Álgebra Booleana y Circuitos Lógicos

UCR – ECCI

CI-0111 Estructuras Discretas

Prof. Kryscia Daviana Ramírez Benavides

Álgebra Booleana

- Tanto los conjuntos como las proposiciones tienen propiedades similares. Estas propiedades se usan para definir una estructura matemática llamada **álbebra de Boole** o **álgebra booleana**, en honor de George Boole (1813-1864).
- Esta álgebra se utiliza en dos casos concretos:
  - Compuertas lógicas.
  - Circuitos de interruptores.

Álgebra Booleana (cont.)

- Sea B un conjunto en el cual se han definido dos operaciones binarias, + y *, y una operación unitaria, denotada ’; sean 0 y 1 dos elementos diferentes de B. Entonces a la sextupla

$$\langle B,+,*,',0,1\rangle$$

  se le llama **álgebra de Boole** si se cumplen los axiomas de la tabla para elementos a, b y c cualesquiera en el conjunto B:
  - Leyes conmutativas.
  - Leyes distributivas.
  - Leyes de identidad.
  - Leyes de complemento.

Álgebra Booleana (cont.)

- Aspectos importantes del álgebra:
  - Al elemento 0 se le llama el **elemento cero**.
  - Al elemento 1 se le llama **elemento unidad**.
  - A la operación unitaria *a*’ se le llama **complemento** de *a*.
  - A los resultados de las operaciones binarias + y * se les llama, respectivamente, **suma** y **producto**.
- Aparte de los axiomas, en la tabla se muestran otras propiedades que tiene el álgebra de Boole, que se pueden obtener mediante los axiomas.

Axiomas del Álgebra de Boole	
Leyes Conmutativas	
$a + b = b + a$	$a * b = b * a$
Leyes Distributivas	
$a + (b * c) = (a + b) * (a + c)$	$a * (b + c) = (a * b) + (a * c)$
Leyes de Identidad	
$a + 0 = a$	$a * 1 = a$
Leyes de Complemento	
$a + a' = 1$	$a * a' = 0$
Leyes de Idempotencia	
$a + a = a$	$a * a = a$
Leyes de Acotamiento	
$a + 1 = 1$	$a * 0 = 0$
Leyes de Absorción	
$a + (a * b) = a$	$a * (a + b) = a$
Leyes Asociativas	
$(a + b) + c = a + (b + c)$	$(a * b) * c = a * (b * c)$
Unicidad del Complemento	
Si $a + x = 1$ y $a * x = 0$, entonces $x = a'$	
Ley de Involución	
$(a')' = a$	
Teoremas	
$0' = 1$	$1' = 0$
Leyes de DeMorgan	
$(a + b)' = a' * b'$	$(a * b)' = a' + b'$

Álgebra Booleana (cont.)

- Ejemplos:
  - Sea B el conjunto de dos elementos, $\{0,1\}$, con operaciones + y * definidas:

+	1	0
1	1	1
0	1	0

*	1	0
1	1	0
0	0	0

    Los complementos se defines por 1’ = 0 y 0’ = 1.
  - El ejemplo anterior se puede extender para sucesiones de n bits, sea B_n.

Álgebra Booleana (cont.)

- Ejemplos:
  - Sea ζ una colección de conjuntos cerrados bajo uniones, intersecciones y complementos. Se tiene como elemento cero al conjunto vacío ∅ y como elemento unidad al conjunto universal *U*.

$$\langle \zeta, \cup, \cap, \neg, \varnothing, U \rangle$$

  - Sea Π el conjunto de proposiciones, que tiene como operaciones ∨ y ∧, con la negación ~ como complemento. Se tiene como elemento cero una contradicción *f* y como elemento unidad una tautología *t*.

$$\langle \Pi, \vee, \wedge, \approx, f, t \rangle$$

Álgebra Booleana (cont.)

- Ejemplos:
  - Sea $D_{70} = \{1,2,5,7,10,14,35,70\}$, los divisores de 70. Se tienen las operaciones de mínimo común múltiplo de a y b como la suma, máximo común divisor de a y b como el producto, y 70 dividido entre a el complemento de a. Se tiene como elemento cero al 1 y como elemento unidad al 70.

$$\langle D_{70}, MCM(a,b), MCD(a,b), 70/a, 1, 70\rangle$$

Dualidad

- El **dual** de cualquier enunciado en un álgebra de Boole B es el enunciado obtenido al intercambiar las operaciones + y *, e intercambiar los correspondientes elementos identidad 0 y 1, en el enunciado original.
  - Ejemplo: $(1 + a) * (b + 0) = b \Rightarrow$ el dual es: $(0 * a) + (b * 1) = b$
- **Principio de Dualidad:** El dual de cualquier teorema en un álgebra de Boole es también un teorema.
  - En otras palabras, si cualquier enunciado es una consecuencia de los axiomas de un álgebra de Boole, entonces el dual también es una consecuencia de estos axiomas; ya que el enunciado dual se puede probar usando el dual de cada paso en la demostración del enunciado original.

Orden y Álgebra de Boole

- Una relación $\preceq$ es un conjunto S se llama un **orden parcial** en S si cumple las tres propiedades siguientes:
  - $a \preceq a, \forall a \in S.$
  - Si $a \preceq b$ y $b \preceq a$, entonces $a = b$.
  - Si $a \preceq b$ y $b \preceq c$, entonces $a \preceq c$.
- Un conjunto S junto con un orden parcial se llama **conjunto parcialmente ordenado**. En tal caso se puede escribir y leer:
  - $a \preceq b \Rightarrow a$ precede a b.
  - $a \prec b \Rightarrow a$ precede estrictamente a b, si $a \preceq b$ pero $a \neq b$.
  - $a \succeq b \Rightarrow a$ sigue a b, si $b \preceq a$.
  - $a \succ b \Rightarrow a$ sigue estrictamente a b, si $b \prec a$.

Orden y Álgebra de Boole (cont.)

- El término parcial se usa al definir un conjunto parcialmente ordenado S, porque puede haber elementos a y b de S que no son comparables, o sea, tales que ni $a \precsim b$ ni $b \precsim a$.
- Si por otra parte, todo par de elementos de S es comparable, entonces se dice que S es **totalmente ordenado**, o **linealmente ordenado**, y S se denomina **cadena**.

Orden y Álgebra de Boole (cont.)

- Ejemplos:
  - Sea ζ una clase cualquiera de conjuntos, la relación de inclusión $\subset$ es un orden parcial de ζ.
  - En los números enteros positivos, se dice que "a divide a b", escrito $a \mid b$, si existe un entero c tal que $ac = b$; esta relación de divisibilidad es un orden parcial en $\boldsymbol{N}$. Notar que, por ejemplo, 3 y 5 no son comparables ya que ninguno divide al otro.
  - La relación $\leq$ también es un orden parcial de los enteros positivos $\boldsymbol{N}$. Notar que $\boldsymbol{N}$ es totalmente ordenado por medio de esta relación.

Orden y Álgebra de Boole (cont.)

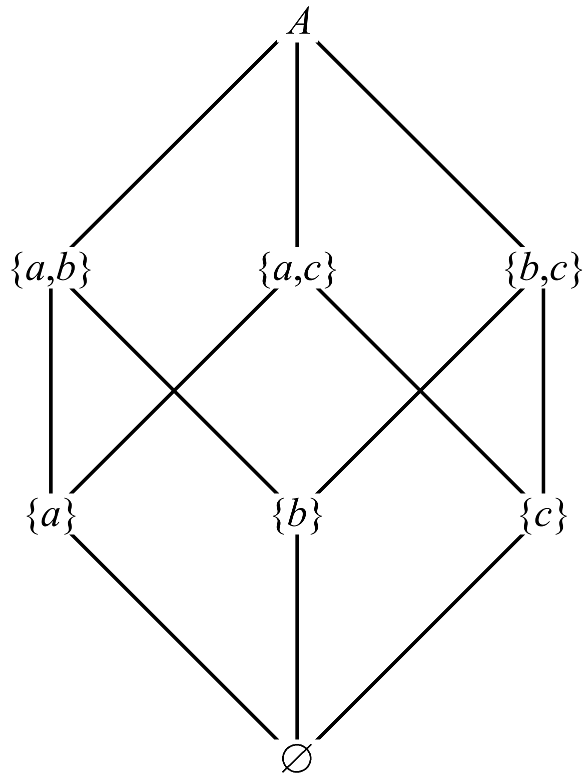
- Sea B un álgebra de Boole; B es entonces parcialmente ordenado, siendo $a \precsim b$ si y sólo si $a + b = b$.
- Sea B cualquier álgebra de Boole; entonces para cualquier elemento a de B, $0 \precsim a \precsim 1$, ya que $0 + a = a$ y $a + 1 = 1$.
  - Ejemplos:
    - El álgebra de Boole de conjuntos, el conjunto A precede al conjunto B si A es subconjunto de B.
    - El álgebra de Boole del cálculo proposicional, la proposición P precede a la proposición Q si P implica lógicamente a Q.

Orden y Álgebra de Boole (cont.)

- Un conjunto finito parcialmente ordenado *S* y, en particular, un álgebra de Boole finita *S*, se puede representar por un diagrama de la siguiente manera.
  - Un elemento *B* de *S* se dice que es un **sucesor inmediato** de un elemento *a*, escrito $a \prec\prec b$; si $a \prec b$, pero no hay ningún elemento *x* de S tal que $a \prec x \prec b$.
  - Los elementos se representan por puntos y habrá una flecha, o una línea dirigida hacia arriba, de un elemento *a* a un elemento *b* cada vez que $a \prec\prec b$.
  - En caso de que *S* sea un álgebra de Boole, el elemento cero estará en la parte más baja del diagrama y el elemento unidad en la parte más alta.

Orden y Álgebra de Boole (cont.)

- **Ejemplo:** Sea $A = \{a,b,c\}$, y sea $\zeta(A)$ la colección de todos los subconjuntos de A: $\zeta(A) = [A, \{a,b\}, \{a,c\}, \{b,c\}, \{a\}, \{b\}, \{c\}, \varnothing]$. $\zeta(A)$ es un álgebra de Boole de conjuntos cuyo diagrama se muestra a la derecha, observar que $\varnothing$ está abajo en el diagrama y A está arriba.



Orden y Álgebra de Boole (cont.)

- Sea B una álgebra de Boole, entonces:
  - Un elemento a de B se llama **átomo** de B si es un sucesor inmediato del elemento cero. En el diagrama anterior, los átomos son: $\{a\}$, $\{b\}$ y $\{c\}$.
  - Un elemento M de B se llama **maxitérmino** (***maxterm***) de B si el elemento unidad es su único sucesor estricto. En el diagrama anterior, los maxitérminos son: $\{a,b\}$, $\{a,c\}$ y $\{b,c\}$.
- Sea B una álgebra de Boole finita con n átomos; entonces B tiene 2^n elementos, y todo elemento no nulo de B es la suma de un conjunto único de átomos.

Expresiones de Boole

- Una **expresión booleana *E*** en un conjunto de variables (x_1, x_2, …, x_n), algunas veces escrito $E(x_1, x_2, \ldots, x_n)$, es una variable o una expresión construida con estas variables que usan las operaciones booleanas +, * y ’.
  - Ejemplos:
    - $E(x,y,z) = (x + y'z)' + (xyz' + x'y)'$
    - $E(x,y,z) = ((xy'z' + y)' + x'z)'$

Expresiones de Boole (cont.)

- Cuando se trabaja con expresiones booleanas, es deseable que estas se encuentren expresadas en una de dos formas:
  - Suma de productos o minitérminos o forma normal disyuntiva (FND).
  - Producto de sumas o maxitérminos o forma normal conjuntiva (FNC).
- Esas dos formas básicas de expresiones son canónicas y permiten asociar a una función una expresión algebraica única (la tabla de verdad también es única).
  - Se puede pasar una expresión booleana a suma de productos o producto de sumas utilizando las leyes distributivas.

Expresiones de Boole (cont.)

- **Suma de productos.** Consiste de dos o más grupos de literales, cada literal es recibida como entrada por un AND y la salida de cada una de estas compuertas (AND) es recibida como entrada por una compuerta OR.
  - Cuando dos o más productos se suman mediante la suma booleana.
- **Producto de sumas.** Un producto de sumas consiste de dos o más grupos de literales, cada literal es recibida como entrada por un OR y la salida de cada una de estas compuertas (OR) es recibida como entrada por una compuerta AND.
  - Cuando dos o más términos de suma se multiplican mediante la multiplicación booleana.

Expresiones de Boole (cont.)

- Si tal tabla de verdad ha sido dada, entonces la expresión booleana en:
  - FND puede escribirse por inspección, a cada conjunto de condiciones para los cuales la expresión booleana sea 1, corresponderá un término en la FND.
    - La suma de estos términos da la función aunque no necesariamente en la forma más simple.
  - FNC puede escribirse por inspección, a cada conjunto de condiciones para los cuales la expresión booleana sea 0, corresponderá un término en la FNC.
    - La multiplicación de estos términos da la función aunque no necesariamente en la forma más simple.

Expresiones de Boole (cont.)

- Dada la tabla de verdad:

A	00001111
B	00110011
C	01010101
Y	00110101

  - FND $\rightarrow$ $Y = ABC + AB'C + A'BC + A'BC'$
  - FNC $\rightarrow$ $Y = (A + B + C')(A + B' + C')(A' + B' + C)(A' + B' + C')$

Expresiones de Boole (cont.)

- Un **literal** es una variable o una variable complementada, por ejemplo: x, x', etc.
- Un **producto fundamental** es un literal o un producto de dos o más literales en los cuales no hay dos literales con una misma variable, por ejemplo: x, x', xy, $x'y$, xz', $x'yz$, etc.
- Un **producto de Boole** es producto de dos o más literales, por ejemplo: $xyx'z$, $xyzy$, etc.
  - $xyx'z = xx'yz = 0yz = 0$ ($x * x' = 0$ por la ley del complemento)
  - $xyzy = xyyz = xyz$ ($y * y = y$ por la ley de idempotencia)
- **Todo producto de Boole se puede reducir a 0 o a un producto fundamental.**

Expresiones de Boole (cont.)

- Un producto fundamental P_1 se dice que está **incluido** o **contenido** en otro producto fundamental P_2, si los literales de P_1 son también literales de P_2; por lo tanto $P_1 + P_2 = P_1$ por la ley de absorción.
  - $\boldsymbol{x'}z + \boldsymbol{x}y'z$ ($x'z$ no está incluido en $xy'z$)
  - $x'z + x'yz = x'z$ ($x'z$ está incluido en $x'yz$)
- Una expresión de Boole E se dice que está en **forma de suma de productos** o en **forma minitérmino** (***miniterm***) si E es un producto fundamental o, es la suma de dos o más productos fundamentales, ninguno de los cuales está incluido en otro.
  - $E_1 = \boldsymbol{x'z} + xy'z + \boldsymbol{x'yz}$ (E_1 no está en forma de suma de productos)
  - $E_2 = xz' + x'yz' + xy'z$ (E_2 está en forma de suma de productos)

Expresiones de Boole (cont.)

- Toda expresión de Boole no nula *E* se puede poner en forma de suma de productos con el siguiente procedimiento:
  - Usando las leyes de DeMorgan y la involución, se puede mover la operación de complemento dentro de cualquier paréntesis hasta que finalmente se aplique solamente a variables. *E* consistirá entonces solamente en sumas y productos de literales.
  - Usando la ley distributiva, se puede transformar *E* en una suma de productos.
  - Usando las leyes conmutativas, de idempotencia y de complemento, se puede transformar cada producto en *E* en 0 o en un producto fundamental.
  - Usando la ley de absorción, se puede poner *E* en forma de suma de productos.

Expresiones de Boole (cont.)

- Ejemplo:

$$E(a,b,c) = ((a*b)'*c)'*((a'+c)*(b'+c'))' = ((ab)'c)'((a'+c)(b'+c'))'$$

$$(1)\ E(a,b,c) = ((ab)''+c')((a'+c)'+(b'+c')') = (ab+c')(ac'+bc)$$

$$(2)\ E(a,b,c) = aabc'+abbc+ac'c'+bcc'$$

$$(3)\ E(a,b,c) = abc'+abc+ac'+0 = abc'+abc+ac'$$

$$(4)\ E(a,b,c) = ac'+abc$$

Expresiones de Boole (cont.)

- Una expresión de Boole no nula $E(x_1, x_2, \ldots, x_n)$ se dice que está en **forma completa de suma de productos** si E está en forma de suma de productos, y en cada producto se usan todas las variables.
- Cualquier expresión de Boole E que sea una suma de productos se puede escribir en forma completa de suma de productos.
  - Si un producto fundamental P de E no usa x_i, entonces se puede multiplicar P por $x_i + x_i'$; esto se puede hacer ya que $x_i + x_i' = 1$.
  - Así se continua hasta que todos los productos usen todas las variables.
- Además, la representación que se obtiene de E en forma completa de suma de productos es **única**.

Expresiones de Boole (cont.)

- Ejemplo:

$$E(a,b,c) = ac' + abc$$

$$E(a,b,c) = ac'(b + b') + abc$$

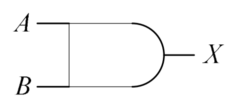
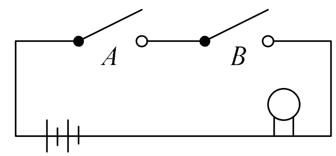
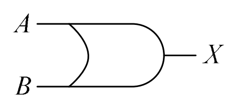
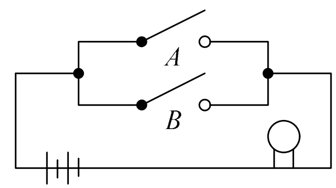
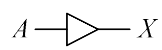
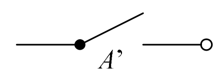
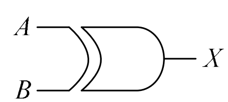
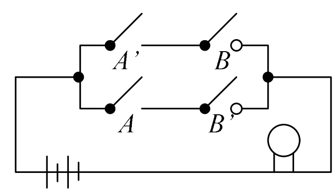
$$E(a,b,c) = abc' + ab'c' + abc$$

$$E(a,b,c) = abc + abc' + ab'c'$$

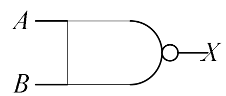
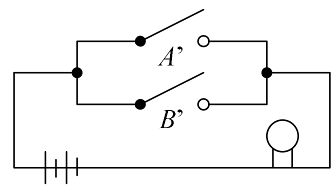
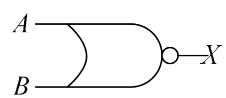
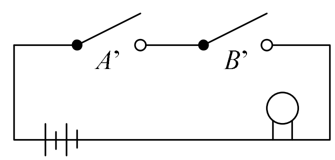
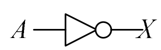
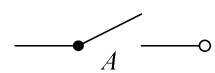
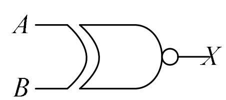
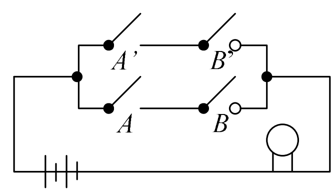
Compuertas Lógicas

- Los circuitos lógicos, que pronto se explicarán, se construyen a partir de ciertos circuitos elementales llamados **compuertas lógicas**.
- A continuación se presentan dos tablas, donde se resumen las compuertas lógicas más importantes.

Compuertas Lógicas (cont.)

Expresión	Compuerta Lógica	Tabla de Verdad	Circuito de Interruptores
$X = AB$	 **AND**	A B | X 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	
$X = A + B$	 **OR**	A B | X 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	
$X = A'$	 **NOT**	A | X 0 | 1 1 | 0	
$X = A \oplus B$ $\Rightarrow$ $X = A'B + AB'$	 **XOR** **(OR exclusivo)**	A B | X 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	

Compuertas Lógicas (cont.)

Expresión	Compuerta Lógica	Tabla de Verdad	Circuito de Interruptores
$X = (AB)'$	 **NAND**	A B | X 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	
$X = (A + B)'$	 **NOR**	A B | X 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	
$X = (A')' \Rightarrow X = A$		A | X 0 | 0 1 | 1	
$X = (A \oplus B)' = A \otimes B$ $\Rightarrow$ $X = A'B' + AB$	 **NOR** **(exclusivo)**	A B | X 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1	

Circuitos Lógicos

- Los **circuitos lógicos** se pueden visualizar como máquinas que contienen uno o más dispositivos de entrada y exactamente un dispositivo de salida.
- En cada instante cada dispositivo de entrada tiene exactamente un bit de información, un 0 o un 1; estos datos son procesados por el circuito para dar un bit de salida, un 0 o un 1, en el dispositivo de salida.
- De esta manera, a los dispositivos de entrada se les puede asignar sucesiones de bits que son procesadas por el circuito bit por bit, para producir una sucesión con el mismo número de bits.

Circuitos Lógicos (cont.)

- Un **bit** se puede interpretar como un voltaje a través de un dispositivo de entrad/salida; aun más, una sucesión de bits es una sucesión de voltajes que pueden subir o bajar (encendido o apagado).
- Se puede suponer que el circuito siempre procesa la sucesión de izquierda a derecha o de derecha a izquierda. Si no se dice otra cosa se adopta la primera convención.

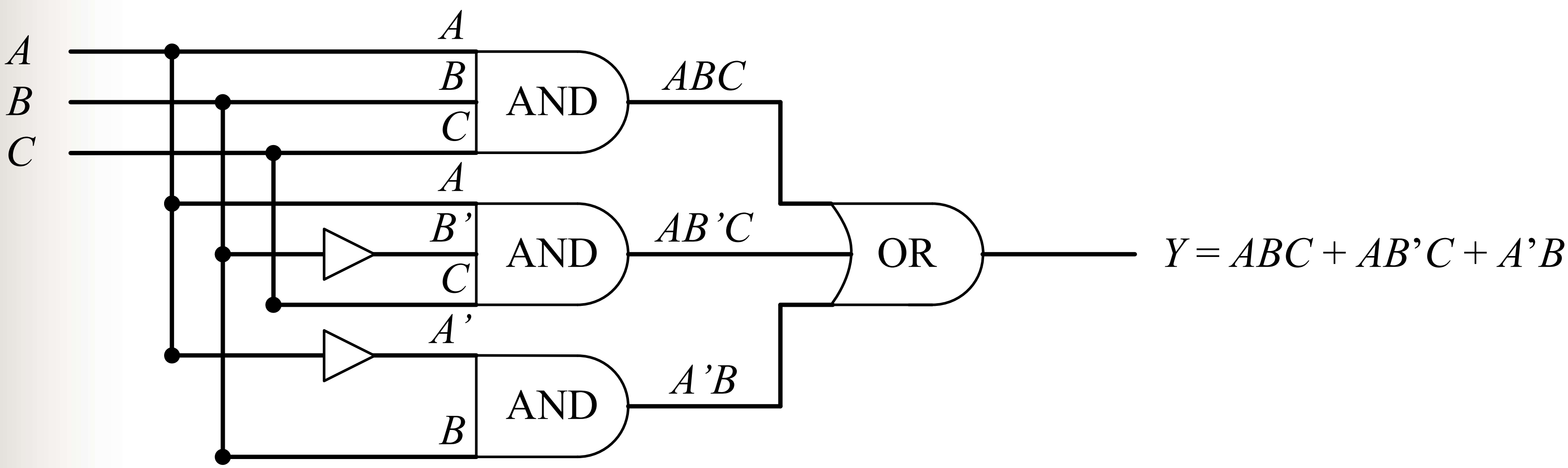
Circuitos Lógicos (cont.)

- Las tablas de verdad para las compuertas lógicas AND, OR y NOT, que se mostraron en la tablas anteriores, son respectivamente idénticas a las correspondientes proposiciones de conjunción ($p \wedge q$), disyunción ($p \vee q$) y negación ($\sim p$).
- La única diferencia entre las tablas de verdad de las compuertas y las proposiciones es que se usa el 1 y 0, en vez de V y F.
- Así que las compuertas lógicas satisfacen las mismas leyes de las proposiciones, y así forman un álgebra de Boole.

Circuitos Lógicos (cont.)

- Los circuitos lógicos vienen en varios patrones. Se tratará especialmente un patrón que corresponde a una expresión de Boole de suma de productos.
  - Un circuito AND-OR tiene varias entradas, con algunas de las entradas o sus complementos alimentando cada compuerta AND.
  - Las salidas de todas las compuertas AND alimentan una sola compuerta OR, la cual de al salida para el circuito.
  - En casos límite, puede haber una sola compuerta AND sin una compuerta OR, o ninguna compuerta AND y una sola compuerta OR.

Circuitos Lógicos (cont.)



Circuitos Lógicos (cont.)

- Dado cualquier circuito lógico L, se quiere averiguar el efecto de L en cualquier entrada arbitraria; usualmente esto se especifica por medio de una tabla de verdad.
- La tabla de verdad de L se obtiene escribiendo primero L como una expresión de Boole $L(A,B,C,\ldots)$, y calculando entonces la tabla de verdad paso por paso.
- La expresión de Boole se obtiene del circuito siguiendo las entradas a través de todas las compuertas.

Circuitos Lógicos (cont.)

- Para el circuito anterior se obtiene la siguiente tabla de verdad:

$$A = 00001111$$
$$B = 00110011$$
$$C = 01010101$$

$$ABC = 00000001$$
$$AB'C = 00000100$$
$$A'B = 00110000$$

$$Y = 00110101$$

A	00001111
B	00110011
C	01010101
Y	00110101

Circuitos Lógicos (cont.)

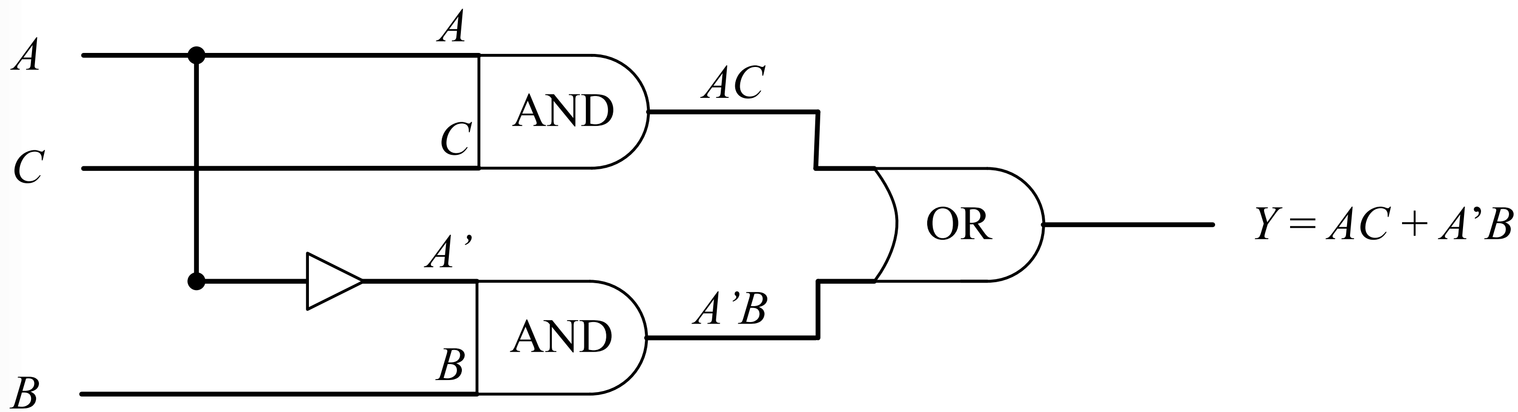
- Como los circuitos lógicos forman un álgebra de Boole, se puede usar los teoremas (axiomas y propiedades) del álgebra para simplificar los circuitos.

$$Y = ABC + AB'C + A'B = AC(B + B') + A'B$$

$$Y = AC * 1 + A'B = AC + A'B$$

- Así el circuito anterior puede ser reemplazado por el circuito lógico más sencillo que se puede formar de la expresión de Boole resultante.
- Los dos circuitos lógicos son equivalentes, es decir, tienen la misma tabla de verdad.

Circuitos Lógicos (cont.)



Circuitos Lógicos (cont.)

- La tabla de verdad (única) de una expresión de Boole equivale a la única forma completa de suma de productos que se puede obtener de una expresión de Boole.
- Esta correspondencia surge del hecho que se asigna cualquier combinación de 1s y 0s a las variables, cada uno de los productos fundamentales que involucran todas las variables de la salida toma el valor 1; todos los demás toman el valor de 0.
- Por lo tanto, de la tabla de verdad se puede obtener, por inspección, la forma completa de suma de productos y recíprocamente.

Circuitos Lógicos (cont.)

- La forma completa de suma de productos de la expresión de Boole anterior es:

$$Y = AC + A'B$$

$$Y = AC(B + B') + A'B(C + C')$$

$$Y = ABC + AB'C + A'BC + A'BC'$$

Circuitos Lógicos (cont.)

- La tabla de verdad (única) de la expresión de Boole que se obtiene de la forma completa de suma de productos es:

A	00001111
B	00110011
C	01010101
Y	00110101

Expresiones Boolenas Minimales

- Si E es una expresión de Boole de suma de productos, E_L denotará el número de literales en E (contados de acuerdo con la multiplicidad), y E_S denotará el número de sumandos en E.
  - Ejemplo: $E(a,b,c,d) = abc' + a'b'd + ab'c'd + a'bcd$, entonces $E_L = 14$ y $E_S = 4$.
- Sea ahora F una expresión de Boole de suma de productos equivalente de E, entonces se dice que E es más simple que F si $E_L \leq F_L$ y $E_S \leq F_S$, y por lo menos una de las relaciones es una desigualdad estricta.

Expresiones Boolenas Minimales (cont.)

- Una expresión de Boole está en **forma minimal de suma de productos** o **suma minimal**, si está en forma de suma de productos y no hay ninguna otra expresión equivalente en forma de suma de productos que sea más simple que E.
- Un producto fundamental P se llama **implicante primo** de una expresión de Boole E si $P + E = E$, pero ningún otro producto fundamental incluido en P tiene esta propiedad.
  - Ejemplo: $P = \mathbf{xz'}$ es implicante primo de $E(x,y,z) = xy' + \mathbf{xyz'} + x'yz'$.
- **Si una expresión de Boole E está en forma minimal de suma de productos, entonces cada sumando de E es un implicante primo de E.**

Expresiones Boolenas Minimales (cont.)

- El **método de consenso** se puede usar para representar cualquier expresión de Boole como la suma de todos sus implicantes primos.
- Una manera de encontrar una suma minimal para *E* es expresar cada implicante primo en forma completa de suma de productos, y quitar uno por uno aquellos implicantes primos cuyos sumandos aparecen entre los sumandos de los implicantes primos que quedan.

Expresiones Booleanas Minimales (cont.)

- Ejemplo:

$$E(x,y,z) = x'z' + xy + x'y' + yz'$$

$$x'z' = x'z'(y+y') = \underline{x'yz'} + \boxed{x'y'z'}$$ (los sumandos de este implicante primo aparecen en otros, por lo que se elimina)

$$xy = xy(z+z') = xyz + xyz'$$

$$x'y' = x'y'(z+z') = x'y'z + \boxed{x'y'z'}$$

$$yz' = yz'(x+x') = xyz' + \underline{x'yz'}$$

$$E(x,y,z) = xy + x'y' + yz'$$ (ya está en forma de suma minimal)

Expresiones Booleanas Minimales (cont.)

- Ejemplo:

$$E(x,y,z) = x'z' + xy + x'y' + yz'$$

$$x'z' = x'z'(y+y') = \underline{x'yz'} + \boxed{x'y'z'}$$

$$xy = xy(z+z') = xyz + xyz'$$

$$x'y' = x'y'(z+z') = x'y'z + \boxed{x'y'z'}$$

$$yz' = yz'(x+x') = xyz' + \underline{x'yz'}$$ (los sumandos de este implicante primo aparecen en otros, por lo que se elimina)

$$E(x,y,z) = x'z' + xy + x'y'$$ (ya está en forma de suma minimal)

Expresiones Boolenas Minimales (cont.)

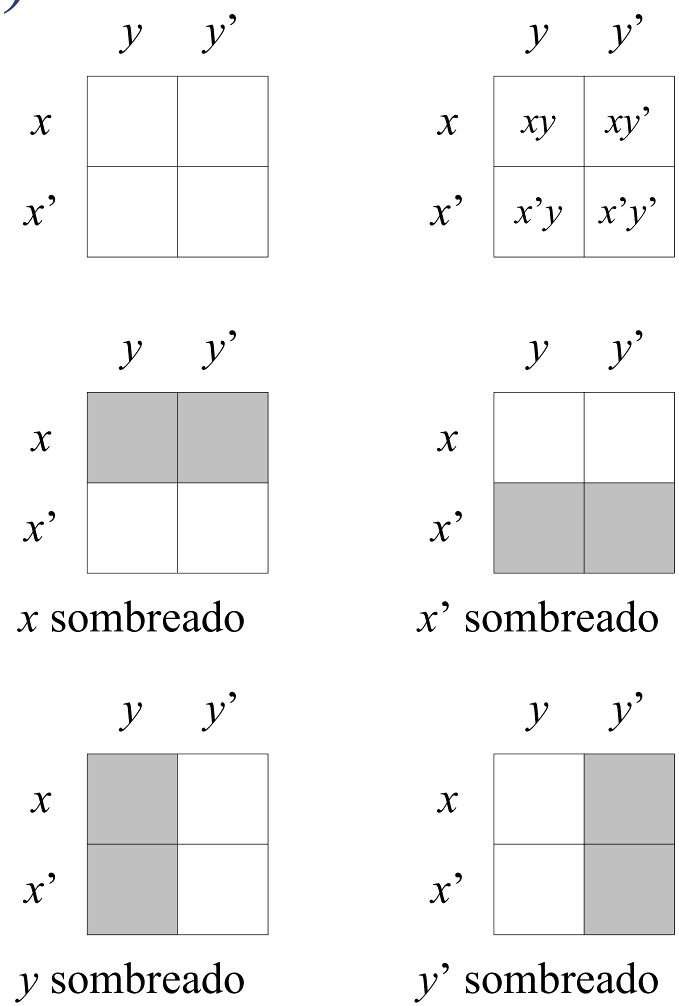
- En el ejemplo anterior se puede quitar alguno de dos implicantes primos, $x'z'$ o yz', y de esta manera se obtiene para la expresión de Boole E dos formas de suma minimal; lo cual muestra que la suma minimal para una expresión de Boole no es necesariamente única.
- El método de consenso para encontrar formas de suma minimal para expresiones de Boole es directo, pero ineficiente.
- Por este motivo, a continuación se dará un método geométrico, llamado mapas de Karnaugh, cuando el número de variables no es muy grande.

Mapas de Karnaugh

- Los **mapas de Karnaugh** son maneras pictóricas de encontrar implicantes primos y formas de sumas minimales para las expresiones de Boole que involucran máximo seis variables.
- Los casos que estudiaremos serán de dos, tres y cuatro variables.
- Estos mapas se representan por cuadrados los productos fundamentales en las mismas variables. Dos productos fundamentales son **adyacentes** si difieren en exactamente un literal, lo cual tiene que ser una variable complementada en un producto y no complementada en el otro.

Mapas de Karnaugh (cont.)

- **Caso de dos variables.**
- Un implicante primo de $E(x,y)$ será una pareja de cuadrados adyacentes o un cuadrado aislado (un cuadrado que no está adyacente a ningún otro cuadrado).



Mapas de Karnaugh (cont.)

- **Caso de dos variables.**
- Ejemplos:

$E_1(x,y) = xy + xy'$

	y	y'
x	✓	✓
x'		

Suma Minimal
$E_1(x,y) = x$

$E_2(x,y) = xy + x'y + x'y'$

	y	y'
x	✓	
x'	✓	✓

Suma Minimal
$E_2(x,y) = x' + y$

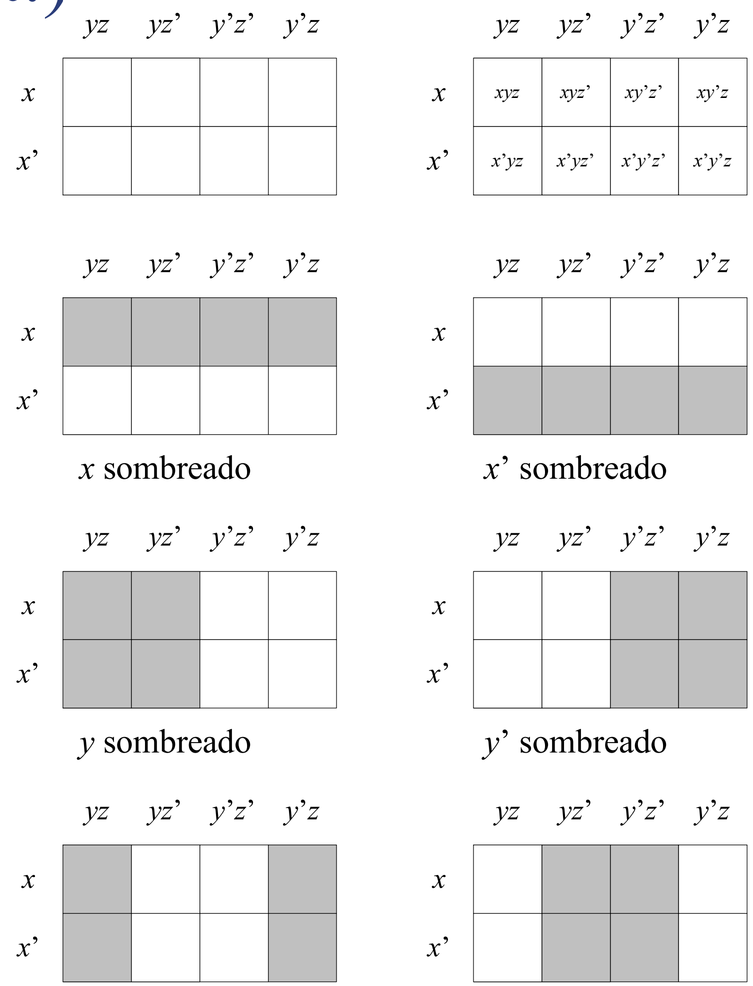
$E_3(x,y) = xy + x'y'$

	y	y'
x	✓	
x'		✓

Suma Minimal
$E_3(x,y) = xy + x'y'$

Mapas de Karnaugh (cont.)

- **Caso de tres variables.**
- Un implicante primo de $E(x,y,z)$ será una pareja de cuadrados adyacentes, un conjunto de cuatro cuadrados adyacentes o un cuadrado aislado (un cuadrado que no está adyacente a ningún otro cuadrado).



x sombreado

x' sombreado

y sombreado

y' sombreado

z sombreado

z' sombreado

Mapas de Karnaugh (cont.)

- **Caso de tres variables.**
- Ejemplos:

$E_1(x,y,z) = xyz + xyz' + x'yz' + x'y'z$

	yz	yz'	$y'z'$	$y'z$
x	✓	✓		
x'		✓		✓

Suma Minimal
$E_1(x,y,z) = xy + yz' + x'y'z$

$E_2(x,y,z) = xyz + xyz' + xy'z + x'yz + x'y'z$

	yz	yz'	$y'z'$	$y'z$
x	✓	✓		✓
x'	✓			✓

Suma Minimal
$E_2(x,y,z) = z + xy$

$E_3(x,y,z) = xyz + xyz' + x'yz' + x'y'z + x'y'z'$

	yz	yz'	$y'z'$	$y'z$
x	✓	✓		
x'		✓	✓	✓

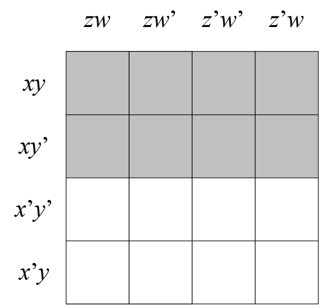
Suma Minimal
$E_3(x,y,z) = xy + yz' + x'y'$
$E_3(x,y,z) = xy + x'z' + x'y'$

Mapas de Karnaugh (cont.)

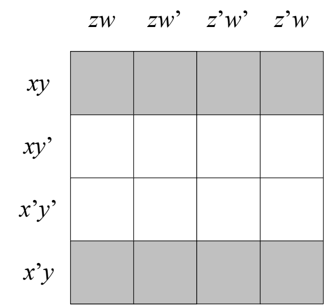
- **Caso de cuatro variables.**
- Un implicante primo de $E(x,y,z,w)$ será una pareja de cuadrados adyacentes, un conjunto de cuatro cuadrados adyacentes, un conjunto de ocho cuadrados adyacentes o un cuadrado aislado (un cuadrado que no está adyacente a ningún otro cuadrado).

	zw	zw’	z’w’	z’w
xy				
xy’				
x’y’				
x’y				

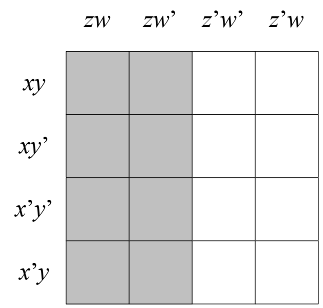
	zw	zw’	z’w’	z’w
xy	xyzw	xyzw’	xyz’w’	xyz’w
xy’	xy’zw	xy’zw’	xy’z’w’	xy’z’w
x’y’	x’y’zw	x’y’zw’	x’y’z’w’	x’y’z’w
x’y	x’yzw	x’yzw’	x’yz’w’	x’yz’w



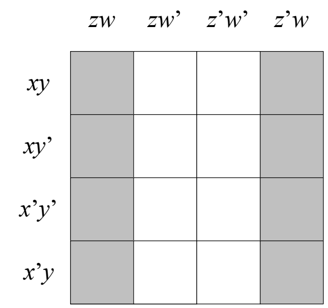
x sombreado



y sombreado



z sombreado



w sombreado

Mapas de Karnaugh (cont.)

- **Caso de cuatro variables.**
- Ejemplos:

$E_1(x,y,z,w) = xyz'w + xyz'w' + xy'zw + xy'zw' + x'y'zw + x'y'zw' + x'yz'w'$

	zw	zw'	$z'w'$	$z'w$
xy			✓	✓
xy'	✓	✓		
$x'y'$	✓	✓		
$x'y$			✓	

Suma Minimal

$E_1(x,y,z,w) = y'z + xyz' + yz'w'$

$E_2(x,y,z,w) = xyzw' + xy'zw + xy'zw' + xy'z'w + xy'z'w' + x'y'zw + x'y'zw' + x'y'z'w + x'y'z'w'$

	zw	zw'	$z'w'$	$z'w$
xy		✓		
xy'	✓	✓	✓	✓
$x'y'$	✓	✓	✓	✓
$x'y$				

Suma Minimal

$E_2(x,y,z,w) = y' + xzw'$

$E_3(x,y,z,w) = xyzw + xyz'w + xy'zw' + x'yzw + x'yz'w + x'y'zw + x'y'zw' + x'y'z'w + x'y'z'w'$

	zw	zw'	$z'w'$	$z'w$
xy	✓			✓
xy'		✓		
$x'y'$	✓	✓	✓	✓
$x'y$	✓			✓

Suma Minimal

$E_3(x,y,z,w) = yw + x'y' + y'zw'$

Conceptos de los K-mapa

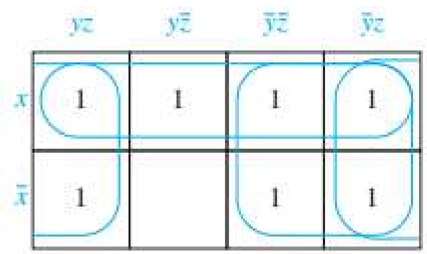
- El producto de los literales que corresponde a un bloque de todos los 1s en el K-mapa (mapa de Karnaugh) se llama implicante de la función que es minimizada.
- Los implicantes primos son los grupos de 1s que se forman en el K-mapa.
- Los implicantes primos esenciales son los grupos de 1s que forman parte de la suma minimal.

Conceptos de los K-mapa
Ejemplo

- Usar K-mapas para minimizar las siguientes sumas de productos. Indique los implicantes, implicantes primos e implicantes primos esenciales.
  - $xyz + xyz' + xy'z + xy'z' + x'yz + x'y'z + x'y'z'$
  - $xyz' + xy'z' + x'y'z + x'y'z'$

Conceptos de los K-mapa
Ejemplo

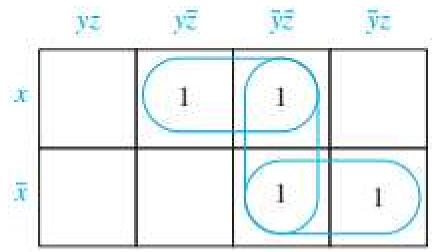
- K-map de $xyz + xyz' + xy'z + xy'z' + x'yz + x'y'z + x'y'z'$



  - Implicantes: $xyz + xyz' + xy'z + xy'z' + x'yz + x'y'z + x'y'z'$
  - Implicantes primos: $x + y' + z$
  - Implicantes primos esenciales: $x + y' + z$

Conceptos de los K-mapa
Ejemplo

- K-map de $xyz' + xy'z' + x'y'z + x'y'z'$



  - Implicantes: $xyz' + xy'z' + x'y'z + x'y'z'$
  - Implicantes primos: $xz' + x'y' + y'z'$
  - Implicantes primos esenciales: $xz' + x'y'$

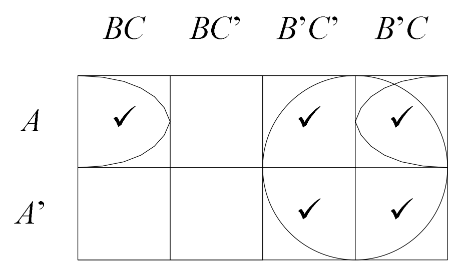
Circuitos Minimales AND-OR

- Se puede aplicar toda la teoría anterior a un importante problema de diseño de circuitos, que tiene dos versiones un poco diferentes:
  - La construcción de un circuito AND-OR cuya expresión de Boole está en la forma de suma minimal (un circuito minimal AND-OR) y que es equivalente a un circuito lógico *L* dado.
  - La construcción de un circuito minimal AND-OR que tendrá una tabla de verdad prescrita.

Circuitos Minimales AND-OR (cont.)

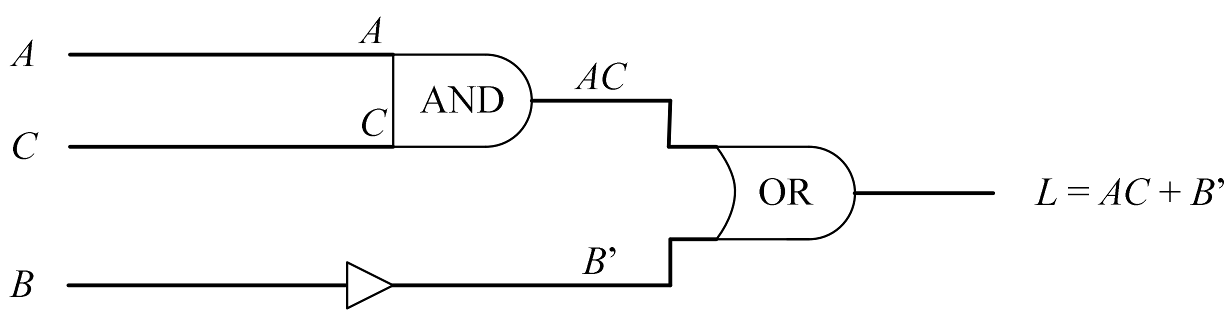
A	00001111
B	00110011
C	01010101
L	11001101

$$L(A,B,C) = A'B'C' + A'B'C + AB'C' + AB'C + ABC$$



Suma Minimal

$$L(A,B,C) = AC + B'$$



Referencias Bibliográficas

- Rosen, Kenneth. "Discrete Mathematics and Its Applications". Séptima Edición, Mc Graw Hill. New York, 2012.
- Jonnsonbaugh, Richard. "Matemáticas Discretas". Prentice Hall, México. Sexta Edición, 2005.