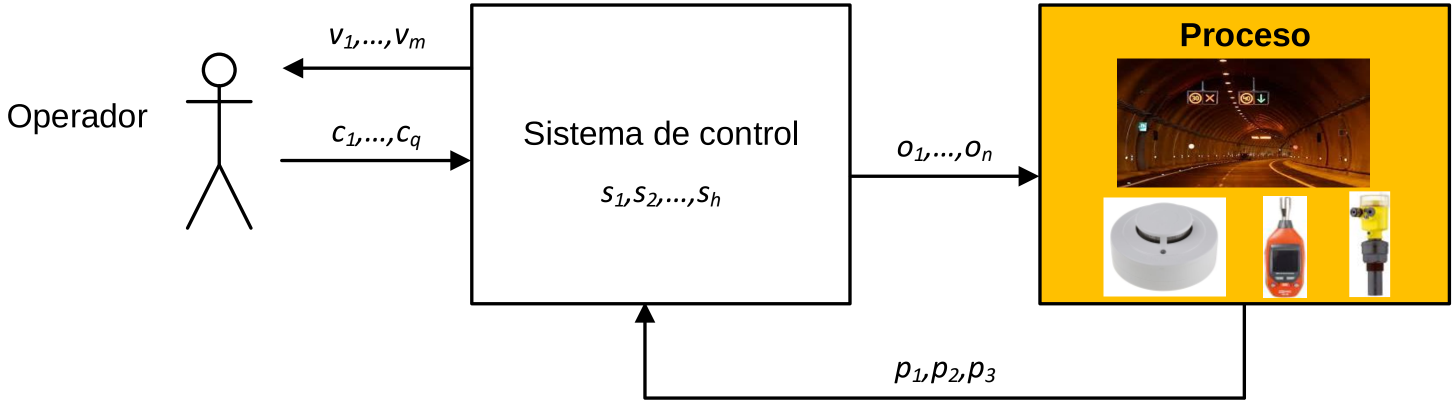
Ejemplo de fuentes de consigna: botonera para elevador



- **El operador tiene dos botones para dar las consignas: dos fuentes de consigna**
  - Conjunto $C_1 = \{$"No orden subir", "Subir"$\}$
    - Asociado a pulsador subir
  - Conjunto $C_2 = \{$"No orden bajar", "Bajar"$\}$
    - Asociado a pulsador bajar
- **Variables**
  - Variable $c_1 \in C_1$ asociada a pulsador subir
  - Variable $c_2 \in C_2$ asociada a pulsador bajar
- **Tabla para indicar cómo cada variable toma sus posibles valores**
  - Asociación entre acción del operador y valor de la variable

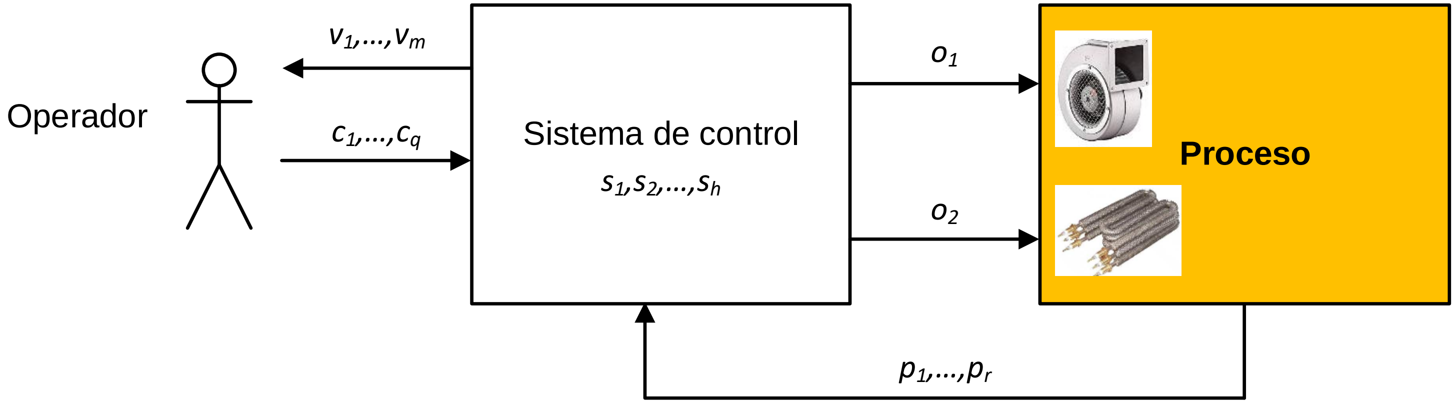
	Pulsador no pulsado	Pulsador pulsado
c_1	"No orden subir"	"Subir"
c_2	"No orden bajar"	"Bajar"

Ejemplo de información sobre el proceso: túnel



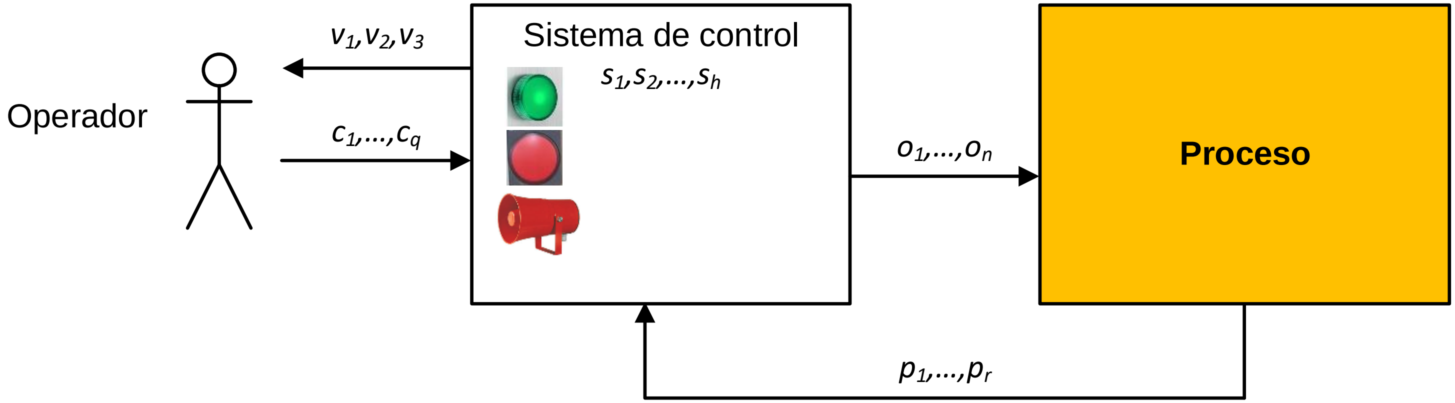
- **Tres fuentes de información sobre el proceso**
  - Conjunto $P_1 = \{\text{"No fuego"}, \text{"Fuego"}\}$
    - Asociado a sensor de detección de incendio
  - Conjunto $P_2 = \{\text{"No contaminación"}, \text{"Contaminación"}\}$
    - Asociado a sensor de contaminación
  - Conjunto $P_3 = \{\text{"No alto"}, \text{"Alto"}\}$
    - Asociado a sensor de nivel de agua
- **Variables**
  - $p_1 \in P_1$ asociada a sensor de detección
  - $p_2 \in P_2$ asociada a sensor de contaminación
  - $p_3 \in P_3$ asociada a nivel de agua

Ejemplo de receptores de órdenes: calefacción por aire



- **Dos receptores de órdenes (accionamientos o actuadores)**
  - Conjunto $O_1 = \{\text{"No ventilar", "Ventilar"}\}$
    - Asociado a ventilador
  - Conjunto $O_2 = \{\text{"No calentar", "Calentar"}\}$
    - Asociado a resistencia
- **Variables**
  - $o_1 \in O_1$ asociada a ventilador
  - $o_2 \in O_2$ asociada a resistencia

Ejemplo de medios para informar: panel de información

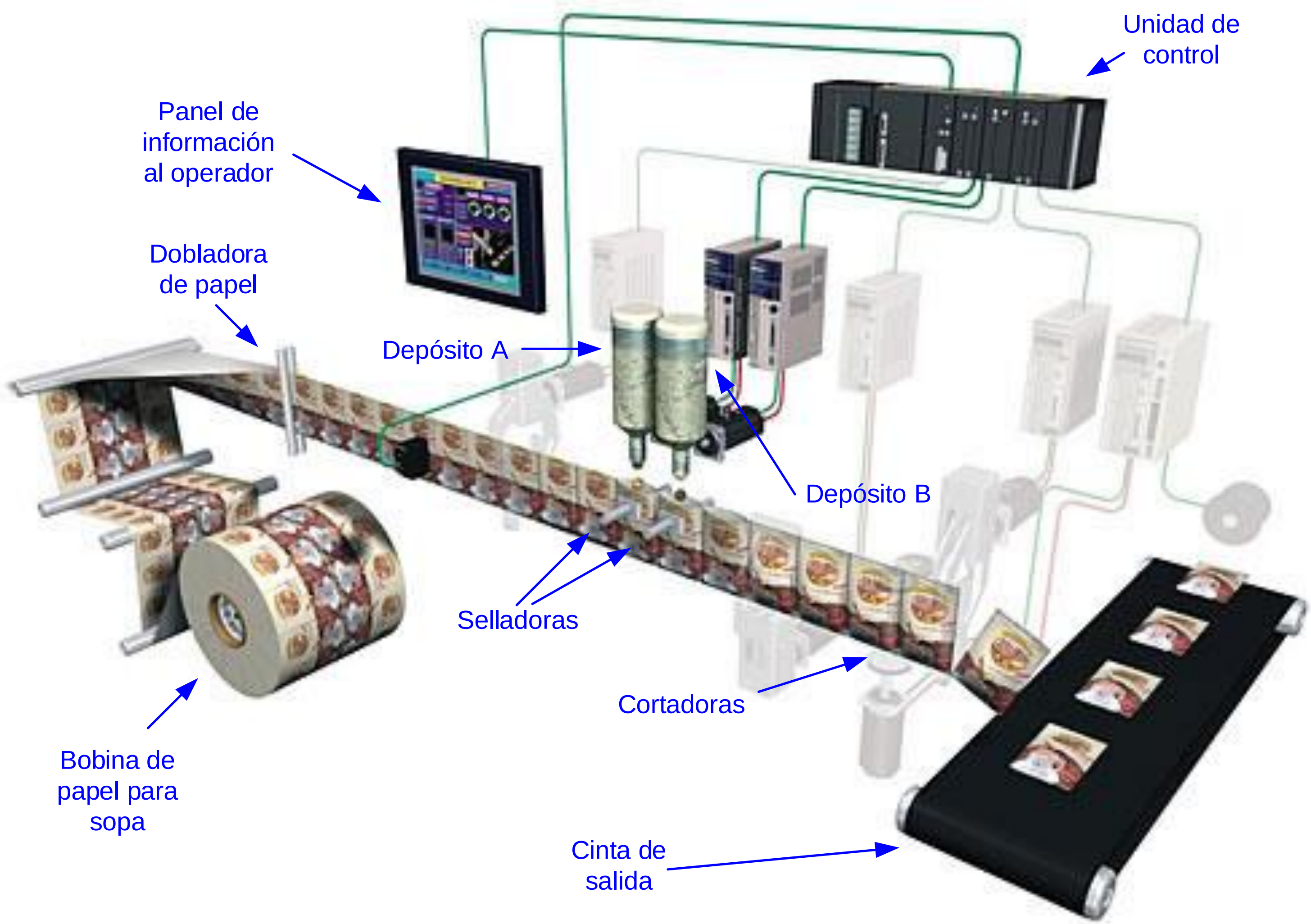


- **Tres medios para informar al operador**
  - Conjunto $V_1 = \{$"Sistema parado", "Sistema en marcha"$\}$ asociado a piloto verde
  - Conjunto $V_2 = \{$"No caliente", "Caliente"$\}$ asociado a piloto rojo
  - Conjunto $V_3 = \{$"Sin peligro", "Peligro"$\}$ asociado a bocina
- **Variables**
  - $v_1 \in V_1$ asociada a piloto verde
  - $v_2 \in V_2$ asociada a piloto rojo
  - $v_3 \in V_3$ asociada a bocina

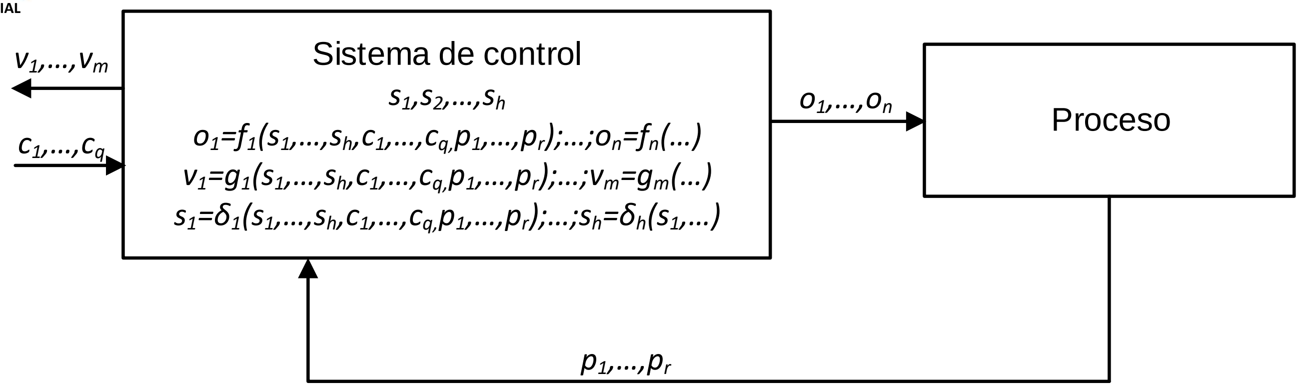
	Piloto no luce	Piloto luce
v_1	"Sistema parado"	"Sistema en marcha"
v_2	"No caliente"	"Caliente"

	No suena	Suena
v_3	"Sin peligro"	"Peligro"

Ejercicio: determinar variables con tipo y conjunto asociado



¿Cuántas funciones hay en un sistema automatizado?



- n **funciones de control:** f_i
  - Calculan el valor de las variables o_i: $o_i = f_i(s_1, \dots, s_h, c_1, \dots, c_q, p_1, \dots, p_r)$
  - $f_i: S_1 \times S_2 \times \dots S_h \times C_1 \times C_2 \times \dots C_q \times P_1 \times P_2 \times \dots P_r \to O_i$
- m **funciones de observación:** g_i
  - Calculan el valor de las variables v_i: $v_\mathrm{i} = g_i(s_1, \dots, s_h, c_1, \dots, c_q, p_1, \dots, p_r)$
    - $g_i: S_1 \times S_2 \times \dots S_h \times C_1 \times C_2 \times \dots C_q \times P_1 \times P_2 \times \dots P_r \to V_i$
- h **funciones de estado:** δ_i
  - Calculan el valor de las variables s_i: $s_i = \delta_i(s_1, \dots, s_i, \dots, s_h, c_1, \dots, c_q, p_1, \dots, p_r)$
    - $\delta_i: S_1 \times \dots \times S_i \times \dots \times C_1 \times C_2 \times \dots C_q \times P_1 \times P_2 \times \dots P_r \to S_i$
  - La variable está en ambos lados: dificultad para definir la función

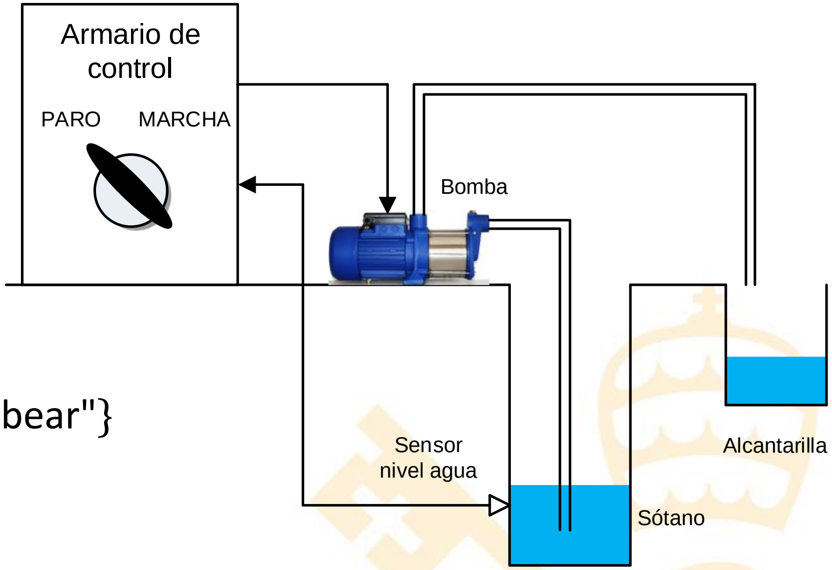
¿Cómo se determinan las funciones?

1. **Identificar**
   - Las fuentes de consigna y sus valores posibles
   - Las fuentes que informan sobre el proceso (sensores) y sus valores posibles
   - Los receptores de órdenes (accionamientos o actuadores) y sus valores posibles
   - Los medios para informar al operador y sus posibles valores
   - Las agrupaciones de estados internos y sus posibles valores (estados)
2. **Definir las variables**
   - Una variable tipo consigna por cada fuente de consigna
   - Una variable tipo proceso por cada fuente de información sobre el proceso
   - Una variable tipo orden por cada receptor de órdenes
   - Una variable tipo información por cada medio para informar al operador
   - Una variable tipo estado por cada agrupación de estados internos
3. **Definir el algoritmo de cálculo de las funciones de control, observación y de estado**
   - A partir de las variables definidas y los requisitos que deben cumplir las funciones
   - Método posible: tabla
     - Cada fila de la tabla: cada elemento del producto cartesiano

Ejemplo de función de control: sistema de achique (I)

MASTER EN INGENIERÍA INDUSTRIAL

- **Requisitos de la automatización: la bomba de achique sólo funciona si el conmutador está en posición de marcha y hay agua en el sótano**
- **Fuentes, receptores, …**
  - Fuentes de consigna:
    - Conmutador
    - Valores posibles $C = \{"\text{Paro}", "\text{Marcha}"\}$
  - Fuentes de información sobre el proceso:
    - Sensor de nivel
    - Valores posibles $P = \{"\text{No Agua}", "\text{Agua}"\}$
  - Receptores de órdenes:
    - Bomba
    - Valores posibles $O = \{"\text{No bombear}", "\text{Bombear}"\}$
- **Variables**
  - Variables de tipo consigna:
    - c asociada al conmutador: $c \in C$
  - Variables de tipo proceso:
    - p asociada al sensor de nivel: $p \in P$
  - Variables de tipo orden:
    - o asociada a bomba: $o \in O$



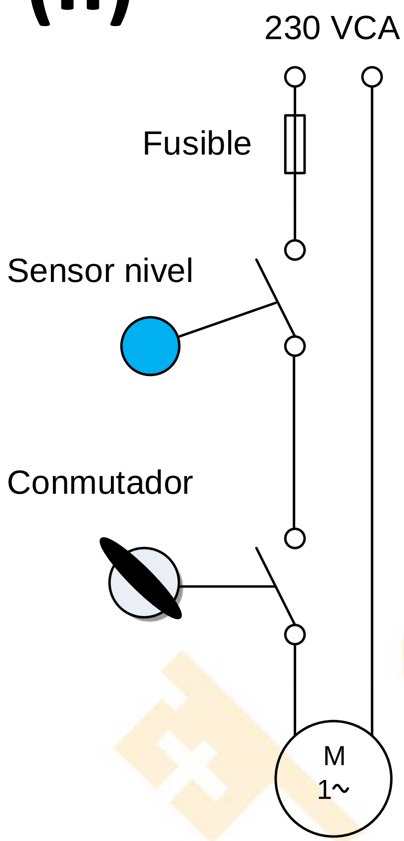
comillas.edu

Ejemplo de función de control: sistema de achique (II)

- **Función de control** $o = f(c, p)$
  - Cálculo mediante tabla:

c	p	o
"Paro"	*"No agua"*	*"No bombear"*
"Paro"	*"Agua"*	*"No bombear"*
"Marcha"	*"No Agua"*	*"No bombear"*
"Marcha"	*"Agua"*	*"Bombear"*

  - No hay otras funciones
- **Posible implementación**
  - Conmutador, contacto de conmutador: c
  - Sensor de nivel, contacto del sensor: p
    - Sube el nivel cierra el contacto
  - Motor que mueve la bomba de achique: o
    - Se aplica tensión: se mueve el motor
  - Función matemática y circuito son equivalentes
    - Variables son los elementos conectados
    - Función es el conexionado (conexión serie)
      - Las dos tablas son equivalentes

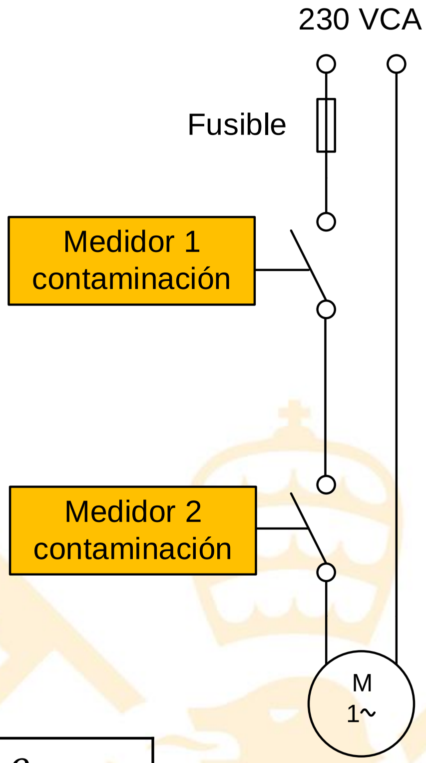


c	p	o
Abierto	*Abierto*	*Sin tensión*
Abierto	*Cerrado*	*Sin tensión*
Cerrado	*Abierto*	*Sin tensión*
Cerrado	*Cerrado*	*Con tensión*

Caso semejante: control de flujo de aire en un túnel

- **Requisitos: el ventilador funciona cuando los dos medidores de contaminación indican nivel alto**
- **Fuentes, receptores, estados y variables asociadas:**

Fuente, receptor, …	Valores posibles	Tipo variable	Variable
Medidor 1 contaminación	{"*No alta*","*Alta*"}	Proceso	p_1
Medidor 2 contaminación	{"*No alta*","*Alta*"}	Proceso	p_2
Ventilador	{"*No ventilar*","*Ventilar*"}	Orden	o



- **Función de control**
  - $o = f(p_1, p_2)$

p_1	p_2	o
"*No alta*"	"*No alta*"	"*No ventilar*"
"*No alta*"	"*Alta*"	"*No ventilar*"
"*Alta*"	"*No alta*"	"*No ventilar*"
"*Alta*"	"*Alta*"	"*Ventilar*"

- **Implementación: semejante a sistema de achique**
  - Cada medidor tiene un contacto
    - Variables independientes
  - Se conectan en serie con el motor
    - Función de control

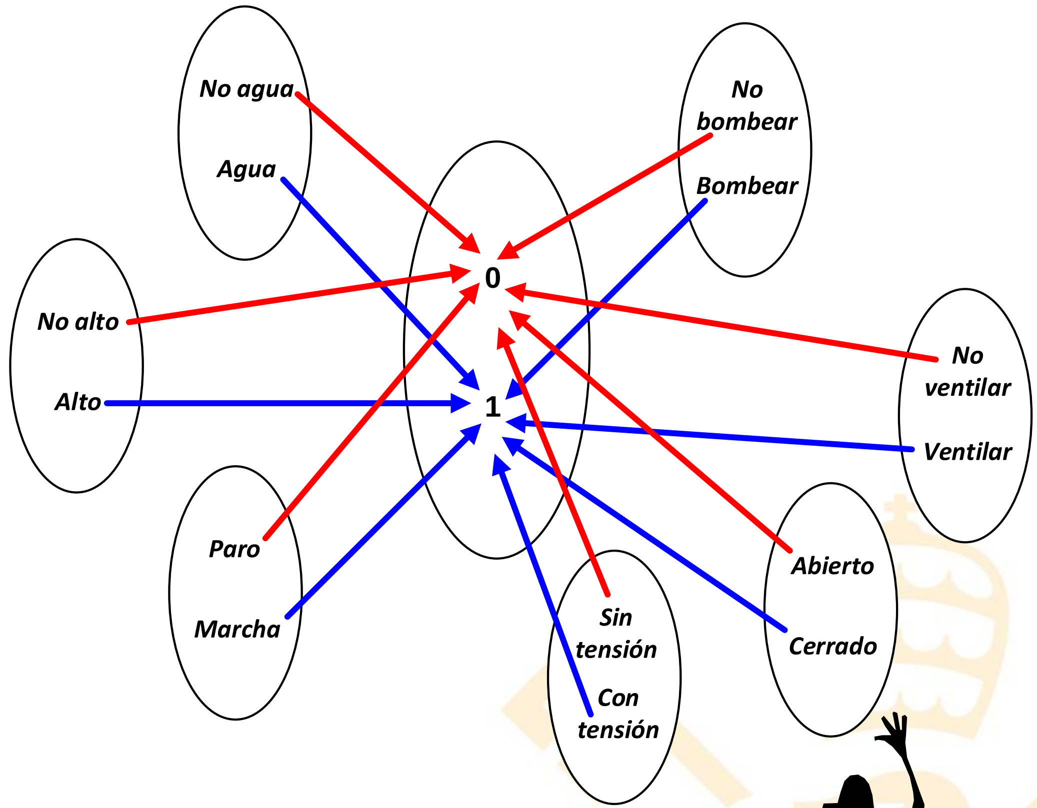
p_1	p_2	o
Abierto	*Abierto*	*Sin tensión*
Abierto	*Cerrado*	*Sin tensión*
Cerrado	*Abierto*	*Sin tensión*
Cerrado	*Cerrado*	*Con tensión*

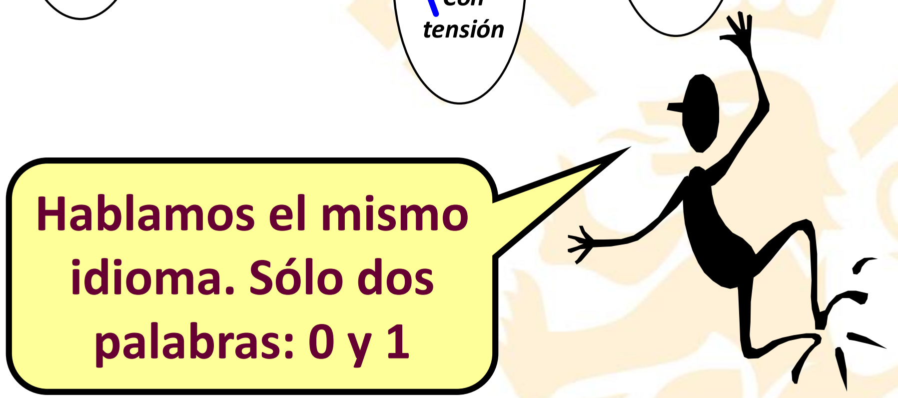
Enseñanzas de los dos ejemplos

- **Tablas de la función de control son muy parecidas**
- **Implementaciones son muy parecidas**
- **Problema: dificultad para generalizar**
  - Hablan idioma diferente
    - Fácilmente traducible
  - Las tablas son poco manejables
    - Si se pasa de 2 variables independientes a 5 variables: 32 filas
- **Solución para generalizar**
  - Primer paso: utilizar un idioma común
    - Con traducción a los idiomas propios
  - Segundo paso: utilizar operadores en vez de tablas
  - Tercer paso: utilizar expresiones que permitan combinar más de 2 variables independientes mediante operadores

Primer paso para generalizar: hablar el mismo idioma

- **Todos los conjuntos utilizan los mismos valores: 0 y 1**
  - Conjunto $B = \{0, 1\}$
- **Aplicación biyectiva para traducir desde cada realidad al conjunto B**
  - Mediante tabla
  - Asignación arbitraria
    - Típicamente 1 significa acción en las órdenes
- **Peligro: olvidarse de la realidad**
  - ¿El sistema hace lo que yo quiero o yo hago lo que el sistema quiere?





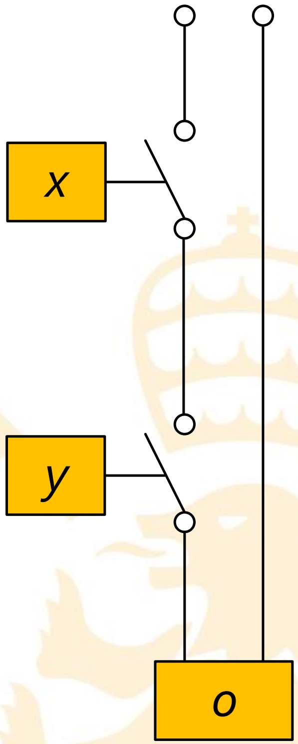
Primera consecuencia de utilizar el mismo idioma

- **Si dos funciones de control utilizan la misma tabla tienen el mismo patrón de implementación**
  - Igual conexionado
  - Se puede aplicar el mismo tipo de solución
- **Ejemplo (achique y ventilación)**

c	p	o
"Paro"	"No agua"	"No bombear"
"Paro"	"Agua"	"No bombear"
"Marcha"	"No Agua"	"No bombear"
"Marcha"	"Agua"	"Bombear"

p_1	p_2	o
"No alta"	"No alta"	"No ventilar"
"No alta"	"Alta"	"No ventilar"
"Alta"	"No alta"	"No ventilar"
"Alta"	"Alta"	"Ventilar"

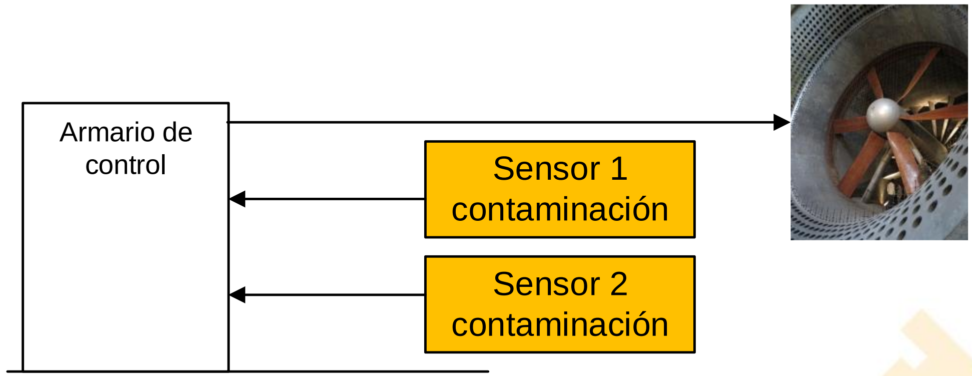
x	y	o
0	0	0
0	1	0
1	0	0
1	1	1



comillas.edu

Otro ejemplo: reducir contaminación (I)

- **Requisitos: el ventilador funciona cuando, al menos, uno de los dos medidores de contaminación en el túnel indica nivel alto (por simplicidad no hay conmutador)**



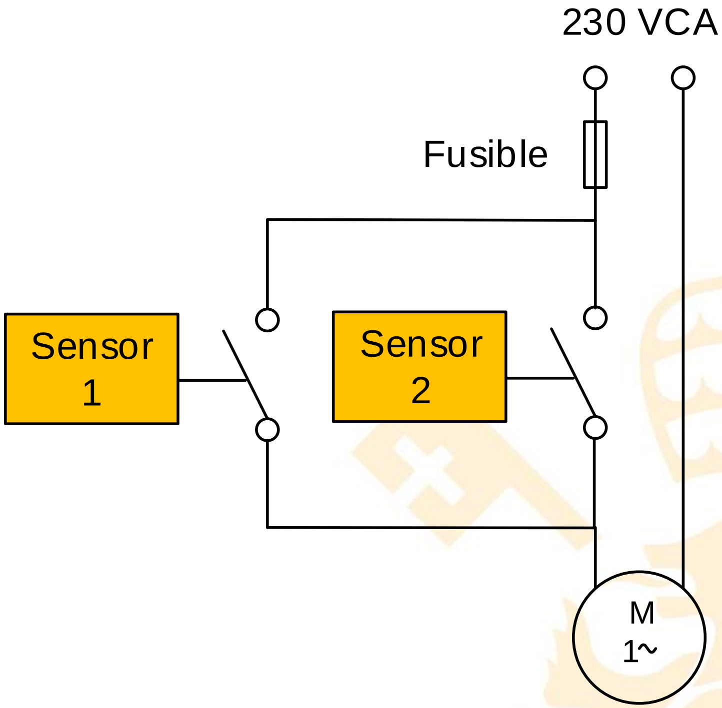
- **Variables**

Fuente, receptor, …	**Valores posibles**	**Tipo variable**	**Variable**	**0**	**1**
Sensor 1 contaminación	{"*No alta*","*Alta*"}	Proceso	p_1	"*No alto*"	"*Alto*"
Sensor 2 contaminación	{"*No alta*","*Alta*"}	Proceso	p_2	"*No alto*"	"*Alto*"
Ventilador	{"*No ventilar*", "*Ventilar*"}	Orden	o	"*No ventilar*"	"*Ventilar*"

Otro ejemplo: reducir contaminación (II)

- **Función de control**
  - $o = f(p_1, p_2)$

p_1	p_2	o
0	0	0
0	1	1
1	0	1
1	1	1



- **Implementación**
  - Sensores conectados en paralelo
    - Nuevo patrón de implementación

Segundo paso para generalizar: operadores en vez de tablas

- **Función de control** $o = f(x, y); x, y, o \in B; B = \{0,1\}$
  - Dos posibles patrones de combinación de x e y

x	y	o
0	0	0
0	1	1
1	0	1
1	1	1

x	y	o
0	0	0
0	1	0
1	0	0
1	1	1

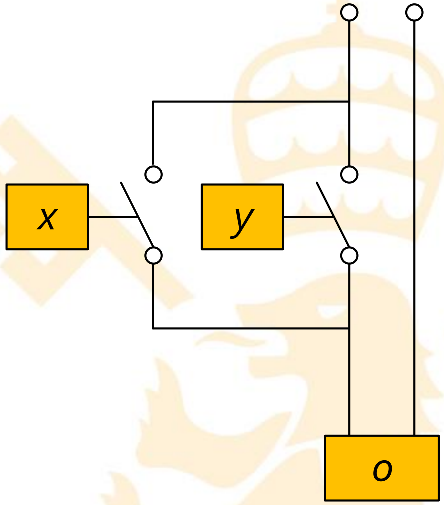
  - Estas tablas se denominan tablas de la verdad
- **¿Cómo indicar que x e y se combinan según uno de los patrones sin necesidad de utilizar la tabla?**
  - Como en una expresión aritmética: un símbolo entre las variables denominado operador aritmético
    - Ejemplo: $2 \cdot 3, 2 + 3$
  - Utilizando operadores lógicos

Operador lógico +

- **Nombre del operador: suma o suma lógica**
- **Expresamos** $o = f(x, y)$ **como** $o = x + y$ **para indicar que la función se calcula según la tabla:**

x	y	o
0	0	0
0	1	1
1	0	1
1	1	1

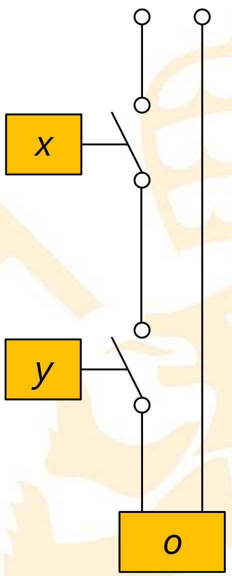
- **Se lee: *o* es la suma de** x **e** y
- **Desde el punto de vista de requisitos corresponde a expresiones del tipo "se ejecuta si al menos una"**
- **Implementación cableada: elementos representados por las variables conectados en paralelo**
- **Cumple la propiedad conmutativa** $x + y = y + x$
  - Cambiar el orden de las columnas no altera el resultado
  - Cambiar el orden de los contactos no altera el resultado



Operador lógico ·

- **Nombre del operador: producto o producto lógico**
- **Expresamos $o = f(x, y)$ como $o = x \cdot y$ para indicar que la función se calcula según la tabla:**
- **Se lee: *o* es el producto de x e y**
- **Desde el punto de vista de requisitos corresponde a expresiones del tipo "se ejecuta sólo cuando todas"**
- **Implementación cableada: elementos representados por las variables conectados en serie**
- **Cumple la propiedad conmutativa $xy = yx$**
  - Cambiar el orden de las columnas o de los contactos

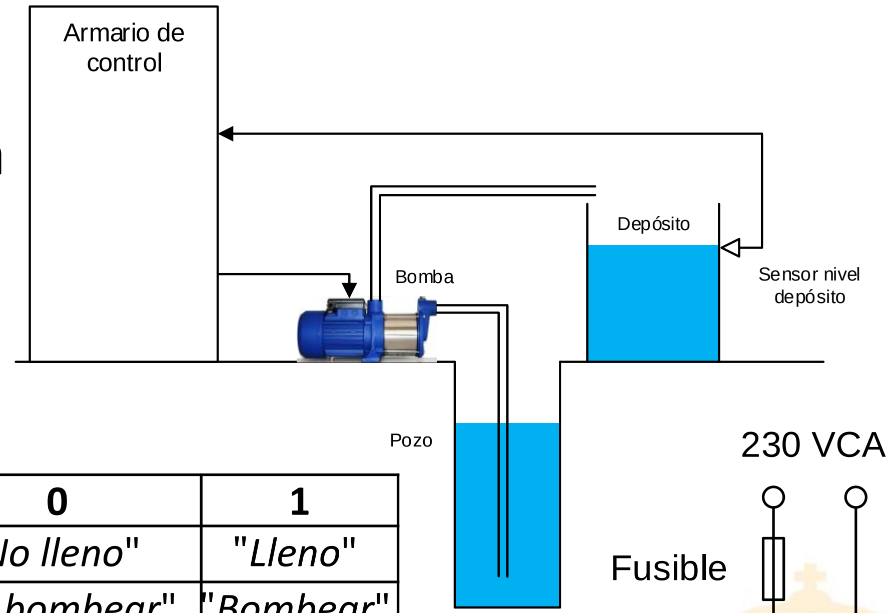
x	y	o
0	*0*	*0*
0	*1*	*0*
1	*0*	*0*
1	*1*	*1*



Otra situación no contemplada todavía

- **Requisitos: la bomba funciona cuando el sensor de nivel del depósito no indica lleno.**
- **Variables**

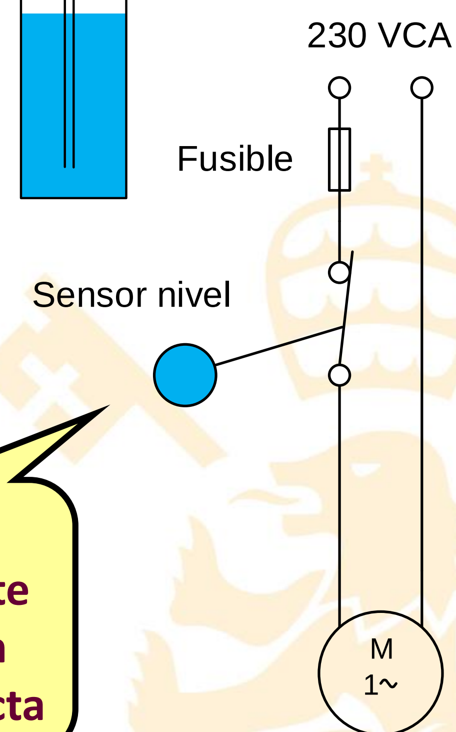
Fuente, receptor, …	Tipo variable	Variable	0	1
Sensor de nivel	Proceso	p	*"No lleno"*	*"Lleno"*
Bomba	Orden	o	*"No bombear"*	*"Bombear"*



- **Función de control**
  - Cuando uno sí el otro no

p	o
0	*1*
1	*0*

- **Implementación: al subir abre**
  - No cierra como en el sótano



Contacto normalmente cerrado: en reposo conecta

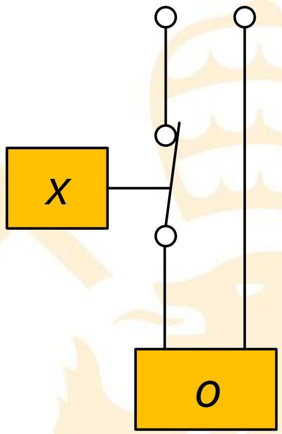
comillas.edu

Operador lógico ’

- **Nombre del operador: negado o complemento o producto lógico**
- **Expresamos $o = f(x, y)$ como $o = x'$ para indicar que la función se calcula según la tabla:**

x	o
0	1
1	0

- **Se lee: o es el negado (complementario) de i**
- **Desde el punto de vista de requisitos corresponde a expresiones del tipo “se ejecuta cuando no”**
- **Implementación cableada: contacto negado**



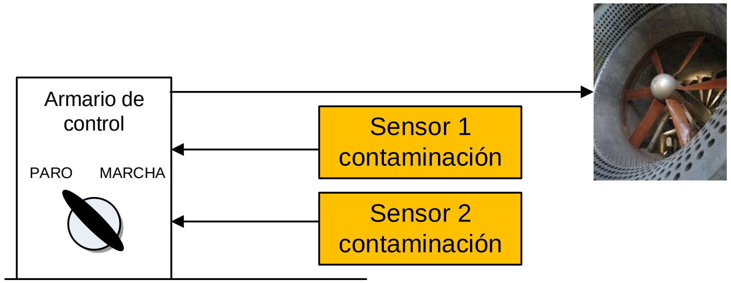
Tercer paso para generalizar: expresiones lógicas

MASTER EN INGENIERÍA INDUSTRIAL

- **¿Qué sabemos hacer hasta ahora?**
  - Identificar e implementar (cableado) funciones de control que correspondan con los siguientes tipos:
    - $o = p_1 + p_2$
    - $o = p_1 \cdot p_2$
    - $o = p'$
  - Esta forma de describir la función se denomina expresión lógica
    - Variables combinadas mediante operadores lógicos
- **¿Cómo trabajar con más de 2 variables?**
  - Descomponer en funciones intermedias de 2 variables
  - Los requisitos son los que indican cómo obtener las funciones intermedias
    - Pueden aparecer diferentes soluciones equivalentes

Ejemplo de 3 variables: función de control

- **Requisitos: el ventilador funciona cuando el conmutador está en posición de marcha y al menos uno de los dos medidores de contaminación indica nivel alto.**



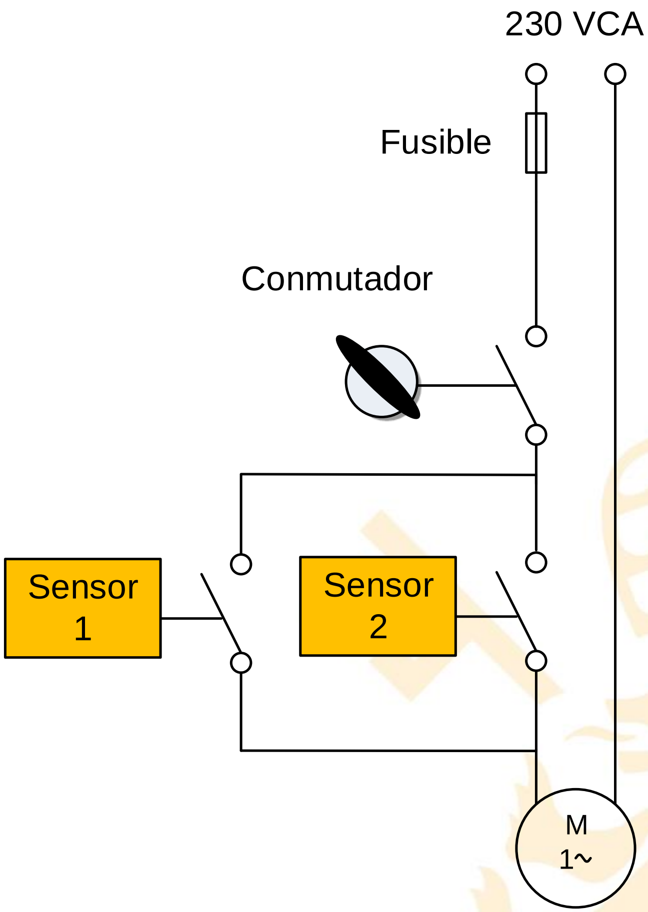
- **Variables**

Fuente, receptor, …	**Tipo variable**	**Variable**	**0**	**1**
Conmutador	Consigna	c	*"Paro"*	*"Marcha"*
Sensor 1 contaminación	Proceso	p_1	*"No alto"*	*"Alto"*
Sensor 2 contaminación	Proceso	p_2	*"No alto"*	*"Alto"*
Ventilador	Orden	o	*"No ventilar"*	*"Ventilar"*

- **Función de control**
  - Expresión lógica de la función intermedia: $t = p_1 + p_2$
    - “Al menos uno de los medidores indica nivel alto”
  - Expresión lógica de la función final: $o = c \cdot t = c(p_1 + p_2)$
    - “Conmutador en posición de marcha y …”

Ejemplo de 3 variables: implementación

- $o = c(p_1 + p_2)$
- **Implementación**
  - Serie de conmutador con paralelo de sensores
  - Expresión lógica y circuito tienen la misma topología
    - Tal como se lee la expresión lógica se organiza el circuito



Ejemplo de 3 variables: cumple la propiedad distributiva

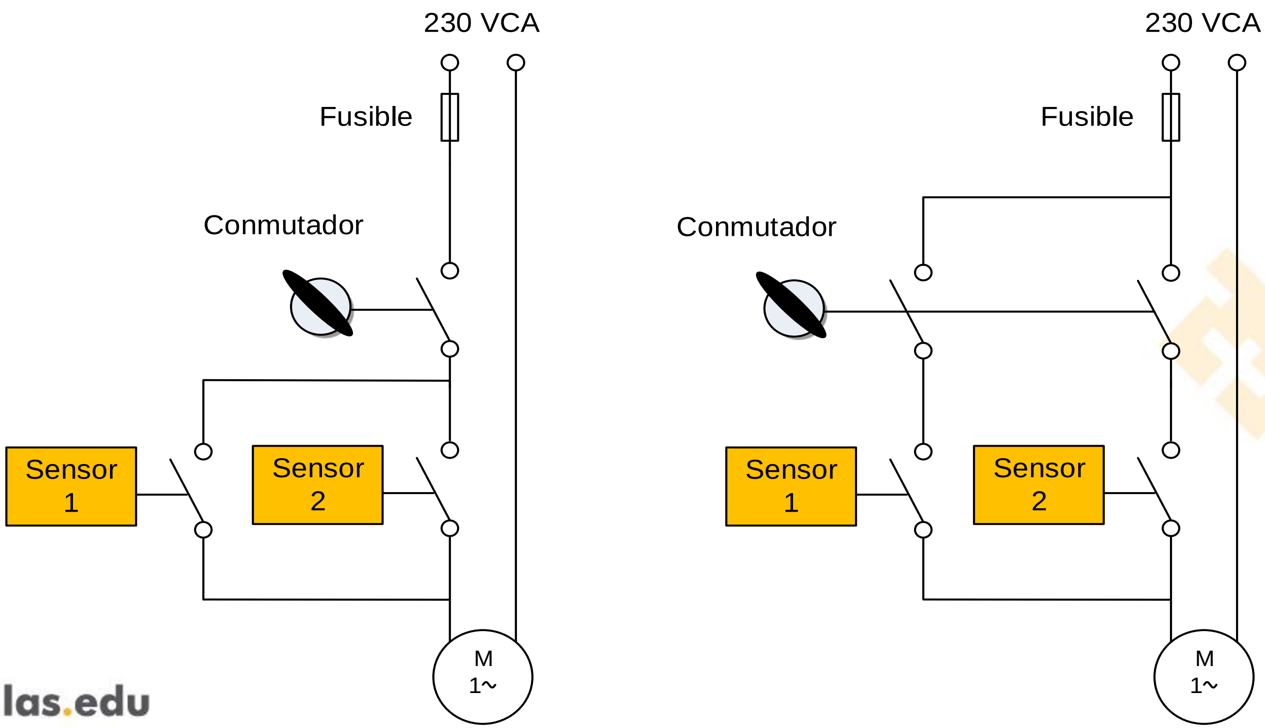
COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

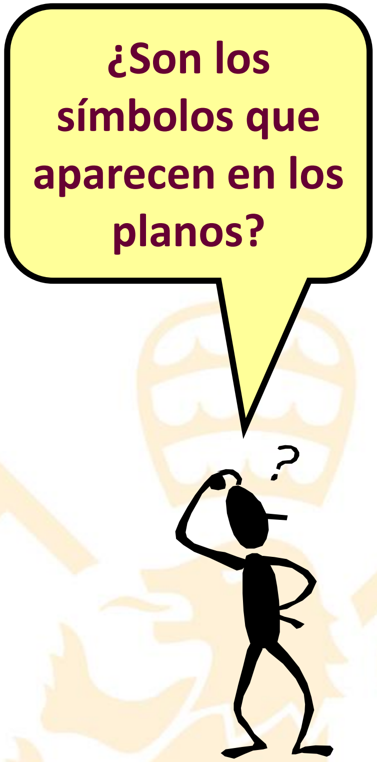
- $o = c(p_1 + p_2) = c \cdot p_1 + c \cdot p_2$
- **Demostración: tabla de la verdad igual**
  - Cada fila representa una combinación de valores en las variables
  - Se sustituyen las variables de la expresión lógica por sus valores en la fila y se operan según la expresión lógica

c	p_1	p_2	$p_1 + p_2$	o	$c \cdot p_1$	$c \cdot p_2$	o
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Propiedad distributiva también en la implementación

- $o = c(p_1 + p_2) = c \cdot p_1 + c \cdot p_2$
- **El conmutador mueve dos contactos en vez de uno**
  - Es más cara la implementación
  - A desarrollar en capítulo de automatismos cableados

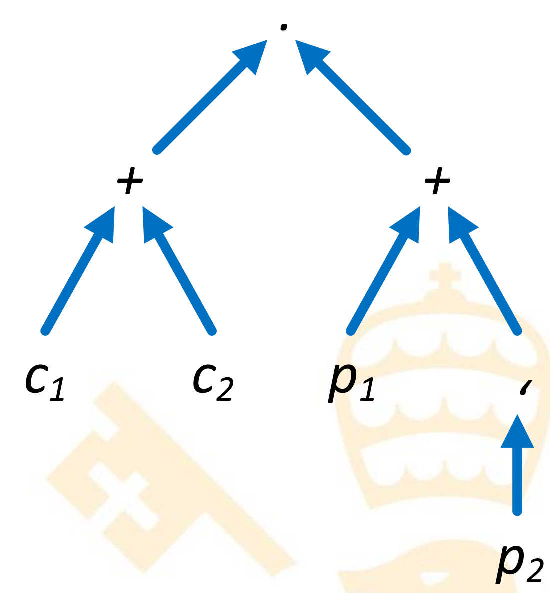




comillas.edu

Expresiones lógicas en general

- **Expresiones formadas por variables combinadas con los operadores lógicos +, •, ’**
  - Similares a las expresiones algebraicas
  - El orden de prioridad de los operadores es: ’, •, +
    - El resultado final es una jerarquía de operaciones lógicas de una o dos variables
  - Si es necesario se pueden utilizar paréntesis para especificar un orden concreto en las operaciones
    - Operación • implícita
- **Ejemplos de funciones calculadas como expresiones lógicas**
  - $f = c' (p_1{}' + p_2)$
  - $f = p_1 + p_2 + p_3 p_4$
  - $f = (c_1 + c_2)(p_1 + p_2{}')$



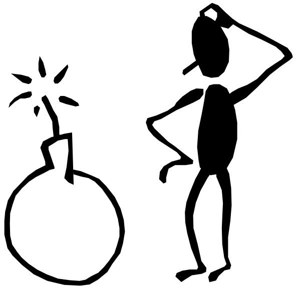
- **Tabla de la verdad y expresión lógica son dos caras de la misma función**
  - La expresión lógica es más sintética

Reflexión sobre lo visto hasta ahora

MASTER EN INGENIERÍA INDUSTRIAL

- **Todo problema se puede reducir a trabajar con el conjunto {0, 1}**
  - Cada sensor, cada consigna, cada orden tiene dos posibles valores
- **Podemos escribir la función mediante una expresión como si fuese una función convencional**
  - Variables
    - Representan cada sensor, cada consigna, cada orden al proceso, …
  - Operadores para combinar las variables: $+$, $\bullet$, $'$ y paréntesis
- **Las expresiones se pueden implementar**
- **Los operadores cumplen propiedades**
  - Conmutativa
    - $xy = yx$
    - $x + y = y + x$
  - Distributiva
    - $x(y + z) = xy + xz$

Sorpresa: potente aparato matemático disponible



- **El conjunto** *B={0, 1}* **junto con los operadores + , • , ’ es un ejemplo de álgebra de Boole (o álgebra booleana)**
  - Toda la artillería de las álgebras de Boole está disponible
  - También la implementación mediante interruptores es un ejemplo de álgebra de Boole
- **¿Por qué es un álgebra de Boole?**
  - Primero: es un álgebra
  - Segundo: cumple con la definición de un álgebra de Boole
- **Definición de álgebra**
  - Un álgebra es una estructura formada por:
    - Un conjunto de elementos $K = \{k_1, k_2, \ldots)$ (no vacío)
    - Un conjunto de operaciones $\phi = \{\varphi_1, \varphi_2, \ldots\}$ definidas sobre dicho conjunto ($\varphi_n: K \times K \times \cdots \to K$) y que cumplen unas ciertas propiedades
  - El álgebra se representa por (K, ϕ)

Definición de un álgebra de Boole

- **Un álgebra de Boole es un álgebra donde:**
  - K contiene al menos los elementos: 0, 1
  - Hay tres operaciones: +, •, $'$
  - Tales que $\forall a, b, c \in K$ se cumplen los Postulados de Huntington
    1. Conjunto cerrado: $a \cdot b \in K, a + b \in K, a' \in K$
    2. Elemento neutro: $a + 0 = a \,; a \cdot 1 = a$
    3. Ley conmutativa: $a + b$ = $b + a$; $a \cdot b$ = $b \cdot a$
    4. Ley distributiva:
       - $a \cdot (b + c) = a \cdot b + a \cdot c$
       - $a + (b \cdot c) = (a + b) \cdot (a + c)$
    5. Elemento complementario: $a + a' = 1; a \cdot a' = 0$
    6. Tamaño de K mayor o igual que 2: asegurado por los elementos 0 y 1
  - Hay otras definiciones equivalentes a los postulados

Teoremas útiles que cumple un álgebra de Boole: $\forall a, b, c \in K$

Principio de dualidad	Intercambio de operaciones y elementos neutros
Teorema de idempotencia	$a + a = a$ $a \cdot a = a$
Maximalidad del 1 y minimalidad del 0	$a + 1 = 1$ $a \cdot 0 = 0$
Teorema de absorción	$a + ab = a$ $a(a + b) = a$
Ley asociativa	$a + (b + c) = (a + b) + c$ $a(bc) = (ab)c$
Teorema de involución	$(a')' = a$
Teorema del complementario del elemento neutro	$0' = 1$ $1' = 0$
Leyes de De Morgan	$(a + b + c + \cdots)' = a'b'c' \ldots$ $(abc \ldots)' = a' + b' + c' + \cdots$
Consenso	$ab + a'c + bc = ab + a'c$ $(a + b)(a' + c)(b + c) = (a + b)(a' + c)$

Se demuestran a partir de los postulados o mediante tablas de la verdad

comillas.edu

MASTER EN INGENIERÍA INDUSTRIAL

Nuestras álgebras son álgebras de Boole: ¿Por qué?

- **Hemos “inventado” dos álgebras de Boole**
  - Álgebra 1: conexión de interruptores o equivalentes
    - Dos elementos: no permite el paso de la corriente (abierto), permite el paso de la corriente (cerrado)
    - Variables: elementos que pueden alternar entre permite o no permitir el paso de la corriente
    - Operadores: serie, paralelo, negado (falta mejorar la implementación)
  - Álgebra 2: abstracta
    - Dos elementos: 0 y 1
    - Variables: símbolo que puede tomar valor 0 o 1
    - Operadores: definidos mediante tablas
- **Se cumple tamaño del conjunto: sólo dos elementos**
  - Conexión: abierto, cerrado
  - Abstracto: 0, 1
- **Las operaciones definidas cumplen los postulados de Huntington**
  - Conexión: comprobando las conexiones
  - Abstracto: utilizando las tablas

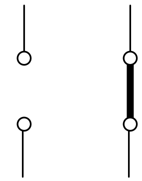
Nuestras álgebras son álgebras de Boole: nombre y uso

- **Nombre del álgebra:**
  - Conexión de interruptores: álgebra de contactos
  - Abstracta: "Álgebra de Boole"
    - Confusa si se habla con matemáticos
    - Típica en ambientes ingenieriles
    - Switching algebra en ambientes anglosajones
- **¿Qué va a permitir saber que es un álgebra de Boole?**
  - Simplificar las expresiones para una implementación más barata
  - Adecuar las expresiones para una mejor implementación

Nuestras álgebras son álgebras de Boole: isomorfismo

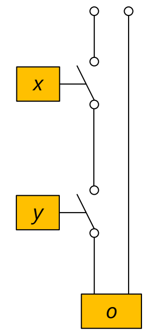
MASTER EN INGENIERÍA INDUSTRIAL

{0,1}



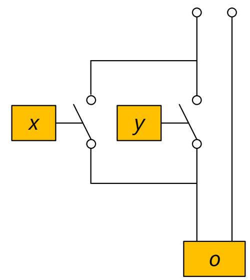
Nunca permite el paso de la corriente / Siempre permite el paso de la corriente

x	y	o=x•y
0	0	0
0	1	0
1	0	0
1	1	1



Isomorfismo: tienen la misma estructura

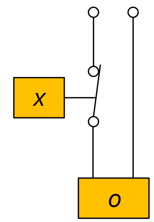
x	y	o=x+y
0	0	0
0	1	1
1	0	1
1	1	1



"Álgebra de Boole"

Álgebra de contactos

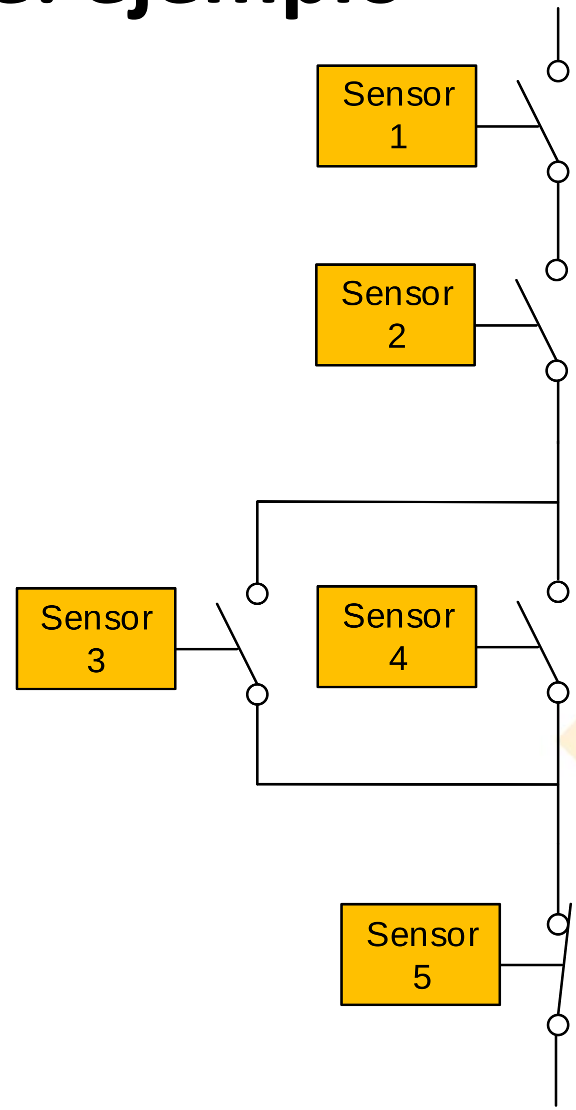
x	o=x'
0	1
1	0



comillas.edu

Nuestras álgebras son álgebras de Boole: ejemplo

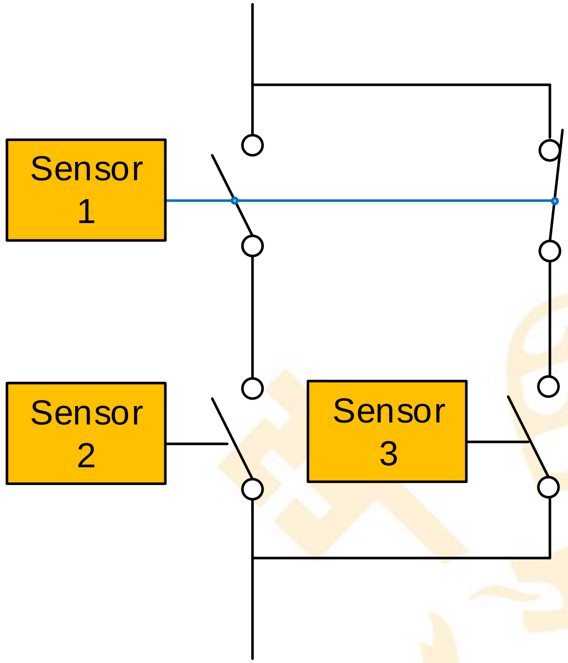
$$o=p_1 \bullet p_2(p_3+p_4)p_5'$$



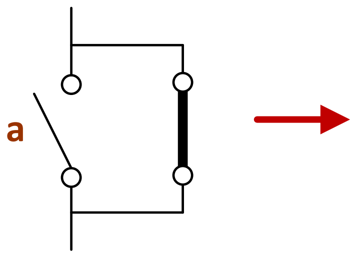
¿Qué ocurre si una variable aparece más de una vez?

MASTER EN INGENIERÍA INDUSTRIAL

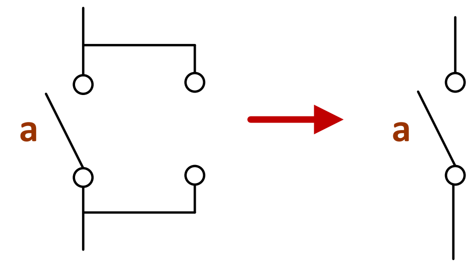
- **Desde el punto de vista abstracto**
  - No es un problema. Tantas veces como sea necesario.
- **Desde el punto de vista de la implementación**
  - Dependen de la tecnología
  - Ejemplo: $f = p_1 p_2 + p_1{}' p_3$
    - A tratar en el capítulo de automatismos cableados y programados



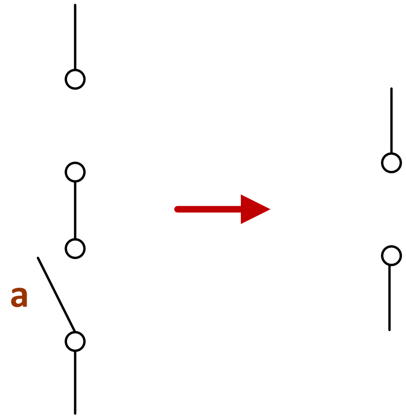
Algunas equivalencias entre propiedades (I)



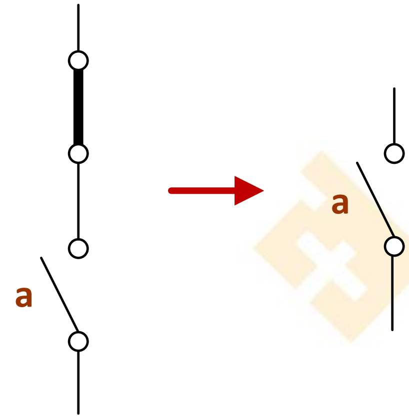
$a+1=1$



$a+0=a$

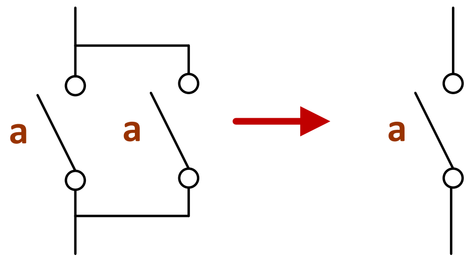


$a \bullet 0=0$

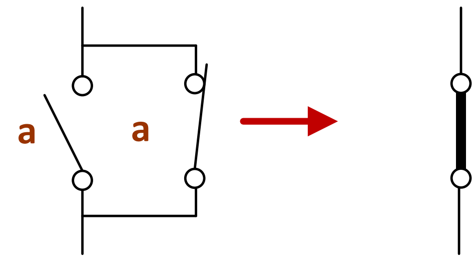


$a \bullet 1=a$

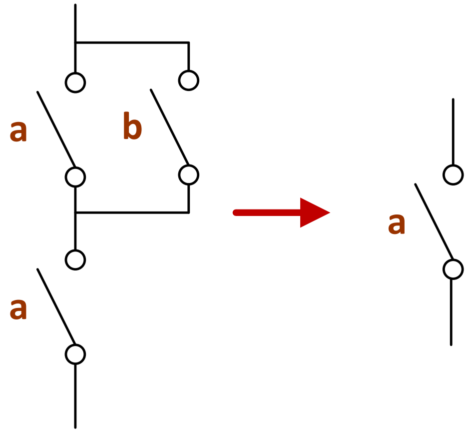
Algunas equivalencias entre propiedades (II)



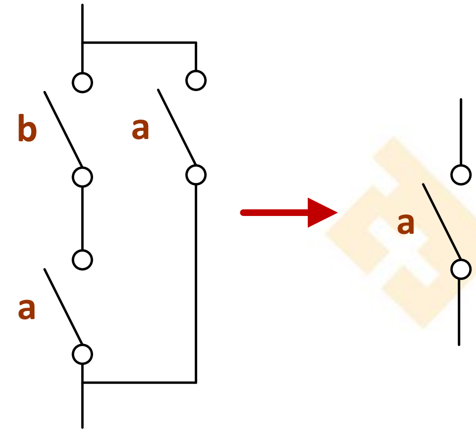
a+a=a



a+a’=1



a(a+b)=a



a+ab=a

comillas.edu

Isomorfismo con el cálculo proposicional (lógica)

MASTER EN INGENIERÍA INDUSTRIAL

- **Cálculo proposicional**
  - Elementos: falso (F), verdadero (T)
  - Variable: proposición lógica con dos posibles valores T o F
  - Operaciones: ∨, ∧, ¬
- **Isomorfismo entre cálculo proposicional, "Álgebra de Boole" (switching algebra), álgebra de contactos**
- **Diferencias mínimas entre cálculo proposicional (lógica) y "Álgebra de Boole"**
  - Variable Abierta en "Algebra de Boole" que indica:
    - 0: "No está abierta"
      - No significa que esté cerrada: puede estar entreabierta
    - 1: "Abierta"
  - Variable Abierta en Lógica que representa la proposición "Está abierta"
    - F: ""Está abierta" es falsa"
    - T: ""Está abierta" es verdadera"
- **El vocabulario de lógica se suele utilizar en "Álgebra de Boole"**

comillas.edu

¿Qué es un isomorfismo?

- **Dadas dos estructuras algebraicas (K_1, ϕ_1) y (K_2, ϕ_2) y una correspondencia h entre ellas:**
  - $h: (\mathrm{K}_1, \phi_1) \to (\mathrm{K}_2, \phi_2)$
- **Se dice que h es un isomorfismo cuando se cumple:**
  - $\forall a, b \in \mathrm{K}_1$ y $\forall r, s \in \mathrm{K}_2$
  - $h(a\phi_1 b) = h(a)\phi_2 h(b)$
  - $h^{-1}(r\phi_2 s) = h^{-1}(r)\phi_1 h^{-1}(s)$
- **Ejemplo:**
  - Variable lógica p_1 y p_2 que representan contacto de sensor 1 y contacto de sensor 2
  - La representación de $p_1 + p_2$ es convertir cada variable en un contacto y conectarlos en paralelo.
    - La inversa también funciona

Terminología a utilizar (I)

- **Conjunto binario (lógico)**
  - Conjunto formado por los elementos *{0, 1}*
    - Nombre: *B*
  - Todo conjunto B_i que sólo tenga dos valores donde se pueda establecer una única aplicación biyectiva con $B = \{0, 1\}$
- **Variable lógica (binaria, booleana)**
  - Variable cuyo valor es 0 o 1.
  - Variable cuyo valor sólo puede ser uno de los posibles valores de un conjunto binario

Terminología a utilizar (II)

- **Función lógica (binaria, booleana)**
  - Función establecida entre un producto cartesiano de conjuntos binarios y otro conjunto binario
    - Caso de conjuntos binarios cualquiera: $f: B_1 \times B_2 \times \cdots \to B_O$
    - Caso de *B*: $f: B^n \to B$
  - Puede ser expresada en función de variables lógicas: $b_i \in B$ (ó $b_i \in B_i$)
    - $f = f(b_1, b_2, \cdots)$, tantas variables como conjuntos formen el producto cartesiano
    - $b_1, b_2, \cdots$ son las variables independientes: no dependen unas de otras
      - También se denominan variables de entrada
    - f es la variable dependiente
      - También se denomina variable de salida
  - Dos métodos para indicar el algoritmo de cálculo de la función (hay más)
    - Expresión lógica
    - Tabla de la verdad

Terminología a utilizar (III)

COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

- **Expresión lógica**
  - Combinación entre variables lógicas utilizando los operadores + (suma), • (producto) y ′ (negado)
    - Similar a una expresión aritmética añadiendo el concepto de negado
  - La prioridad en las operaciones es de mayor a menor: ′, •, +
  - Se pueden utilizar paréntesis que tienen prioridad sobre el resto
  - Las variables pueden pertenecer a diferentes conjuntos binarios
    - La correspondencia con *B* resuelve las posibles ambigüedades.
  - Ejemplo: $f = xy + xy'z + x'yz$
- **Tabla de la verdad**
  - Tabla que indica el valor de la función para cada posible valor del producto cartesiano
    - Cada posible combinación de valores en las variables lógicas
  - Número de filas: $2^{Número\ de\ conjuntos\ del\ producto\ cartesiano}$
    - $2^{Número\ de\ variables\ lógicas}$
  - Ejemplo: tabla de la verdad de $f = xy + xy'z + x'yz$

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

comillas.edu

¿Cómo construir la tabla de la verdad de una función lógica?

- **Identificar las variables lógicas independientes a partir de los requisitos: sensores, consignas, … y la dependiente**
- **Preparar una tabla vacía:**
  - Columnas: variables lógicas más una para el valor de la función
  - Filas: $2^{Número\ de\ variables\ lógicas}$
- **Rellenar las columnas de variables con todas las combinaciones**
  - Siempre se utiliza una organización sistemática:
    - Columna de variables más a la derecha: alterna ceros y unos
    - Siguiente columna a la izquierda: alterna 2 ceros y 2 unos
    - Siguiente columna a la izquierda: alterna 4 ceros y 4 unos
    - Y así sucesivamente hasta la columna más a la derecha: el anterior duplicado
- **Por cada fila se calcula el valor de la función**
  - Se determina a partir de los requisitos qué valor debe tener la función para esa combinación de valores en las variables

Ejemplo de calcular la tabla a partir de la expresión lógica

- **Un piloto se enciende cuando sólo dos interruptores de los tres interruptores están cerrados:**
  - 3 variables lógicas, una por cada interruptor: i_1, i_2, i_3
  - Función lógica: *f* representa el piloto

i_1	i_2	i_3	f

→

i_1	i_2	i_3	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

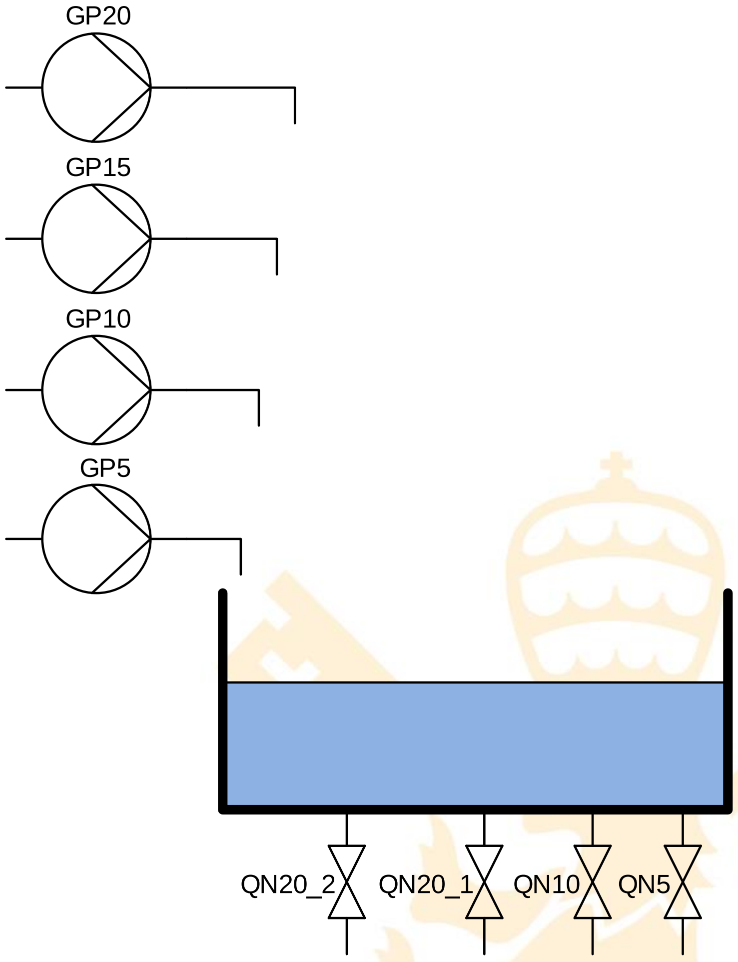
→

i_1	i_2	i_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Ejercicio control de 4 válvulas: tablas

COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

- **Un depósito intermedio es alimentado con cuatro bombas de 5, 10, 15 y 20 l/s. A la salida del depósito hay 4 válvulas de 5, 10, 20 y 20 l/s. La lógica de control de las válvulas es:**
  - El caudal teórico de salida del depósito debe ser mayor que el de entrada.
  - Se debe abrir el menor número de válvulas para que se cumpla el punto 1. En caso de varias posibilidades, se tomará aquella que tenga menor caudal de salida. Si aún hay varias posibilidades, son prioritarias las válvulas situadas más a la derecha del depósito.
- **Calcular la tabla de la verdad de cada válvula**



¿Cómo construir la expresión lógica de una función?

MASTER EN INGENIERÍA INDUSTRIAL

- **1. Identificar las variables lógicas independientes a partir de los requisitos: sensores, consignas, … y la dependiente**
- **2. O escribir la expresión directamente a partir de los requisitos**
  - Buscar expresiones en el texto que nos indiquen cómo combinar las variables:
    - “basta que uno”, “con uno que”, “al menos uno”… suma
    - “todos deben”, “no hay ninguno”… producto
    - “no debe estar”, “es el contrario de”… negación
  - Posible método para combinar las variables
    - Buscar los casos donde la función lógica debe tomar valor 1
    - Combinarlos mediante suma
- **2. O construir la tabla de la verdad y obtener la expresión lógica a partir de ella**

Ejemplo de escribir la expresión lógica directamente

- **Un piloto se enciende cuando sólo dos interruptores de los tres interruptores están cerrados:**
  - 3 variables lógicas, una por cada interruptor: i_1, i_2, i_3
  - Función lógica: f representa el piloto
  - Casos donde la función lógica debe tener valor 1:
    - i_1 e i_2 cerrados e i_3 abierto $\rightarrow i_1 i_2 i_3'$
      - Producto porque se deben cumplir que i_1 igual a 1, i_2 igual a 1 e i_3 igual 0
    - i_1 e i_3 cerrados e i_2 abierto $\rightarrow i_1 i_2' i_3$
    - i_2 e i_3 cerrados e i_1 abierto $\rightarrow i_1' i_2 i_3$
  - Si uno de los casos se da, la función debe tomar valor 1: suma de los casos
    - La función lógica es $f = i_1 i_2 i_3' + i_1 i_2' i_3 + i_1' i_2 i_3$

comillas.edu

Cuidado al escribir la expresión directamente

MASTER EN INGENIERÍA INDUSTRIAL

- **Peligro: no traducir correctamente el "espíritu" del texto a la expresión lógica**
- **Ejemplo de error**
  - Un piloto se enciende cuando sólo dos interruptores de los tres interruptores están cerrados:
    - El piloto se enciende cuando se cumpla alguno de estos casos:
      - i_1 e i_2 cerrados $\rightarrow i_1 i_2$
      - i_1 e i_3 cerrados $\rightarrow i_1 i_3$
      - i_2 e i_3 cerrados $\rightarrow i_2 i_3$
    - La función lógica es:
      - $f = i_1 i_2 + i_1 i_3 + i_2 i_3$
    - Error: con los tres interruptores cerrados también se enciende
      - Los requisitos dicen: "sólo dos interruptores"
        - $i_1 i_2$ no recoge la condición sólo interruptor 1 e interruptor 2 cerrados
        - $i_1 i_2$ significa interruptor 1 e interruptor 2 cerrados con independencia del estado de interruptor 3: $i_1 i_2 = i_1 i_2 (i_3 + i_3') = i_1 i_2 i_3 + i_1 i_2 i_3'$
- **La tabla de la verdad no tiene este problema**
  - Se analiza cada combinación de forma exhaustiva

Ejercicio de escribir la expresión lógica directamente

- **En un parque eólico con 3 molinos, se quiere visualizar en la caseta general cuántos molinos están girando mediante 2 pilotos (amarillo y verde). Cuando un molino gira cierra un contacto asociado. Cada piloto indica la siguiente información:**
  - Todos apagados: ningún molino girando
  - Amarillo: hay sólo un molino girando
  - Verde: hay más de un molino girando
- **Escribir directamente la expresión de la función lógica que controla los pilotos amarillo y verde**

Obtener la expresión lógica desde una tabla de la verdad

- **Problema: una misma función lógica tiene diferentes expresiones lógicas equivalentes**
  - Ejemplo: *f=x+yz* y y *f=(x+y)(x+z)* son expresiones lógicas equivalentes
  - Una función lógica tiene una única tabla de la verdad
    - La tabla de la verdad permite comprobar que dos expresiones son equivalentes
- **Hay estructuras de expresiones lógicas representativas:**
  - Suma de productos de variables
    - Ejemplo: *f=xy+xy'z+x'yz*
    - En cada sumando intervienen todas o parte de las variables sin haber repetición
  - Producto de sumas de variables
    - Ejemplo: *f=(x+y+z)(x+y'+z)(x'+y+z)*
    - En cada producto intervienen todas o parte de las variables sin haber repetición
  - Combinación de ambas
    - Ejemplo: *f=xy+(x+z')(y+x'z)*
  - Cada suma o producto recibe el nombre de término

Expresiones lógicas canónicas

- **Suma de productos canónica y producto de sumas canónica**
  - En cada producto o suma intervienen todas las variables de la función lógica sin haber repetición
- **Muy fáciles de construir desde la tabla de la verdad:**
  - Suma de productos canónica (SOP)
    - Tantos sumandos como filas donde la función es 1
    - Cada sumando que representa una fila donde la función es 1 es el producto de cada variable lógica tal cual o negada según la siguiente regla
      - Tal cual si la variable toma valor 1 en esa fila
      - Negada si la variable toma valor 0 en esa fila
  - Producto de sumas canónica (POS)
    - Tantos producto como filas donde la función es 0
    - Cada producto que representa una fila donde la función es 0 es la suma de cada variable lógica tal cual o negada según la siguiente regla
      - Tal cual si la variable toma valor 0 en esa fila
      - Negada si la variable toma valor 1 en esa fila

Ejemplo de obtener la suma de productos canónica

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

x'yz'
x'yz
xy'z
xyz'
xyz

f=x'yz'+x'yz+ xy'z +xyz'+ xyz

Idea: El resultado es 1 si alguna de las combinaciones para las cuales es 1 se da

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

Ejemplo de obtener el producto de sumas canónico

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$x+y+z$

$x+y+z'$

$x'+y+z$

$f = (x+y+z)(x+y+z')(x'+y+z)$

Idea: El resultado es 1 si ninguna de las combinaciones para las cuales es 0 se da

$f = (x'y'z')'(x'y'z)'(xy'z')'$

Ejercicio control de 4 válvulas

- **Determinar la expresión SOP de la válvula QN5**
  - Suma de productos
- **Determinar la expresión POS de la válvula QN20_1**
  - Producto de sumas

¿Qué expresión lógica interesa?

COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

- **¿Por qué interesa la expresión lógica?**
  - Más fácil de implementar que la tabla de la verdad
    - Isomorfismo con el álgebra de contactos
    - Programación
  - Problema: múltiples expresiones lógicas no triviales para una misma función lógica
    - Ejemplo: $f=x'yz'+x'yz+ xy'z +xyz'+ xyz$ e $f=y+xz$ son equivalentes
- **Criterios para elegir la expresión lógica adecuada**
  - Coste de implementación: más barata
    - Mínimo de operaciones
      - Es independiente de la tecnología utilizada
  - Mantenimiento
    - Más fácil de entender la lógica que representa
      - Más fácil de depurar
      - Puede que no sea la más barata

comillas.edu

Obtener una expresión lógica mínima: simplificar

- **Objetivo:**
  - Expresiones lógicas con un número menor de variables y operaciones
    - El óptimo es difícil de obtener
- **Métodos de simplificación**
  - Manipulación algebraica utilizando los teoremas
    - Muy laborioso
  - Mapas de Karnaugh
    - Método gráfico
    - Número limitado de variables: no más de 6.
  - Métodos más sofisticados como Quine-McCluskey
    - Mediante programas
- **Problema general asociado a la simplificación:**
  - No hay una única solución
    - $f=ac'd+a'd(b+c)$ ó $f=ad(c'+b)+a'dc$ son equivalentes
      - Cualquiera de ellas vale

Ejemplo de simplificación algebraica

- $f=x'yz'+x'yz+xy'z+xyz'+xyz$
  1. Distributiva: $f=x'y(z'+z)+xy'z+xy(z'+z)$
  2. Complementario: $f=x'y+xy'z+xy$
  3. Asociativa, distributiva: $f=y(x'+x)+xy'z$
  4. Complementario, conmutativa: $f=y+xy'z=y+y'xz$
  5. Distributiva: $f=(y+y')(y+xz)$
  6. $\boldsymbol{f=y+xz}$
- $f=(x'y'z'+x'y'z+xy'z')'$
  1. Asociativa y distributiva: $f=(x'y'(z'+z)+xy'z')'$
  2. Complemento: $f=(x'y'+xy'z')'$
  3. Leyes de De Morgan: $f=(x'y')'(xy'z')'$
  4. Leyes de De Morgan: $f=(x+y)(x'+y+z)$
  5. $f=xx'+xy+xz+yx'+yy+yz$
  6. $f=xz+y+xy+yx'+yz$
  7. $f=xz+y(1+x+x'+z)$
  8. $\boldsymbol{f=xz+y}$ Es equivalente a la de arriba

comillas.edu

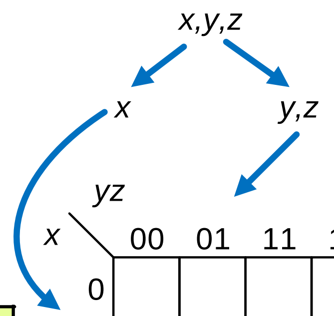
Simplificación mediante mapas de Karnaugh

- **Se basa en el concepto de productos adyacentes entre sí en una suma de productos**
- **Varios productos son adyacentes entre sí cuando tienen parte de las variables en el mismo estado de complemento y el resto con todas los posibles combinaciones**
  - $xy'z'+xy'z$
    - Son productos adyacentes
      - x e y en el mismo estado de complemento: xy'
      - todas las posibles combinaciones de z
  - $v'xy'z'+v'xyz'+v'xy'z+v'xyz+vx'yz$
    - $v'xy'z'+v'xyz'+v'xy'z+v'xyz$ son productos adyacentes
    - $vx'yz$ no es adyacente con el resto
- **Cada grupo de productos adyacentes se simplifica por un solo producto con las variables que no cambian**
  - $xy'z'+xy'z=xy'(z'+z)=xy'$
  - $v'xy'z'+v'xyz'+v'xy'z+v'xyz=v'x\,(y'z'+yz'+y'z+yz)=v'x$
  - Los conjuntos de productos tienen que ser potencia de 2
- **Para buscar fácilmente los productos adyacentes se reordena la tabla de la verdad para que los posibles productos sean adyacentes entre sí**
  - Al ir de una celda a otra en horizontal o en vertical sólo cambia una variable
  - Esta tabla recibe el nombre de tabla o mapa de Karnaugh.

¿Por qué el número de productos adyacentes tiene que ser potencia de 2?

Método para simplifica por Karnaugh (de 2 a 4 variables)

- **Dividir las variables en dos grupos**
  - El primer grupo representa las filas: *r* variables
  - El segundo grupo representa las columnas: *c* variables
- **Dibujar una matriz de 2^r filas y 2^c columnas**
  - Las filas representan todas las combinaciones posibles de las *r* variables
  - Las columnas representan todas las combinaciones posibles de las *c* variables
- **Etiquetar cada fila y cada columna con la combinación en binario que representa de tal manera que al ir de una fila a otra las combinaciones son adyacentes**
  - Sólo cambia una de las variables
  - Cada celda de la matriz representa una combinación de las *(r+c)* variables
    - Cada celda es adyacente con sus vecinas en vertical y horizontal
- **Etiquetar con 1 cada celda de la matriz cuya combinación haga que la función lógica tome valor 1**
  - Requisitos, tabla de la verdad, expresión lógica
- **Etiquetar el resto de celdas con 0**
- **Formar grupos (paralelogramos) con las celdas que tienen 1**
  - Máximo número de elementos con los lados potencia de 2
  - Si es necesario, reutilizar casilla de otro grupo para ampliar el tamaño del grupo
- **Escribir el producto correspondiente a cada grupo:**
  - Sólo las variables que no cambian de valor
  - Si está a 1 la variable se escribe tal cual, y si está a 0, se complementa.
- **Sumar los productos**



x\yz	00	01	11	10
0				
1				

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

x\yz	00	01	11	10
0	0	0	1	1
1	0	1	1	1

x\yz	00	01	11	10
0	0	0	1	1
1	0	1	1	1

xz y

f=x'yz'+x'yz+ xy'z +xyz'+ xyz → ***f=xz+y***

Ejemplos de simplificación por Karnaugh para 4 variables

Adyacentes

ab \ cd	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	1	1
10	1	1	1	1

f = a+b'd'+bc'd

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	1	0	1
10	0	0	1	0

f = abd'+abc'+a'cd+b'cd

Dada una función lógica de n variables independientes: ¿Cuántos productos tendrá el caso menos simplificable?

Todavía se puede realizar una pequeña simplificación algebraica: f=ab(d'+c')+cd(a'+b')

comillas.edu

Ejercicio: Control de 4 válvulas

- **Obtener las expresiones de las funciones lógicas que controlan las 4 válvulas utilizando mapa de Karnaugh**
- **¡Ojo!**
  - Si cambia más de 1 variable entre filas o columnas adyacentes en un mapa de Karnaugh
  - Grupos que no son potencias de 2

comillas.edu

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

Tarjeta roja

- **Cuando en la respuesta a una pregunta de examen o prueba se detecta una tarjeta roja:**
  - Se para la corrección a la pregunta
    - No importa lo que venga a continuación en la solución a la pregunta
  - Se califica la pregunta con 0
  - Se sigue la corrección por la siguiente pregunta sin tener en cuenta que la anterior tenía una tarjeta roja

comillas.edu

Ejemplo de simplificación por Karnaugh usando don't care

- **Uso de don't care**
  - Son combinaciones de entradas que teóricamente no se dan
  - Se señalan como X en el mapa de Karnaugh
  - Pueden ser utilizadas como 0 o como 1 para formar grupos mayores
    - Nunca se forman grupos de X's aisladas
- **En Automatización Industrial es discutible que una combinación nunca aparezca**
  - Se suele preparar una función adicional de error

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	X	X	X
10	1	1	1	1

f = a+bd

Usando los don't care

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	**1**	**1**	**1**
10	1	1	1	1

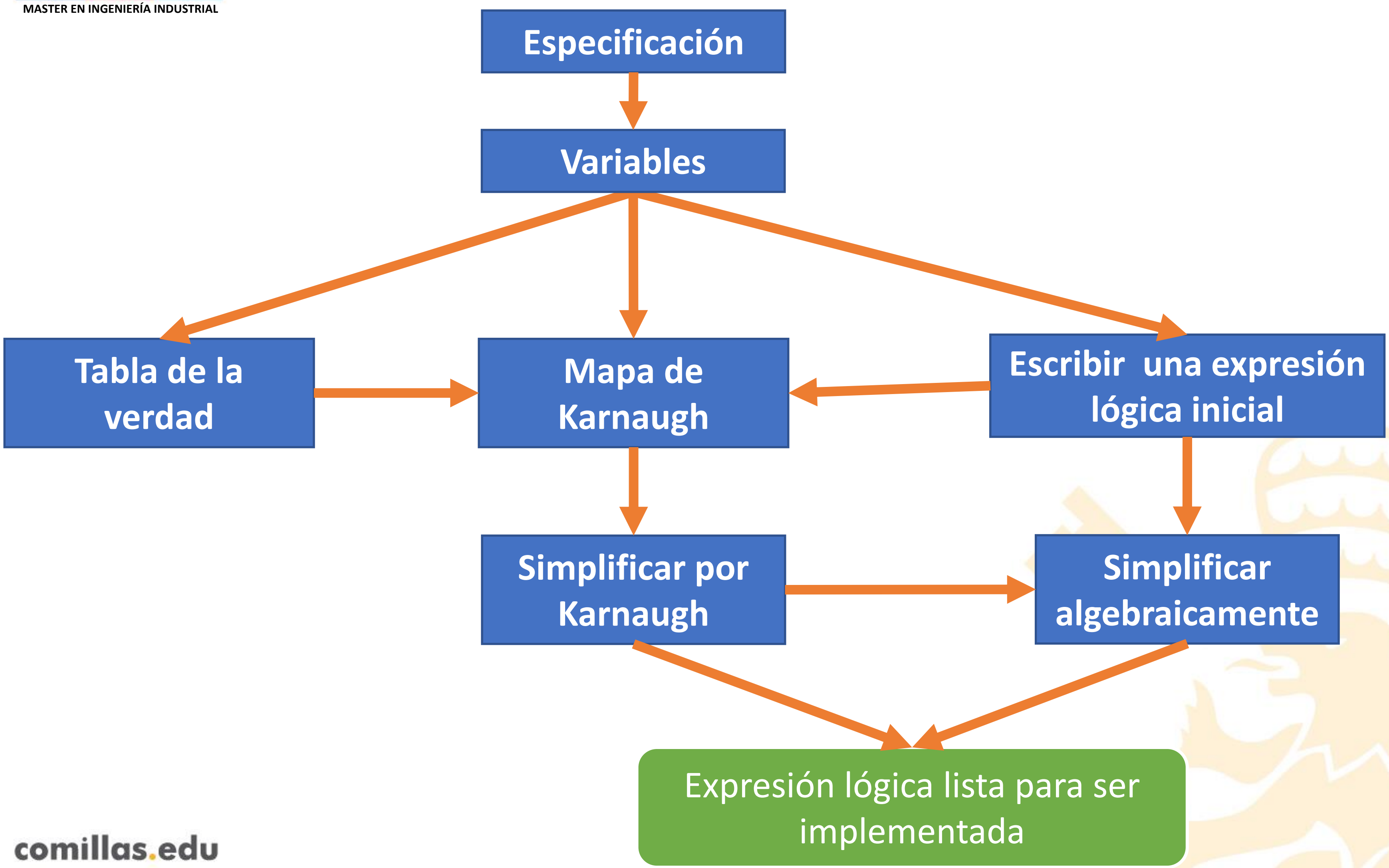
f = ac'd'+a'bd+ab'

e =abd+abc

Función de error

comillas.edu

Resumen cálculo de la expresión lógica



Comparación de métodos para calcular la expresión lógica

MASTER EN INGENIERÍA INDUSTRIAL

- **A través de tabla (verdad o Karnaugh)**
  - A favor: todas las posibilidades son analizadas
  - En contra: hasta 4 variables (5 y 6 también se puede manejar)
    - Tamaño exponencial
- **Escribiendo directamente**
  - A favor: no hay limitación de variables
  - En contra:
    - Posibilidad de no tener en cuenta alguna condición
    - La expresión puede ser compleja
      - Se puede simplificar
- **¿Cómo son las funciones lógicas en automatización?**
  - Muchas variables
  - Muy importante el criterio de seguridad
    - Método para escribirlas

Mejor que NO FUNCIONE cuando debe que FUNCIONE cuando no debe

Solución: escribir la función con criterios de seguridad

COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

- **Paso 1: Identificar todas las variables asociadas a la seguridad**
  - La variable con valor 1 indica que no se cumple el criterio de seguridad asociado
    - Muy importante saber qué significa 1 y qué significa 0
  - Ejemplo:
    - c_1: seta de emergencia (c_1 igual a 1 indica seta de emergencia activada)
    - p_1: protección térmica del motor (p_1 igual a 1 indica protección térmica disparada)
    - p_2: detector de humos (p_2 igual a 1 indica que el detector detecta humo)
- **Paso 2: Producto del negado de esas variables**
  - Todos los criterios de seguridad se deben satisfacer
  - Ejemplo: $f = c_1' p_1' p_2'$
- **Paso 3: Multiplicar por la suma de las expresiones lógicas de las variables que indican cuando la función lógica debe tomar valor 1 una vez tenidas en cuenta las variables asociadas a criterios de seguridad**
  - Ejemplo:
    - c_2: pulsador de marcha 2
    - c_3: pulsador de marcha 3
    - $f = c_1' p_1' p_2' (c_2 + c_3)$
      - La orden (ejemplo: aspirar) se activa si la seta de emergencia no está pulsada, la protección térmica no está activada, el detector de humos no detecta humo y hay una orden positiva del operador (pulsador 2 o 3)

Escribir en general la función con criterios de seguridad

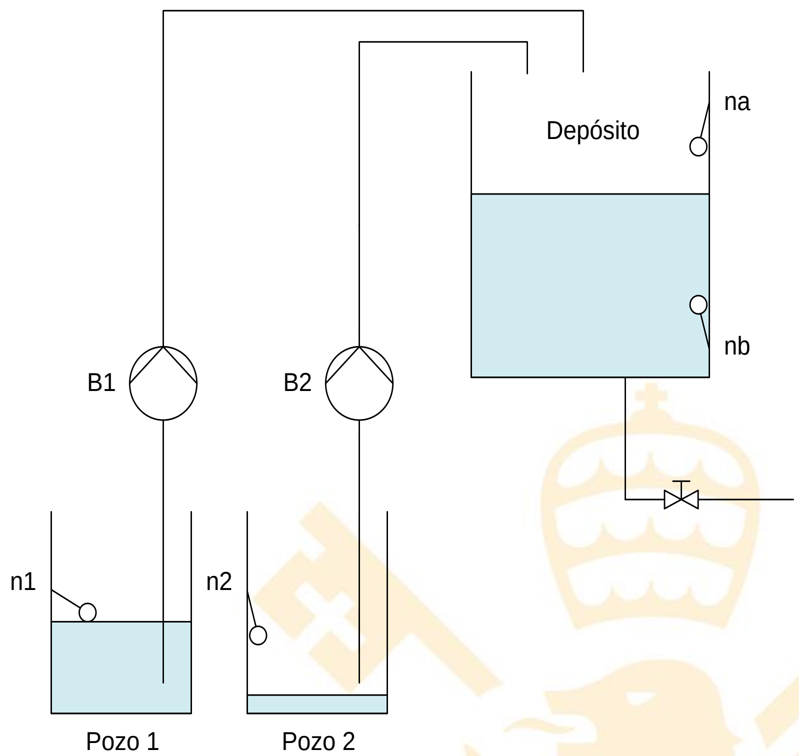
COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

- **Determinar todas las condiciones de seguridad (CSx) que se deben cumplir antes de poder darse una orden**
  - Ejemplo de condiciones de seguridad:
    - Seta de emergencias en condiciones de reposo
    - Detectores de humo que no detectan humo
    - Al menos dos sistemas disponibles para ejecutar la orden
    - Envolvente de seguridad de la máquina cerrada
    - Agua en el pozo desde donde hay que bombear
- **Determinar las condiciones (CMx) para que la orden se ejecute una vez cumplidos los criterios de seguridad**
  - Ejemplo de condiciones para que la orden se ejecute
    - El nivel de agua en el depósito es bajo
    - La temperatura es insuficiente
    - No hay llegado la caja al extremo de la cinta transportadora
- **Se construye la expresión de la función lógica como:**
  - f = CS1·CS2· …·CSn (CM1+CM2+ … +CMm)
    - CS1·CS2· …·CSn se suele denominar cadena de seguridad

comillas.edu

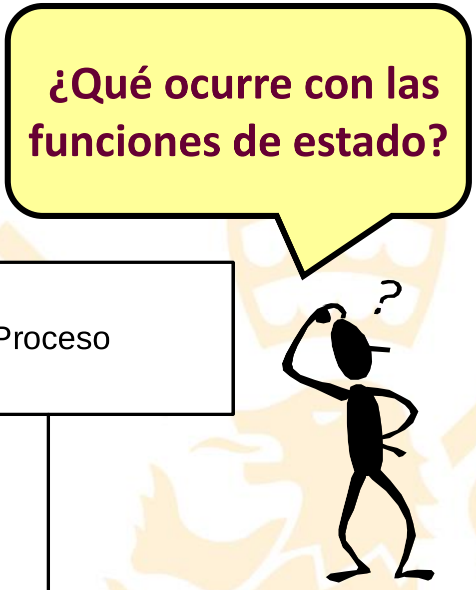
Ejercicio para escribir la función con criterios de seguridad

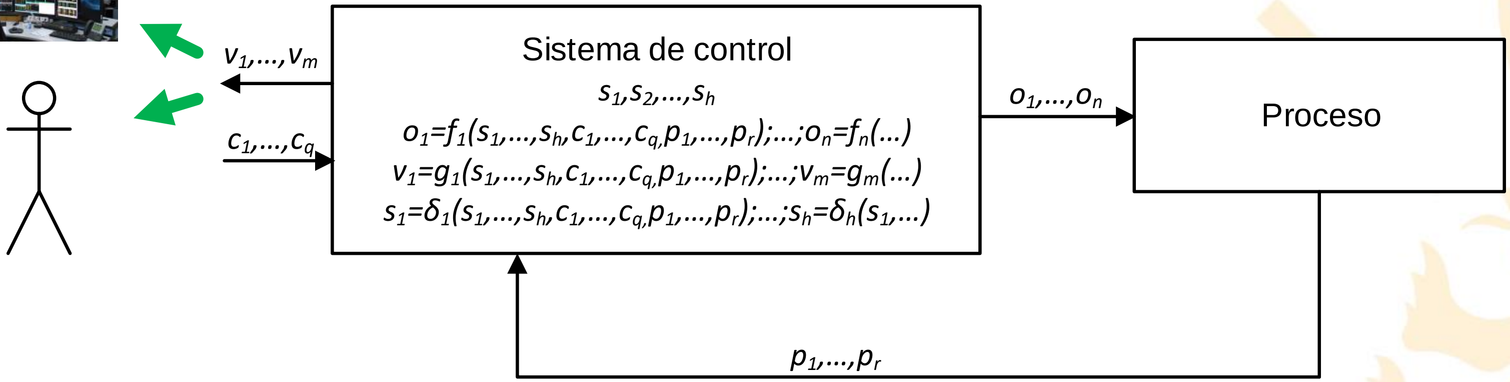
- **Se quiere automatizar el sistema de llenado del depósito de la figura a partir de dos pozos mediante las bombas B1 y B2. El depósito tiene dos boyas, na y nb, para indicar nivel alto y nivel bajo. Cada pozo tiene un sensor para detectar si hay agua (n1 y n2).**
- **El sistema funciona según las siguientes reglas:**
  - Cada bomba sólo puede bombear si hay agua en su pozo con el fin de protegerla.
  - No pueden bombear las bombas si se ha alcanzado el nivel máximo del depósito con el fin de evitar inundaciones.
  - Si el nivel del depósito está entre la boya na y la nb, funciona la bomba B1, si hay agua suficiente en el pozo 1. Si no hay agua en el pozo 1 pero la hay en el 2, funciona la bomba B2.
  - Si el nivel del depósito está por debajo de la boya nb, se activa la bomba B2, además de la B1, si se puede.
  - Determinar las funciones lógicas de B1 y B2:
    - Mediante Karnaugh
    - Mediante criterios de seguridad
    - Comparar los resultados



Funciones de observación

- **Se definen igual que las funciones de control**
- **Cuidar que las variables calculadas representen información útil para el operador**
- **También se tratan igual cuando se envía información a sistemas superiores (ejemplo: SCADA)**





Simplificación del vocabulario

- **Variables de entrada o simplemente entradas**
  - No se hace distinción entre variables tipo consigna y variables tipo proceso
  - Todas son variables lógicas
- **Variables de salida o simplemente salidas**
  - No se hace distinción entre variables tipo orden y variables tipo información
- **Funciones lógicas o simplemente funciones**
  - No se hace distinción entre funciones de control y funciones de observación
    - "Funciones que calculan las salidas a partir de las entradas"
- **Cuidado al simplificar el vocabulario**
  - Perder de vista el origen físico de la variable puede tener consecuencias graves
    - Ejemplo: una orden a un motor no es igual a controlar un piloto
- **No confundir entradas con salidas**

COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

Tipos de sistemas de control: según los conjuntos de valores

- **Conjunto de 2 valores**
  - Variables lógicas o binarias
  - Visto en el capítulo
- **Conjunto finito de valores: n valores**
  - Se codifican los valores utilizando varias variables binarias
  - Métodos de codificación típicos
    - Codificación mínima
      - Número de variables: $[\log_2 n]$
      - Representación de los valores
        - Código binario obtenido a partir del ordinal del valor dentro de su conjunto
        - Código Grey
    - Codificación máxima (n)
    - Codificaciones especiales: detectar errores, recuperar original
  - Se aplica lo visto en el capítulo

Sistemas de control de tipo lógico

- **Conjunto de infinitos valores**
  - Variables de tipo real
  - Fuera del alcance de la asignatura: Regulación Automática, Control
  - Si sólo interesa si está dentro o fuera de un rango: tratar como binario

Ejemplo de codificación: edificio de 8 plantas

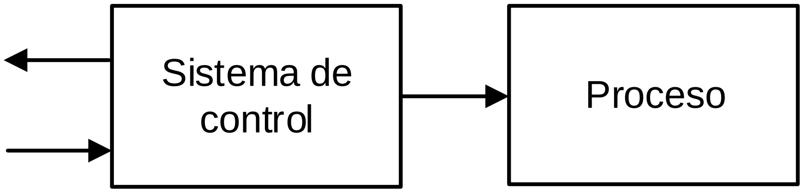
COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

Planta	Codificación mínima binaria (3 variables)	Codificación mínima Grey (3 variables)	Codificación máxima (8 variables)
1	000	000	00000001
2	001	001	00000010
3	010	011	00000100
4	011	010	00001000
5	100	110	00010000
6	101	111	00100000
7	110	101	01000000
8	111	100	10000000

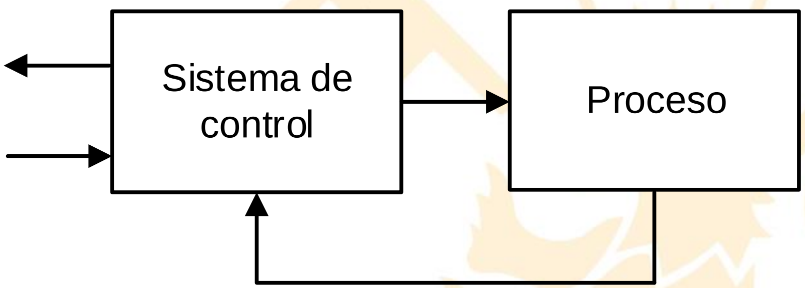
- **Codificación planta 7**
  - Binaria (variables a, b, c): a=1, b=1, c=0
  - Grey (variables a, b, c): a=1, b=0, c=1
  - Máxima (variables a, b, c, d, e, f, g, h): a=0, b=1, c=0, d=0, e=0, g=0, f=0, h=0

Tipos de sistema de control: según relación con el proceso

- **Lazo abierto:**
  - $f_i = fi(c_1, \ldots, c_q)$
  - $f_i = f_i(s_1, \ldots, s_h, c_1, \ldots, c_q)$
  - Las salidas sólo son función de las consignas y del estado, si se utiliza
    - No hay realimentación
    - El algoritmo es ciego a lo que ocurre en el proceso

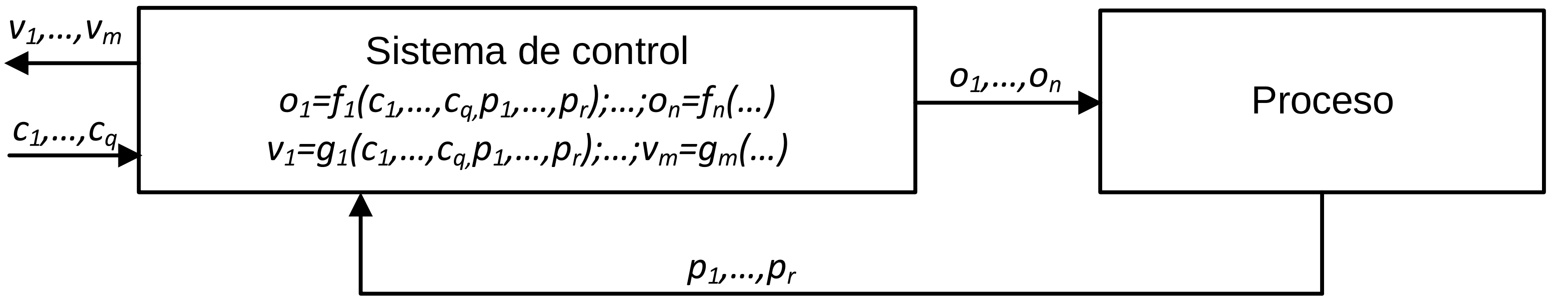


- **Lazo cerrado:**
  - $f_i = fi(c_1, \ldots, c_q, p_1, \ldots, p_r)$
  - $f_i = f_i(s_1, \ldots, s_h, c_1, \ldots, c_q, p_1, \ldots, p_r)$
  - Las salidas son función de las consignas, de las informaciones que vienen del proceso y del estado, si se utiliza

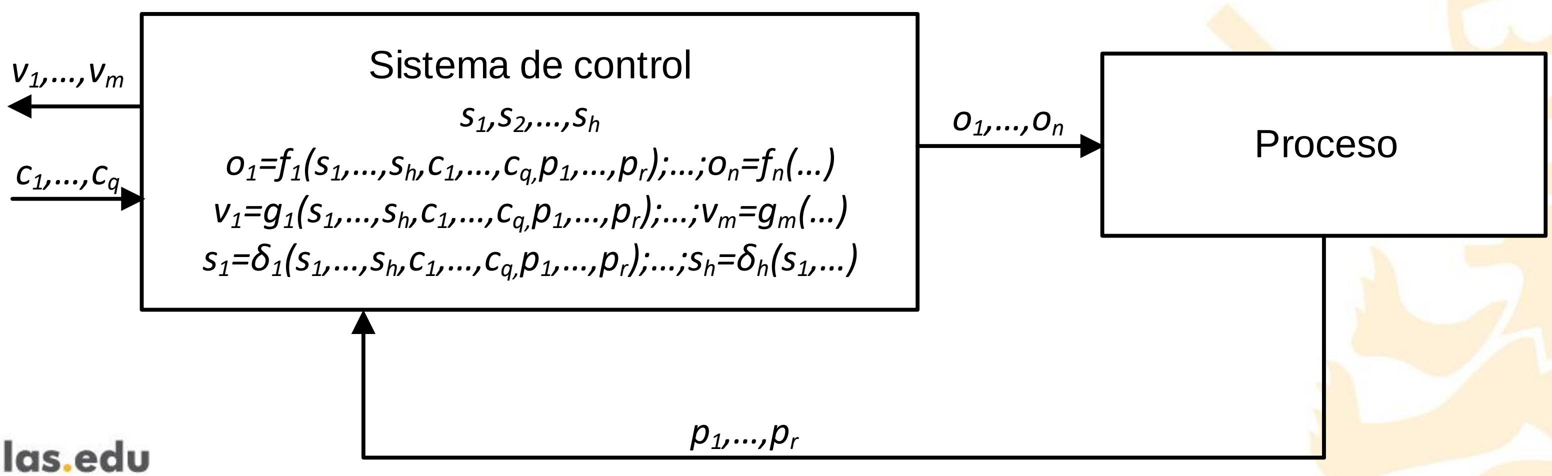


Tipos de sistemas de control: según uso del estado (I)

- **Sistemas de control combinacional: no usan variables de estado**



- **Sistemas de control secuencial: usan variables de estado**
  - Se basan en los combinacionales pero son mucho más complejos



Diferencias entre combinacional y secuencial (I)

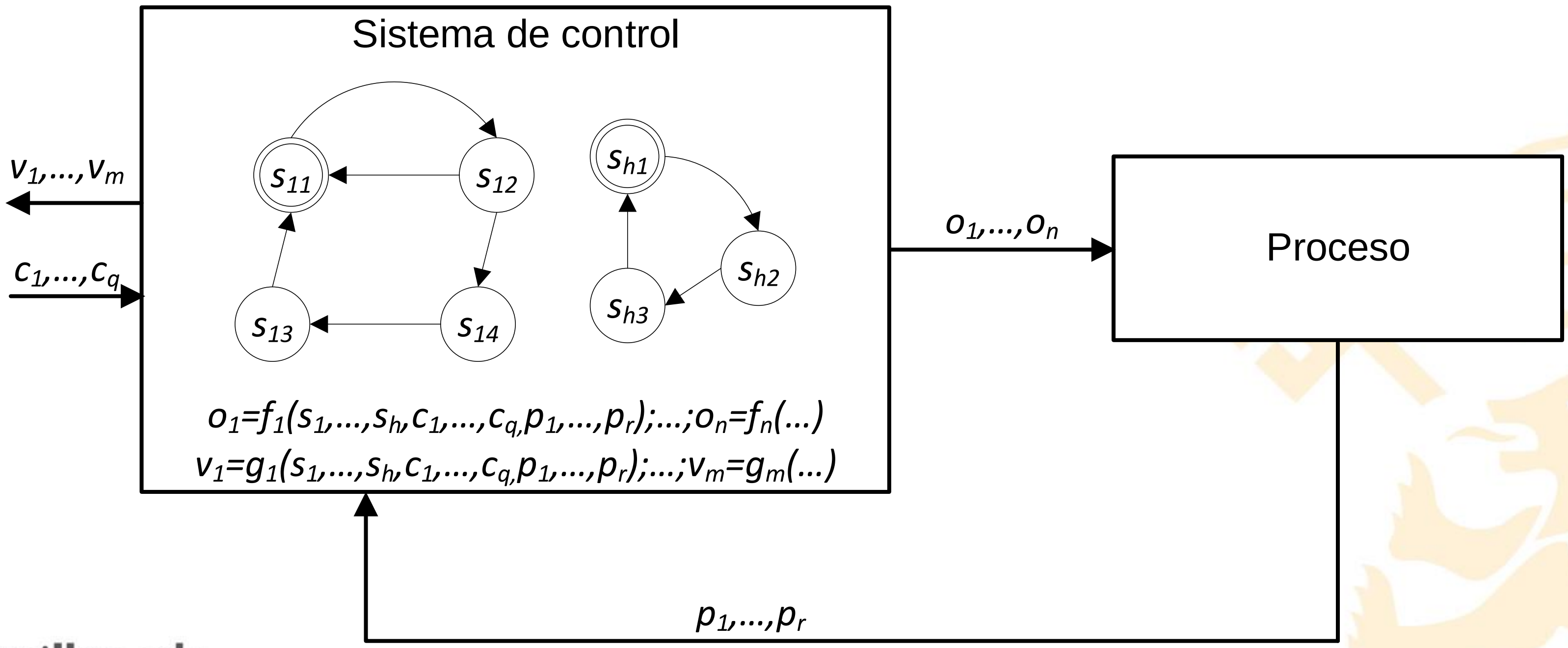
- **Sistema de control combinacional**
  - El valor de las salidas en un instante es función únicamente del valor de las entradas en ese instante
    - Las salidas son "combinación" de las entradas
    - Si un cambio en las entradas provoca un cambio en las salidas: al volver al valor original también las salidas vuelven al valor original
    - Ejemplo: pulsador que activa motor
      - Pulsado: motor en marcha
      - Sin pulsar: motor parado
  - Todos los ejemplos estudiados hasta ahora son de tipo combinacional
  - Determinar las funciones de control y de observación es igual para combinacional que para secuencial

Diferencias entre combinacional y secuencial (II)

- **Sistema de control secuencial**
  - El valor de las salidas en un instante es función del valor de las entradas en ese instante y de la "secuencia" de valores que hayan tenido hasta ese instante
    - Las salidas son combinación de las entradas y de variables de estado que almacenan cómo ha sido la evolución de las entradas
    - Si un cambio en las entradas provoca un cambio en las salidas: al volver al valor original, las salidas pueden no volver a su valor original.
    - Ejemplo: pulsador Marcha para poner en marcha un motor y otro pulsador Parado para pararlo
      - Paso 1: no se pulsa Marcha → Motor parado
      - Paso 2: se pulsa Marcha → Motor arrancado
      - Paso 3: se deja de pulsar Marcha → Motor sigue arrancado
  - Combinacional podría ser tratado como secuencial: desaconsejable

Sistemas de control secuencial

- **Dos elementos**
  - Un conjunto de máquinas de estados
    - Guardan de manera parcial o total la secuencia seguida por los valores de las entradas
  - Funciones que calculan las salidas a partir de las entradas y el estado de cada máquina de estados



MASTER EN INGENIERÍA INDUSTRIAL

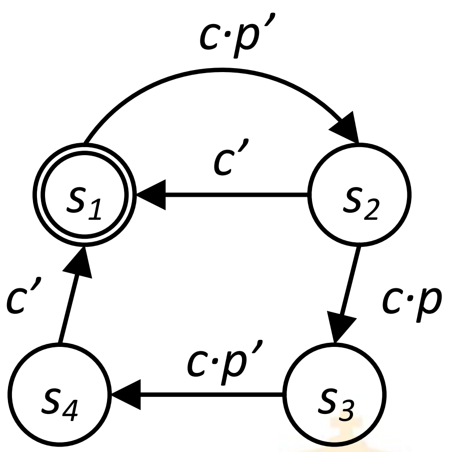
Definición de máquina de estados

- **Modela el comportamiento de un sistema como:**
  - Un conjunto de variables de entrada
  - Un conjunto S de estados en los que puede estar el sistema
    - $S = \{s_1, s_2, \ldots\}$, s_i es un estado
    - s es una variable de S que puede valer cualquier estado de S
    - El primer estado que toma el sistema se denomina estado inicial
  - Un conjunto de transiciones
    - Cada transición indica a qué estado se puede ir desde un determinado estado (estado origen) si se cumple la condición asociada
      - La condición es una expresión lógica que indica que las variables de entrada han tomado una determinada combinación de valores
    - La función que calcula las transiciones se denomina función de transición (δ)
  - Un conjunto de variables de salida
    - Para cada estado, las variables de salida toman un valor
      - Modelo de Moore
    - El valor puede ser además función de las variables de entrada
      - Modelo de Mealy

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

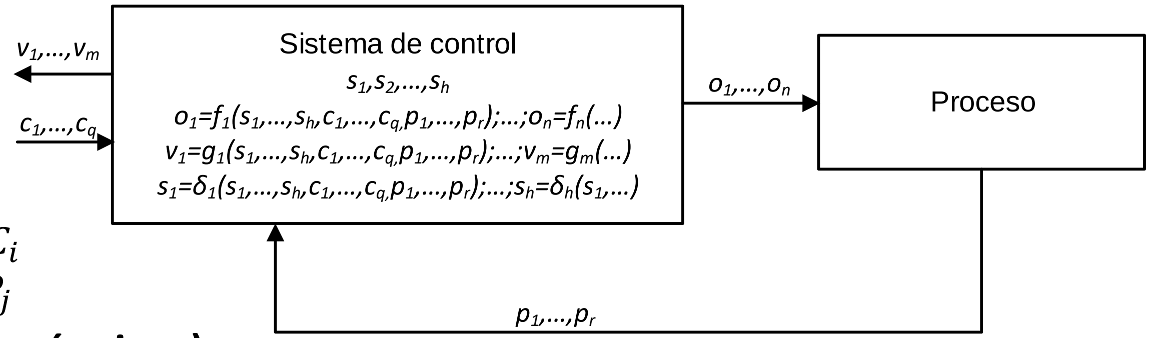
Diagrama de estados

- **Diagrama de estados: representación gráfica de la máquina de estados**
  - Círculos conectados mediante flechas
    - Circulo indica estado
      - Doble círculo indica estado inicial
    - Etiqueta dentro del círculo identifica cada estado
    - Flecha representa la transición
    - Rotulo sobre flecha es la expresión lógica de la condición asociada a la transición
      - Ejemplo: para ir del estado s_1 a s_2 la variable c debe tener valor 1 y p valor 0.
- **Ejecución de la máquina de estados**
  - La máquina comienza por el estado inicial
  - Si se cumple la condición de la transición para ir a un estado, se evoluciona a ese estado
  - Si no se cumple la condición, se mantiene en dicho estado
  - Sólo un estado puede estar activo a la vez



comillas.edu

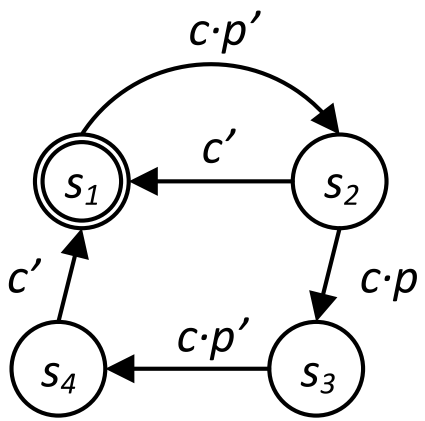
Formulación matemática del sistema de control



- **Variables de entrada**
  - Variables tipo consigna: $c_i \in C_i$
  - Variables tipo proceso: $p_j \in P_j$
- **Varias máquinas de estado (h máquinas)**
  - Un conjunto de estados S_i por cada máquina de estados
    - $S_i = \{s_{i1}, s_{i2}, \dots\}$: tienen más de 2 elementos
    - Se define la variable $s_i \in S_i$ (variable de estado): no es binaria
  - Una función de transición δ_i por cada máquina de estados
    - s_i: $s_i = \delta_i(s_1, \dots, s_i, \dots, s_h, c_1, \dots, c_q, p_1, \dots, p_r)$
    - $\delta_i: S_1 \times \dots \times S_i \times \dots \times C_1 \times C_2 \times \dots C_q \times P_1 \times P_2 \times \dots P_r \rightarrow S_i$
- **Variables de salida**
  - Variables tipo orden
  - Variables tipo información al operador
  - Las funciones de cálculo de estas variables son ahora (Mealy):
    - $o_i = f_i(s_1, \dots, s_h, c_1, \dots, c_q, p_1, \dots, p_r)$
    - $v_i = g_i(s_1, \dots, s_h, c_1, \dots, c_q, p_1, \dots, p_r)$

¿Cómo resolver que la variable de estado no es binaria?

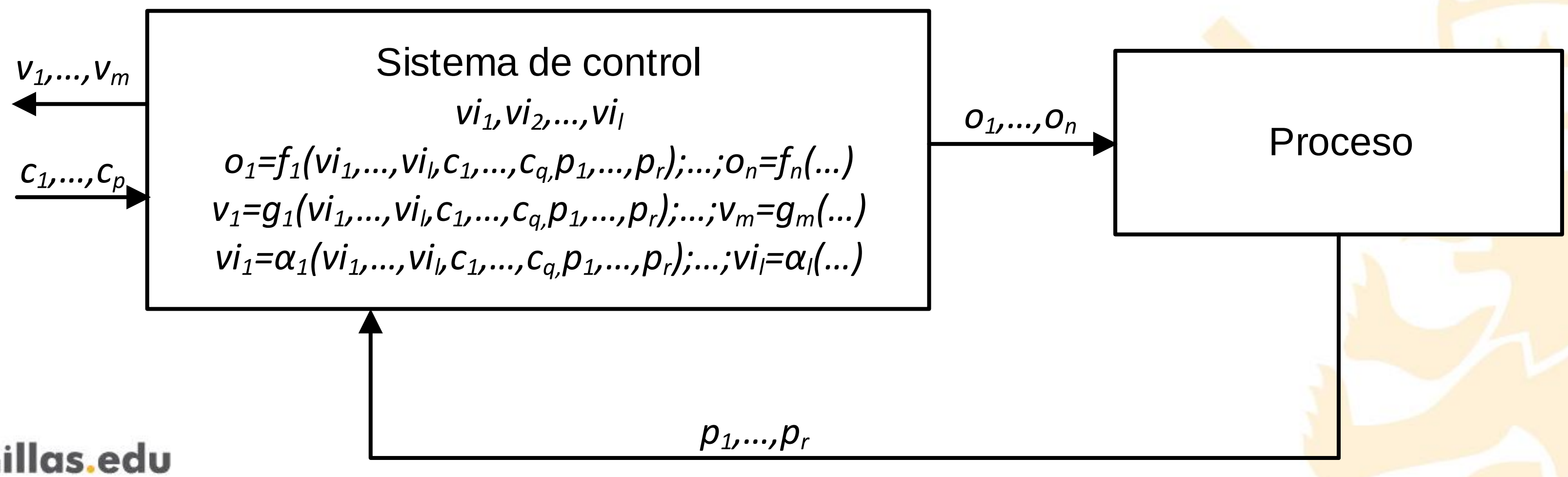
- **Caso típico de codificación**
  - Codificación máxima: una variable por estado
  - Codificación mínima: $\lceil \log_2 n \rceil$
    - n es el cardinal del conjunto de estados
    - Binaria: utilizar el ordinal (menos 1) del estado dentro del conjunto de estados
    - Grey: asignar la codificación de tal manera que en cada transición sólo cambie el valor de una variable
      - No siempre se puede aplicar directamente
- **Ejemplo**



	Máxima: 4 variables				Binaria		Grey	
Estado	vi_1	vi_2	vi_3	vi_4	vi_1	vi_2	vi_1	vi_2
s_1	1	0	0	0	0	0	0	0
s_2	0	1	0	0	0	1	0	1
s_3	0	0	1	0	1	0	1	1
s_4	0	0	0	1	1	1	1	0

Sistema de control secuencial con memoria

- **Para guardar el estado de las máquinas de estado se utilizan l variables binarias internas al sistema de control: vi_i**
  - Equivalentes a memorias
  - Estarán divididas en h grupos
    - Resultado de codificar las h máquinas de estados
- La función de transición es ahora α_i que trabaja a partir de las memorias que guardan el estado de las máquinas de estados

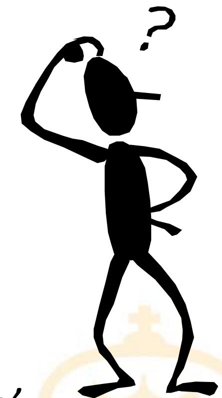


comillas.edu

Ejemplo control secuencial: marcha/paro de motor

COMILLAS UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

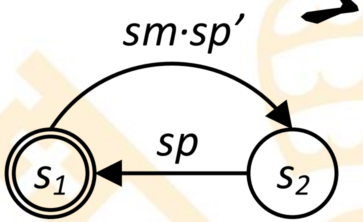
¿Cómo se implementa $sp'(m + sm)$?



- **Requisitos de la automatización: en un panel de control hay dos pulsadores para poner en marcha y parar un motor**
- **Variables (sin incluir estado):**

Fuente, receptor, …	Tipo variable	Variable	0	1
Pulsador de marcha	Consigna	*sm*	*"No orden de marcha"*	*"Orden de marcha"*
Pulsador de paro	Consigna	*sp*	*"No orden de paro"*	*"Orden de paro"*
Motor	Orden	*o*	*"No Girar"*	*"Girar"*

- **Tipo de sistema de control: secuencial**
  - Con los pulsadores sin pulsar el motor puede estar girando y no girando
- **Estados posibles**
  - $S = \{s_1, s_2\}$ (s_1:motor no girando; s_2:motor girando)
  - Codificación: variable m (0:s_1; 1:s_2)
- **Diagrama de estados**
  - Si en la transición sólo se utiliza sm el sistema no tiene un estado estable al pulsar sm y sp a la vez
- **Función de transición: tabla**
  - $m = m \cdot sp' + sm \cdot sp' = sp'(m + sm)$
- **Función de cálculo de la variable de salida:** $o = m$

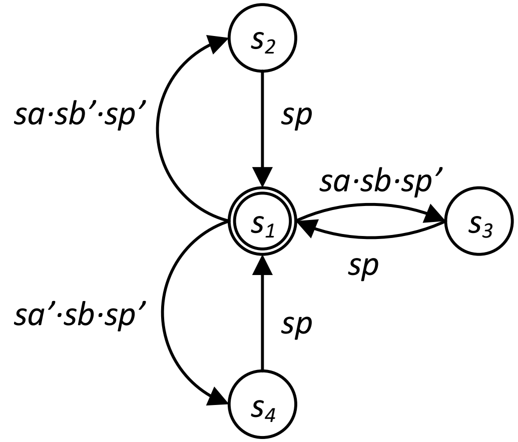


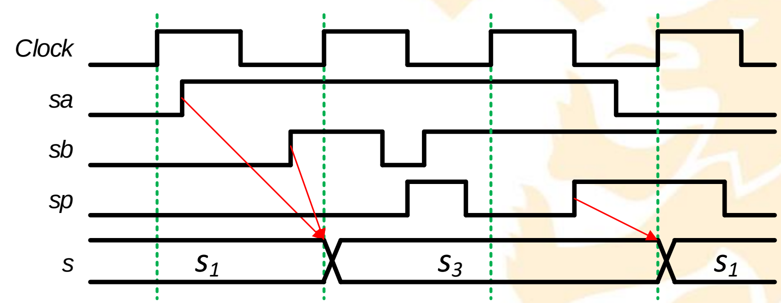
m \ sm, sp	00	01	11	10
0	0	0	0	1
1	1	0	0	1

comillas.edu

Dinámica las máquinas de estados: asíncronas y síncronas

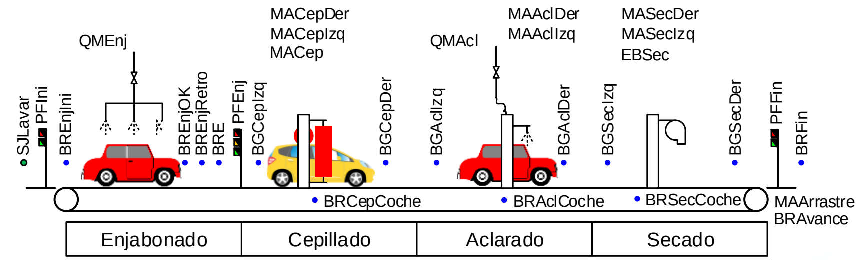
- **Asíncrona**
  - La máquina evoluciona en el momento que ocurre el cambio en la entrada
  - Desventaja: puede evolucionar a dónde no interesa cuando hay más de un cambio "simultáneo" de varias entradas
    - sa y sb: entre 00 y 11 puede aparecer 01 o 10
  - Ventaja: velocidad
- **Síncrona**
  - La máquina evoluciona cuando lo indica la señal de reloj
  - Desventaja: siempre hay un retardo y se pueden perder eventos
  - Ventaja: siempre se evoluciona de forma controlada
    - La implementación soluciona el problema del cambio justo en el flanco de la señal de reloj

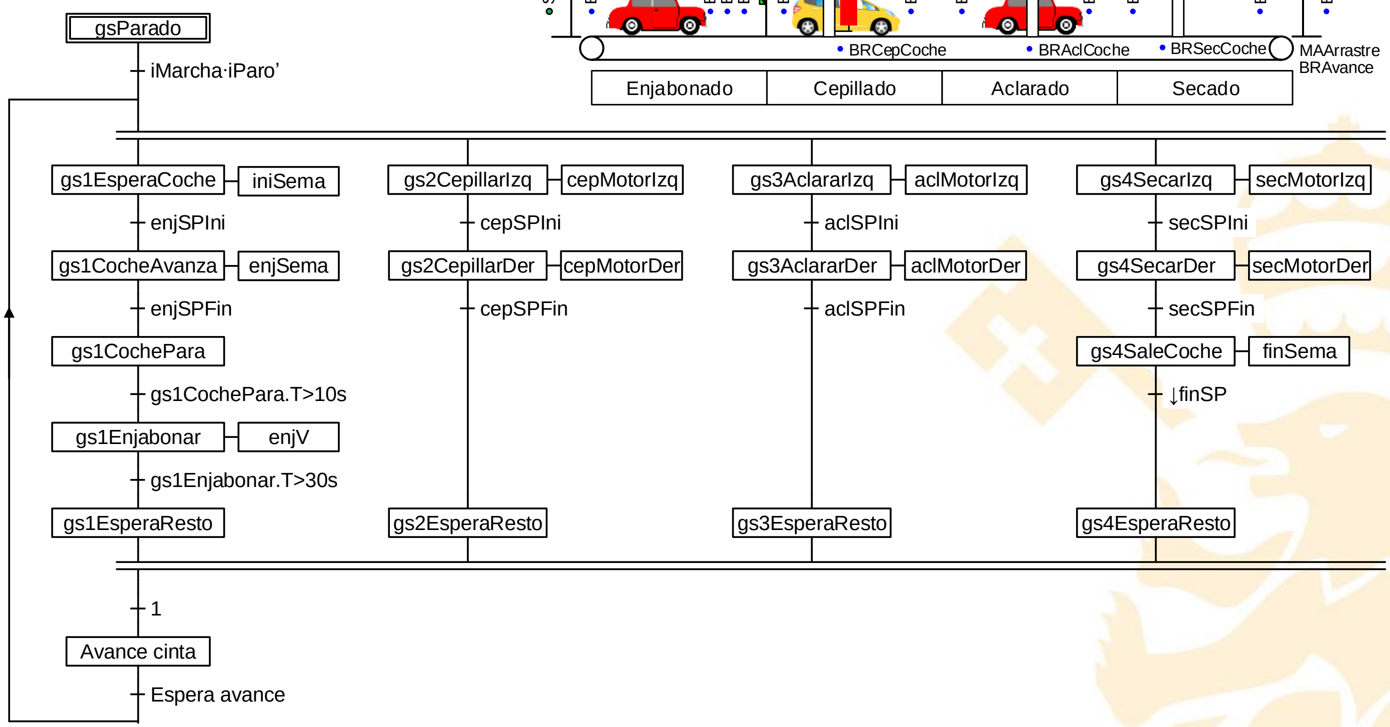




Diseño e implementación de sistemas de control secuencial

- **Casos sencillos:**
  - Soluciones asíncronas basadas en circuitos de marcha/paro (automatismos cableados)
  - Soluciones síncronas basadas en automatismos programados
- **Casos complejos:**
  - Metodología grafcet
    - Ejemplo: túnel de lavado





comillas.edu

Resumen

- **Modelo matemático de un sistema de control**
  - Variables ligadas por funciones
- **Tipos de variables y funciones en un sistema de control**
- **Cálculo de la función mediante tabla**
- **Cálculo de la función mediante expresión lógica**
  - Definición de conjuntos y variables binarias
  - Definición de operadores lógicos
  - "Álgebra de Boole" asociada
- **Conversión entra tabla y expresión lógica**
- **Implementación básica de la función mediante algebra de contactos**
- **Repaso de simplificación de funciones lógicas**
  - Uso del don't care
- **Uso de criterio de seguridad para escribir la función lógica**
- **Tipos de sistemas de control: lazo abierto/lazo cerrado, combinacional/secuencial**
- **Cálculo de la máquina de estados**

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

Preguntas

- **¿Cómo ve las matemáticas un sistema automatizado?**
- **Tipos de variables y de funciones en un sistema automatizado**
- **Identificar las variables incluyendo tipo y conjunto de valores posibles para una determinada especificación**
- **Escribir la expresión lógica de una función lógica a partir de los requisitos**
- **Obtener una expresión lógica simplificada utilizando diferentes métodos**
- **Escribir la expresión lógica de una función lógica a partir de los requisitos utilizando criterios de seguridad**
- **¿Cómo se solucionan los casos de variables con más de 2 valores en un sistema de control de tipo lógico?**
- **Diferencia entre sistemas de control en lazo abierto y en lazo cerrado**
- **Diferencia entre sistemas de control combinacionales y secuenciales**
- **Identificar los problemas entre un sistema de control con máquinas de estado asíncrona y otro síncrona**