Capítulo 3

LÓGICA DIGITAL

3.1 TRANSISTORES

Hemos dicho ya que las modernas computadoras electrónicas digitales funcionan con base en sólo dos valores posibles de corriente eléctrica, de voltaje o de carga según el caso. Solemos abstraer esto diciendo que trabajan en binario: en ceros y unos. Toda función de nuestro interés, es decir, toda función computable, puede representarse usando sólo ceros y unos para la entrada, para la salida y para especificar las operaciones elementales que la constituyen. Así que nos basta con el conjunto $\{0, 1\}$ como alfabeto para representarlo todo. Hemos dicho además que esto resulta conveniente porque hace nuestros dispositivos más confiables, más tolerantes: los obliga a distinguir sólo entre dos valores posibles y no entre diez. Fue Claude Shannon quien, en su tesis de maestría[1] de 1937, demostró que usando álgebra booleana y aritmética binaria, era posible hacer circuitos de relevadores[2] que calcularan funciones.

Interruptores

Lo esencial para construir una computadora basándose en los conceptos de Shannon, es poseer un dispositivo capaz de conducir o in-

[1]Shannon, Claude Elwood, *A symbolic analysis of relay and switching circuits*, Thesis (M.S.), Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1940. También: Shannon, C. E., "A symbolic analysis of relay and switching circuits", *Transactions of the American Institute of Electrical Engineers*, No. 57, 1938, pp. 713-723.
[2]Un relevador es un interruptor operado eléctricamente. Es decir un dispositivo de conmutación electromecánico. Se les utilizaba en los conmutadores de la red telefónica y fueron utilizados en las primeras computadoras electromecánicas, como la Mark I de Harvard (1944).

A	**B**	**A+B**	**AB**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Tabla 3.1. Tabla de verdad de la disyunción y la conjunción.

terrumpir el paso de la corriente eléctrica: un interruptor. Usando interruptores es posible calcular cualquier operación de las que conocemos de la lógica proposicional. En la figura 3.1 se muestran, por ejemplo dos disposiciones diferentes para conectar un par de interruptores: en paralelo o en serie. Digamos que un interruptor puede estar cerrado (dejando fluir la corriente eléctrica), en cuyo caso diremos que está en posición 1, o abierto (impidiendo el paso de corriente) lo que denotaremos como 0. Diremos que un foco puede estar encendido (1) o apagado (0). Así que en la siguiente tabla están todos los posibles estados de los interruptores A y B de la figura 3.1 y el correspondiente estado del foco.

Por supuesto las funciones calculadas por nuestros circuitos son la *disyunción* y la *conjunción* de la lógica proposicional, que el lector recordará de cursos previos. Como éstas, todas las funciones de la lógica proposicional pueden calcularse usando circuitos de interruptores. Lo mismo ocurre con las funciones aritméticas, cualquiera de ellas, si representamos los números en base 2, puede verse como una tabla similar a la que hemos puesto y puede calcularse usando interruptores. Por cierto, basta poder calcular la disyunción, la conjunción y una función más, una que convierta un 0 en un 1 y viceversa. A esto le llamamos *negación*. Estas tres operaciones forman lo que llamaremos un *conjunto completo* de operaciones, con ellas nos basta para poder expresar cualquier otra función que pueda llevarse a cabo con interruptores.

Por supuesto no queremos que los interruptores tengan que ser operados de forma manual, sino automática. Se necesita poder construir una red de esos interruptores, interconectándolos para que unos determinen el comportamiento de otros. Al principio los relevadores electromecánicos fueron la única alternativa, dada la tecnología disponible. Pero después fueron reemplazados por dispositivos puramente

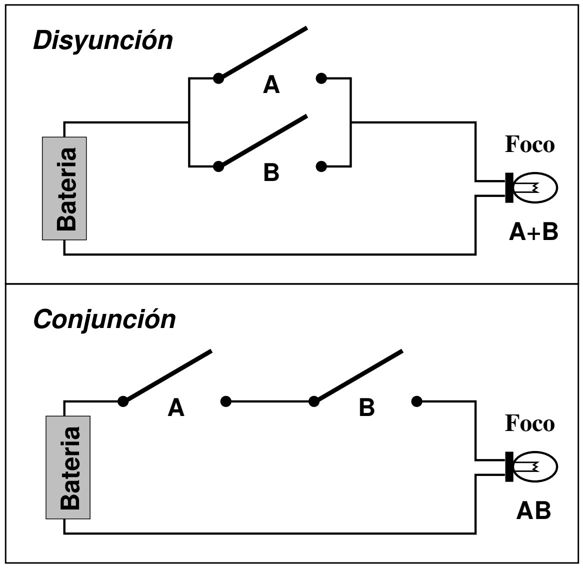

Figura 3.1. Cálculo de la disyunción y la conjunción del cálculo proposicional usando interruptores.

electrónicos. Los primeros de este tipo fueron las válvulas de vacío, conocidos coloquialmente como "bulbos". La desventaja era su velocidad de operación y el hecho de que debían calentarse para funcionar, lo que significa un gran consumo de energía que se disipa en forma de calor.

Las cosas comenzaron a cambiar cuando, a fines de los 40, Bardeen, Brattain y Shockley, basándose en los experimentos de Edgar Lilienfeld de 1923, inventaron el transistor en los laboratorios Bell. Este dispositivo tiene dos propiedades interesantes para la electrónica: puede servir como amplificador de corriente o como interruptor. Por supuesto en el ámbito de la computación digital es ésta segunda la que es fundamental.

Semiconductor

En la naturaleza los materiales pueden dividirse, de acuerdo con su conductividad, en dos grupos fundamentales: los conductores y los aislantes. Los primeros dejan que la corriente eléctrica fluya por ellos, mientras que los segundos no lo permiten; un conductor tiene cargas eléctricas móviles, un aislante no. Esencialmente esto depende de cuantos electrones libres posean los átomos del material. Los conductores tienen muchos, los aislantes muy pocos. Pero existen también materiales que, en estado puro, son aislantes, pero cuando contienen cierto tipo y concentración de impurezas de otro pueden, en cierta circunstancia, conducir la corriente eléctrica. A estos materiales se les denomina *semiconductores*.

Silicio + contaminante

El silicio, por sí solo es un aislante: su resistividad al paso de la corriente eléctrica es alta. Sin embargo si se le agregan átomos de algún otro elemento puede convertirse en conductor. Si se le agrega, por ejemplo, arsénico o fósforo se convierte en un material donante de electrones libres, con carga negativa, por lo que se le llama tipo N; si se le agrega, en cambio, galio o boro, se convierte en un material aceptador, lleno de huecos, lo que le confiere una carga positiva, por lo que se le llama tipo P. Un transistor es un sándwich hecho de una capa de material P flanqueada por dos capas de material N, a lo que se le llama *unión NPN* o bien, a la inversa, una capa de material N flanqueada por capas P, a lo que se le llama *unión PNP*.

En una unión NPN, los electrones libres de una capa N no podrían fluir hacia el otro extremo en la otra capa N, pasando a través de la capa P, porque se los tragarían los huecos de ésta. La capa P actúa como esponja. Si la capa P es suficientemente delgada en contraste con las capas N y se crea un voltaje, entre la capa P y una de las capas N, lo que significa que hay un "empuje" de cargas desde la N a la P, entonces los huecos se llenan del todo y la carga puede fluir de un extremo a otro del transistor. Es como si la carga eléctrica fuera agua, la regiones N son entonces secciones de un tubo llenas de agua y la región P un tapón de esponja en medio de ellas; el voltaje es equivalente a la acción de una bomba en uno de los extremos, si se bombea con suficiente potencia el agua satura la esponja P y el agua termina fluyendo a través del tubo. La única diferencia entre un transistor de unión NPN y uno PNP es la polaridad.

Analogía transistor-válvula

Si un transistor recibe corriente eléctrica por el emisor y no recibe nada por la base entonces es como un interruptor abierto, por el colector no sale nada. Si, en cambio, recibe suficiente corriente por la base, entonces la carga del emisor es "bombeada" a través de la base y del colector. En la figura 3.2 se muestra el corte transversal de un transistor plano, como los que se utilizan en los circuitos integrados actuales. Actualmente el grosor del canal P es de unos 5 o 6 átomos, lo que ocasiona que, aún cuando el transistor esté en estado de corte, algunas cargas se fugan del emisor al colector. A este fenómeno se le llama *fuga de transistor* o *transistor leaking* en inglés y es importante porque, aún cuando la magnitud de la fuga no hace que se confunda un cero con un uno, sí representa un desperdicio de energía que debe disiparse en forma de calor, lo que es muy importante hoy día cuando deseamos dispositivos móviles cuya batería dure mucho tiempo. Sin embargo, recientes avances tecnológicos harán posible fabricar en 2012 chips con tecnología de 22nm y transistores, ya no planos como el mostrado en la

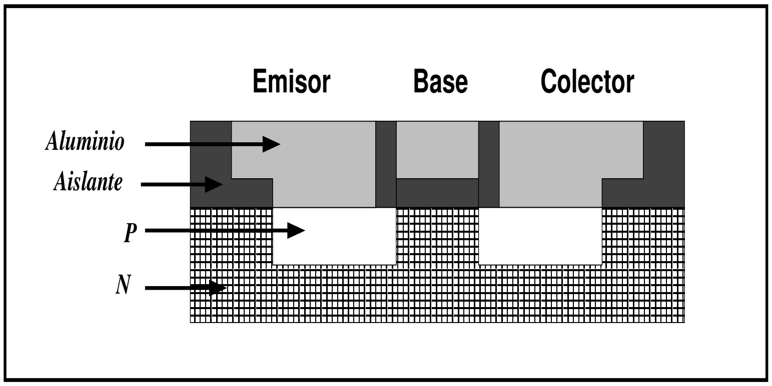

Figura 3.2. Corte transversal de un transistor plano.

figura 3.2, sino tridimensionales, con la base en un plano perpendicular al emisor y al colector. Esto reducirá notablemente la fuga de energía y con ello la duración de la carga de nuestras baterías y reducirá la cantidad de calor producido por los chips. Esta nueva tecnología es llamada Tri-Gate por Intel y FinFET por el resto de la industria.

El siguiente avance luego del uso de transistores no fue, de hecho, un nuevo dispositivo electrónico, sino el tamaño del mismo. Reducir el tamaño de los transistores hizo posible empacar muchos de ellos interconectados para lograr circuitos con alguna función específica. Al empacamiento de circuitos se le llama *integración* y al resultado, por supuesto, *circuito integrado*. Así que el siguiente paso tecnológico en la construcción de computadoras digitales fue el uso de circuitos integrados.

Generaciones de computadoras electrónicas

En la literatura suele emplearse el término *generación* para referirse a los estadios del proceso evolutivo de los dispositivos de cómputo. En general se clasifican como sigue:

Primera generación. Basadas en válvulas de vacío.

Segunda generación. Basadas en circuitos de transistores individuales, es decir, sin empacar.

Tercera generación. Basadas en circuitos integrados a baja escala, unos cientos de transistores por paquete.

Cuarta generación. Basadas en circuitos integrados a gran escala (VLSI o *Very Large Scale Integration*), miles, millones y, hoy día, cientos de miles de millones de transistores por paquete.

3.2 ÁLGEBRA BOOLEANA

El álgebra subyacente a las redes de conmutación de Shannon es un caso particular de la que desarrolló George Boole en 1854 en su trabajo[3] por dotar a la lógica proposicional de un lenguaje algebraico. Más tarde William Stanley Jevons y Edward Huntington le dieron al *álgebra booleana* la forma que hoy tiene. En particular Huntington[4] construyó el álgebra de Boole sobre un conjunto de axiomas en 1904.

3.2.1 Los postulados de Huntington

Un *álgebra booleana* $\mathbb{B}$ es una terna $(B, +, \cdot)$ en la que B es un conjunto con al menos dos elementos distintos, $+$ denota el operador de suma y $\cdot$ el de producto. Si x, y y z son tres elementos cualesquiera de B, el álgebra booleana $\mathbb{B} = (B, +, \cdot)$ satisface axiomas.

1. Cerradura.

   P1a. B es cerrado bajo la operación suma: $x + y \in B$.

   P1b. B es cerrado bajo la operación producto: $x \cdot y \in B$.

2. Identidad.

   P2a. Existe un elemento *identidad de la suma* en B, denotado por 0, tal que: $x + 0 = 0 + x = x$.

   P2b. Existe un elemento *identidad del producto* en B, denotado por 1, tal que: $x \cdot 1 = 1 \cdot x = x$.

3. Conmutatividad.

   P3a. La suma es conmutativa: $x + y = y + x$.

   P3b. El producto es conmutativo: $x \cdot y = y \cdot x$.

4. Distributividad.

   P4a. El producto distribuye a la suma: $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.

   P4b. La suma distribuye al producto: $x + (y \cdot z) = (x + y) \cdot (x + z)$.

5. Complemento. Para cada $x \in B$ existe un único elemento $\overline{x} \in B$, llamado *el complemento* de x, tal que

   P5a. $x + \overline{x} = 1$

[3]Boole, George, *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*, Macmillan, 1854.
[4]Huntington, Edward V., "Sets of independent postulates for the algebra of logic", Transactions of the AMS, Vol. 5, 1904, pp. 288-309.

P5b. $x \cdot \overline{x} = 0$

Por conveniencia consideraremos que, para obtener el complemento de un elemento cualquiera x, se aplica también una operación unaria (de un sólo argumento) y que bien puede verse denotada por la barra que ponemos sobre el argumento cuyo complemento queremos. Esto resulta conveniente porque tendremos entonces una idea clara de que, para construir un álgebra booleana, de acuerdo con los axiomas de Huntington, se requiere de tres operadores: suma, producto y complemento, aún cuando la definición formal sólo contempla dos. En 1933 Huntington reelaboró los axiomas del álgebra booleana para simplificarlos[5]. Es casi un deseo innato de los matemáticos obtener el mínimo número de axiomas que definen un sistema formal, así que Huntington encontró que con sólo tres axiomas y una operación (la suma) se podía tener un álgebra booleana: la conmutatividad, la asociatividad (que en los postulados de 1904, que aquí usaremos, no está contemplada) y la llamada ecuación de Huntington ($\overline{(\overline{x}+y)} + \overline{(\overline{x}+\overline{y})} = x$).

Diversidad de álgebras booleanas

Notará el lector que, además de que la suma, el producto y el complemento se pueden interpretar, respectivamente, como la disyunción, la conjunción y la negación de la lógica proposicional, en cuyo caso el 0 puede interpretarse como *falso* y el 1 como *verdadero*; también se pueden ver como la unión y la intersección y el complemento de la teoría de conjuntos elemental y entonces el 0 hace las veces del conjunto vacío y el 1 se puede interpretar como el conjunto universal. Así las cosas resulta que la teoría de conjuntos elemental, esa en la que el lector se ejercitó en el uso de diagramas de Venn en cursos elementales y la lógica proposicional son, ambas, álgebras booleanas. Además, claro, no son las únicas, este hecho le concedió al álgebra booleana un carácter más general e interesante.

Caso mínimo: dos elementos

El estudio de las álgebras booleanas trascendió el de la lógica proposicional y la teoría de conjuntos básica y se convirtió en un área por sí misma. A nosotros, sin embargo, nos interesa el caso mínimo, cuando el conjunto de elementos es dos, el caso que ocupó a Shannon. Pero antes necesitamos algunas herramientas más. Por simplicidad en adelante omitiremos el operador "·" las más de las veces.

Teorema 3.1 (Idempotencia.)

(a) $x + x = x$

[5] Huntington, E.W. "New sets of independent postulates for the algebra of logic, with special reference to Whithead and Russell's Principia Mathematica", Transactions of the AMS, Vol. 35, 1933, pp. 274-304. Con una corrección posterior: "Boolean algebra. A correction.", ibid., Vol. 35, 1933, pp. 557-558.

(b) $x \cdot x = x$

Demostración:

$$\begin{aligned} x + x &= (x + x) \cdot 1 && P2b \\ &= (x + x)(x + \overline{x}) && P5a \\ &= x + x\overline{x} && P4b \\ &= x + 0 && P5b \\ &= x && P2a \end{aligned}$$

□

Por supuesto el inciso b del teorema se demuestra de manera completamente análoga: reemplazando los 1 por 0, los "+" por "·" y justificando con el inciso complementario de cada postulado.

Teorema 3.2 (Aniquilación)

(a) $x + 1 = 1$

(b) $x \cdot 0 = 0$

Demostración:

$$\begin{aligned} x + 1 &= 1 \cdot (x + 1) && P2b \\ &= (x + \overline{x})(x + 1) && P5a \\ &= x + \overline{x} \cdot 1 && P4b \\ &= x + \overline{x} && P2b \\ &= 1 && P5a \end{aligned}$$

□

Teorema 3.3 (Absorción.)

(a) $x + xy = x$

(b) $x(x + y) = x$

Demostración:

$$\begin{aligned} x + xy &= x \cdot 1 + xy && P2b \\ &= x(1 + y) && P4a \\ &= x(y + 1) && P3a \\ &= x \cdot 1 && Teo.3.2(a) \\ &= x && P2b \end{aligned}$$

$\square$

Teorema 3.4 (Involución.) $\overline{(\overline{x})} = x$.

Demostración: Por el quinto postulado se tiene que las expresiones $x+\overline{x} = 1$ y $x \cdot \overline{x} = 0$ definen al complemento de un elemento cualquiera en B. Si reemplazamos x por su complemento tendremos $\overline{x} + \overline{\overline{x}} = 1$ y $\overline{x} \cdot \overline{\overline{x}} = 0$. Combinándolas tenemos que:

$$x + \overline{x} = \overline{x} + \overline{\overline{x}}$$

y

$$x \cdot \overline{x} = \overline{x} \cdot \overline{\overline{x}}$$

De donde, considerando que el complemento de x es único por el postulado 5, se tiene que:

$$x = \overline{\overline{x}}$$

$\square$

Teorema 3.5 (Eliminación)

(a) $x + \overline{x}\ y = x + y$

(b) $x\ (\overline{x} + y) = xy$

Demostración:

$$\begin{aligned}
x + \overline{x}\ y &= x + xy + \overline{x}\ y && Teo.3.3(a)\\
&= x + 0 + xy + \overline{x}\ y && P2a\\
&= x + x\ \overline{x} + xy + \overline{x}\ y && P5b\\
&= xx + x\ \overline{x} + xy + \overline{x}\ y && Teo.3.1(b)\\
&= xx + xy + x\ \overline{x} + \overline{x}\ y && P3a\\
&= (x+y)(x+\overline{x}) && P4a\\
&= (x+y)\cdot 1 && P5a\\
&= x + y && P2b
\end{aligned}$$

$\square$

Teorema 3.6 (Asociatividad.)

(a) $x + (y + z) = (x + y) + z$

(b) $x \cdot (y \cdot z) = (x \cdot y) \cdot z$

Demostración: Sean $I = x + (y + z)$ y $D = (x + y) + z$, los respectivos lados izquierdo y derecho de la expresión a comprobar. Nótese que:

$$x \cdot I = x \cdot [x + (y + z)] = x \cdot (x + E)$$

donde $E = y + z$. Esa última expresión es de la forma $x(x + y)$, el lado izquierdo del inciso (b) del teorema 3.3 (absorción). Así que:

$$\begin{aligned} x \cdot [x + (y + z)] &= x \cdot I \\ &= x \quad Teo.3.3(b) \end{aligned} \tag{3.1}$$

Por otra parte:

$$\begin{aligned} x \cdot [(x + y) + z] &= x \cdot D \\ &= x \cdot (x + y) + xz \qquad P4a \\ &= x + xz \qquad Teo.3.3(b) \\ &= x \qquad Teo.3.3(a) \end{aligned} \tag{3.2}$$

Sintetizando 3.1 y 3.2:

$$x \cdot [x + (y + z)] = xI = x = xD = x \cdot [(x + y) + z] \tag{3.3}$$

Además:

$$\begin{aligned} \overline{x} \cdot [x + (y + z)] &= \overline{x} \cdot I \\ &= \overline{x}x + \overline{x}(y + z) \qquad P4a \\ &= x\overline{x} + \overline{x}(y + z) \qquad P3b \\ &= 0 + \overline{x}(y + z) \qquad P5b \\ &= \overline{x}(y + z) \qquad P2a \end{aligned} \tag{3.4}$$

Y también:

$$\begin{aligned} \overline{x} \cdot [(x + y) + z] &= \overline{x} \cdot D \\ &= \overline{x}(x + y) + \overline{x}z \qquad P4a \\ &= (\overline{x}x + \overline{x}y) + \overline{x}z \qquad P4a \\ &= (x\overline{x} + \overline{x}y) + \overline{x}z \qquad P3b \\ &= (0 + \overline{x}y) + \overline{x}z \qquad P5b \\ &= \overline{x}y + \overline{x}z \qquad P2a \\ &= \overline{x}(y + z) \qquad P4a \end{aligned} \tag{3.5}$$

Sintetizando 3.4 y 3.5:

$$\overline{x} \cdot [x + (y + z)] = \overline{x}I = \overline{x}(y + z) = \overline{x}D = \overline{x} \cdot [(x + y) + z] \tag{3.6}$$

Finalmente:

$$\begin{aligned} I &= I \cdot 1 & \\ &= I(x + \overline{x}) & P5a \\ &= Ix + I\overline{x} & P4a \\ &= xI + \overline{x}I & P3b \\ &= xD + \overline{x}I & 3.3 \\ &= xD + \overline{x}D & 3.6 \\ &= Dx + D\overline{x} & P3b \\ &= D(x + \overline{x}) & P4a \\ &= D \cdot 1 & P5a \\ &= D & P2b \end{aligned}$$

Con lo que $x + (y + z) = (x + y) + z$. □

Teorema 3.7 (Consenso.)

(a) $xy + \overline{x}z + yz = xy + \overline{x}z$

(b) $(x + y)(\overline{x} + z)(y + z) = (x + y)(\overline{x} + z)$

Demostración:

$$\begin{aligned} xy + \overline{x}z + yz &= xy + \overline{x}z + yz \cdot 1 & P2b \\ &= xy + \overline{x}z + yz \cdot (x + \overline{x}) & P5a \\ &= xy + \overline{x}z + xyz + \overline{x}yz & P4a \\ &= xy(1 + z) + \overline{x}z(1 + y) & P4a \\ &= xy \cdot 1 + \overline{x}z \cdot 1 & Teo.3.2 \\ &= xy + \overline{x}z & P2b \end{aligned}$$

□

Teorema 3.8 (Leyes de De Morgan.)

(a) $\overline{x + y} = \overline{x} \cdot \overline{y}$

(b) $\overline{x \cdot y} = \overline{x} + \overline{y}$

Demostración: Por el postulado 5 sabemos que, para todo elemento x existe un único elemento $\overline{x}$, tal que: $x + \overline{x} = 1$ y $x \cdot \overline{x} = 0$. Así que para probar que $\overline{x} \cdot \overline{y}$ es el complemento de $x + y$, basta probar que $(x + y) + \overline{x} \cdot \overline{y} = 1$ y que $(x + y) \cdot (\overline{x} \cdot \overline{y}) = 0$.

En efecto:

$$
\begin{aligned}
(x+y) + \overline{x}\,\overline{y} &= [(x+y)+\overline{x}] \cdot [(x+y)+\overline{y}] && P4b \\
&= [(y+x)+\overline{x}] \cdot [(x+y)+\overline{y}] && P3a \\
&= [y+(x+\overline{x})] \cdot [x+(y+\overline{y})] && Teo.3.6(a) \\
&= (y+1)\cdot(x+1) && P5a \\
&= 1 \cdot 1 && Teo.3.2(a) \\
&= 1 && P2b
\end{aligned}
$$

También:

$$
\begin{aligned}
(x+y)\cdot(\overline{x}\,\overline{y}) &= (\overline{x}\,\overline{y})\cdot(x+y) && P3b \\
&= (\overline{x}\,\overline{y})x + (\overline{x}\,\overline{y})y && P4a \\
&= (\overline{y}\,\overline{x})x + (\overline{x}\,\overline{y})y && P4a \\
&= \overline{y}(\overline{x}\,x) + \overline{x}(\overline{y}\,y) && Teo.3.6(b) \\
&= \overline{y}(x\,\overline{x}) + \overline{x}(y\,\overline{y}) && P3b \\
&= \overline{y}\cdot 0 + \overline{x}\cdot 0 && P5b \\
&= 0 + 0 && Teo.3.2(b) \\
&= 0 && Teo.3.1(a)
\end{aligned}
$$

□

3.3 FUNCIONES DE CONMUTACIÓN

El conjunto B sobre el que se definen las operaciones del álgebra booleana tiene, sabemos, al menos dos elementos. Cuando tiene exactamente ese número, estos elementos son los que en los axiomas aparecen etiquetados como “0” y “1”. Podemos ahora pensar entonces en funciones cuyas variables independientes puedan adquirir valores en ese conjunto mínimo y calculen, a partir de ello, un valor también en ese conjunto. A estas, un caso particular del álgebra booleana, les llamaremos *funciones de conmutación*.

Regla de correspondencia o tabla de verdad

¿Cómo se puede expresar una función? Bueno, estamos acostumbrados a hacerlo mediante lo que llamamos la *regla de correspondencia*: una expresión analítica que proporciona el valor de la variable dependiente, dado el de la o las variables independientes. Pero eso es porque solemos trabajar con los números reales y en ese conjunto, por ser infinito y continuo, es imposible pensar en otra forma de hacerlo. Pero en estricto sentido una función es una relación entre dos conjuntos: cada elemento de uno de ellos, llamado dominio, se asocia con uno y sólo uno del otro conjunto, llamado contradominio. Así las cosas se puede pensar en hacer una tabla que diga explícitamente, para cada elemento del dominio, cuál elemento del contradominio le es asociado por la función. Claro esto sólo es posible si realmente podemos listar todos los elementos del dominio, es decir, si éste es finito. Si el conjunto dominio de nuestra álgebra booleana contiene el mínimo número de elementos, es decir dos, entonces cualquier función de n variables independientes puede tener hasta 2^n posibles combinaciones de dichas variables de entrada y entonces podemos hacer una tabla que muestre explícitamente cuál es el valor que adquiere la función dada cualquiera de las 2^n posibles entradas. En el álgebra de las funciones de conmutación tenemos entonces dos maneras de especificar una función: con una expresión algebraica o con una tabla análoga a las tablas de verdad que solíamos usar en el cálculo proposicional.

Por cierto, si pensamos en funciones de conmutación de n variables, entonces, cómo dijimos, las variables pueden adquirir 2^n valores diferentes. Por cada una de estas combinaciones la función puede arrojar uno de dos valores posibles, así que en total hay 2^{2^n} funciones de conmutación de n variables.

Resulta además bastante sencillo pasar de la representación de una función mediante su tabla de verdad a su equivalente regla de correspondencia.

■ **EJEMPLO 3.1**

Observemos la función $F(x_0, x_1, x_2)$ representada en la tabla 3.2. Para obtener la expresión analítica de esta función debemos fijar nuestra atención en el primer renglón de la tabla en el que F vale 1, a saber, el primer renglón de la tabla. Si sabemos que en ese renglón $x_2 = x_1 = x_0 = 0$, ¿qué deberíamos hacerle a las variables para obtener justo ese y sólo ese 1 operándolas con un AND? Cada variable vale 0, y basta con un cero en un AND para que el resultado sea 0, así que necesariamente deberemos negar todas las variables, es decir, la función $m_0 = \overline{x_0}\ \overline{x_1}\ \overline{x_2}$ tiene la propiedad de valer 1 en un único sitio en la tabla, justo cuando $x_2 = x_1 = x_0 = 0$, que es lo que buscamos.

$\mathbf{x_2}$	$\mathbf{x_1}$	$\mathbf{x_0}$	**F**
0	0	0	1
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

Tabla 3.2. Tabla de verdad de una función de conmutación.

Podemos luego hacernos la misma pregunta para cada renglón de la tabla en el que F valga 1 y obtendremos cada uno de los productos (AND) que lo generan. Bastaría luego con hacer el OR de todos estos productos para obtener una expresión analítica de F.

Lo que estamos haciendo es, implícitamente, encontrar cuales deben ser los valores de los coeficientes c_i en la expresión:

$$\begin{aligned} F(x_0, x_1, x_2) \quad = \quad & c_0\overline{x_2}\ \overline{x_1}\ \overline{x_0}\ + c_1\overline{x_2}\ \overline{x_1}\ x_0\ + \\ & c_2\overline{x_2}\ x_1\ \overline{x_0}\ + c_3\overline{x_2}\ x_1\ x_0\ + \\ & c_4 x_2\ \overline{x_1}\ \overline{x_0}\ + c_5 x_2\ \overline{x_1}\ x_0\ + \\ & c_6 x_2\ x_1\ \overline{x_0}\ + c_7 x_2\ x_1\ x_0 \end{aligned} \tag{3.7}$$

donde $c_i \in \{0, 1\}$. Cada coeficiente indica, realmente, si el i-ésimo producto que le corresponde debe ser considerado o no para construir la función.

Con base en lo expuesto, la función F se puede expresar como:

$$\begin{aligned} F(x_0, x_1, x_2) \quad = \quad & \overline{x_2}\ \overline{x_1}\ \overline{x_0} + \overline{x_2}x_1\overline{x_0} + \overline{x_2}\ x_1 x_0 + \\ & x_2\overline{x_1}x_0 + x_2 x_1\overline{x_0} + x_2 x_1 x_0 \end{aligned} \tag{3.8}$$

■

Mintérmino	Expresión
m_0	$\overline{x_2}\ \overline{x_1}\ \overline{x_0}$
m_1	$\overline{x_2}\ \overline{x_1}\ x_0$
m_2	$\overline{x_2}\ x_1\ \overline{x_0}$
m_3	$\overline{x_2}\ x_1\ x_0$
m_4	$x_2\ \overline{x_1}\ \overline{x_0}$
m_5	$x_2\ \overline{x_1}\ x_0$
m_6	$x_2\ x_1\ \overline{x_0}$
m_7	$x_2\ x_1\ x_0$

Tabla 3.3. Lista de los mintérminos posibles de tres variables.

3.3.1 Suma de productos o producto de sumas

Mintérminos

Podríamos, de acuerdo con lo que hicimos en el ejemplo, expresar cualquier función de conmutación de tres variables, como la suma de algunos de los 8 productos que se pueden formar tomando cada variable o su negación. En la literatura a estos productos se les suele llamar *mintérminos*. En la tabla 3.3 se muestran todos los mintérminos (m_i) de tres variables.

■ **EJEMPLO 3.2**

La función del ejemplo puede expresarse como suma de mintérminos como:

$$F(x_0, x_1, x_2) = m_0 + m_2 + m_3 + m_5 + m_6 + m_7 = \sum(0, 2, 3, 5, 6, 7)$$

■

Por supuesto no hay una única expresión analítica para la regla de correspondencia que defina una función. De hecho podemos hacer lo que podríamos llamar el *procedimiento dual* del que llevamos a cabo: en vez de ir agregando productos a una suma de tal forma que vayamos generando cada uno de los 1's presentes en la tabla, podemos tratar de generar los 0's.

Maxtérmino	Expresión
M_0	$x_2 + x_1 + x_0$
M_1	$x_2 + x_1 + \overline{x_0}$
M_2	$x_2 + \overline{x_1} + x_0$
M_3	$x_2 + \overline{x_1} + \overline{x_0}$
M_4	$\overline{x_2} + x_1 + x_0$
M_5	$\overline{x_2} + x_1 + \overline{x_0}$
M_6	$\overline{x_2} + \overline{x_1} + x_0$
M_7	$\overline{x_2} + \overline{x_1} + \overline{x_0}$

Tabla 3.4. Lista de los maxtérminos posibles de tres variables.

Cada vez que añadimos un mintérmino a una expresión, forzamos a que, en el lugar de la tabla que corresponde con ese mintérmino la función valdrá 1. Además cada mintérmino aporta uno y sólo un 1. La suma de mintérminos, literalmente, suma las contribuciones de cada uno para generar todos los 1 de la tabla.

Maxtérminos

En pensamiento dual corresponde a considerar un término hecho por la suma de cada variable o su negación de tal forma que sólo valga 0 en un lugar de la tabla, en todos los demás lugares la suma valdrá 1. Si tenemos todos los términos necesarios para generar cada uno de los ceros de la tabla y hacemos el producto de ellos, entonces la tabla resultante tendrá ceros exactamente en los lugares asociados con cada término y unos en el resto. A estos términos se les denomina *maxtérminos*. Los maxtérminos de 3 variables aparecen listados en la tabla 3.4.

■ EJEMPLO 3.3

La función que estamos usando como ejemplo puede escribirse también como:

$$\begin{aligned} F(x_0, x_1, x_2) &= (x_2 + x_1 + \overline{x_0})(\overline{x_2} + x_1 + x_0) \\ &= M_1 \cdot M_4 \\ &= \prod(1, 4) \end{aligned}$$

■

A la expresión de la función como suma de productos (mintérminos) se le denomina *Forma Canónica Normal Disjuntiva* o *CDNF* por sus siglas en inglés y a el equivalente como producto de sumas (maxtérminos) *Forma Canónica Normal Conjuntiva* o *CCNF*. Por supuesto una de ellas puede siempre ser transformada en la otra usando las leyes de De Morgan y, claro está esto puede hacerse algorítmicamente, sólo que en principio requiere una cantidad de pasos que depende exponencialmente del número de variables involucradas. Es por eso que el conocido problema SAT, cuya entrada es una expresión booleana en forma conjuntiva, es intratable, a pesar de que al poner la expresión en forma disyuntiva resulta trivial decidir si es satisfactible o no.

3.4 MINIMIZACIÓN DE FUNCIONES DE CONMUTACIÓN

3.4.1 Manipulación algebraica

Existen entonces varias posibles expresiones para la misma función, pero ciertamente debe haber una que es mínima en el sentido de tener la menor cantidad posible de operaciones. Podemos tratar de obtenerla mediante transformaciones sucesivas usando los axiomas y teoremas del álgebra booleana.

■ **EJEMPLO 3.4**

Podemos simplificar la expresión para la función F del ejemplo 3.1 haciendo lo siguiente.

$$
\begin{aligned}
F(x_0, x_1, x_2) &= \overline{x_2}\ \overline{x_1}\ \overline{x_0} + \overline{x_2} x_1 \overline{x_0} + \overline{x_2}\ x_1 x_0 + \\
&\quad x_2 \overline{x_1} x_0 + x_2 x_1 \overline{x_0} + x_2 x_1 x_0 \\
&= \overline{x_2}\ \overline{x_1}\ \overline{x_0} + \overline{x_2}\ x_1 x_0 + \\
&\quad x_1 \overline{x_0}(x_2 + \overline{x_2}) + x_2 x_0 (x_1 + \overline{x_1}) && P4a \\
&= \overline{x_0}(x_1 + \overline{x_1}\ \overline{x_2}) + x_0(x_2 + \overline{x_2}\ x_1) && P5a\ y\ P4a \\
&= \overline{x_0}(x_1 + \overline{x_2}) + x_0(x_2 + x_1) && Teo.3.5(a) \\
&= \overline{x_0}\ x_1 + \overline{x_0}\ \overline{x_2} + x_0 x_2 + x_0 x_1 && P4a \\
&= x_1(x_0 + \overline{x_0}) + \overline{x_0}\ \overline{x_2} + x_0 x_2 && P4a \\
&= x_1 + \overline{x_0}\ \overline{x_2} + x_0 x_2 && P5a
\end{aligned}
$$

■

Simplificar no es simple

Llevar a cabo este proceso no es una labor fácil, en general dependemos de nuestra habilidad para notar qué axioma o teorema del álgebra booleana se puede usar, es truculento. En general, obtener la mínima expresión de una formula booleana dada es un problema NP-difícil. Pero podemos facilitarnos la vida un poco si reflexionamos acerca de lo que acabamos de hacer en el ejemplo. Nótese que siempre que podemos eliminar una variable (o su negación) de algunos términos es cuando esos términos tienen una subexpresión común que se puede *factorizar* y la variable que se elimina aparece en algunos de los términos afirmada y en otros negada. La variable se elimina gracias al hecho de que $x + \overline{x} = 1$ o $x\ \overline{x} = 0$. Es decir, siempre que tenemos una subexpresión común asociada con una variable que *cambia* entonces la variable desaparece. Si tenemos una función que contiene a los mintérminos 4 ($x_2\ \overline{x_1}\ \overline{x_0}$) y 6 ($x_2\ x_1\ \overline{x_0}$), por ejemplo, entonces tenemos como subexpresión común factorizable: $x_2\ \overline{x_0}$ y la variable x_1 cambia de afirmada a negada entre ambos términos, así que se eliminará dejando sólo la subexpresión común.

Es conveniente notar que, si consideramos la expresión binaria de los índices de los mintérminos, lo que equivale a considerar las variables afirmadas (1 en la expresión binaria del índice) y las negadas (0 en la expresión), entonces necesitamos considerar mintérminos cuyos índices, en binario, difieran en un bit. La subexpresión común factorizable que se preserva corresponde a aquellos bits que son comunes a ambos índices. La variable que se elimina es la que cambia de 0 a 1. En el ejemplo del párrafo anterior, los índices eran $6_{10} = 110_2$ y $4_{10} = 100_2$. Denotando con $*$ el sitio que cambia entre ambos mintérminos: $1 * 0$. Los mintérminos 4 y 6 se simplifican como $x_2\ \overline{x_0}$, x_2 corresponde con el 1 y $\overline{x_0}$ con el cero. La variable x_1 quedó en el lugar del $*$, así que se pierde.

Podemos pensar en aprovechar esta observación para tratar de simplificar la expresión de una función tratando de distinguir visualmente las subexpresiones comunes y las variables que cambian.

3.4.2 Mapas de Karnaugh

Heurística

Entre 1952 y 1953 Edward Veitch[6] y Maurice Karnaugh[7] inventaron un método para simplificar funciones de conmutación basado en el prin-

[6]Veitch, Edward W.,"A Chart Method for Simplifying Truth Functions", *Proceedings of the 1952 ACM Annual Meeting*, 1952, pp. 127–133.

[7]Karnaugh, Maurice, "The Map Method for Synthesis of Combinational Logic Circuits". *Transactions of the American Institute of Electrical Engineers*, Vol. 72, No. 9, noviembre de 1953, pp. 593–599.

cipio descrito y aprovechando la habilidad humana para reconocer patrones. El método, cabe decirlo, es una heurística, es decir, en general no tenemos garantías de obtener la expresión mínima para la función, sobre todo si tiene muchas variables, porque entonces los patrones son menos evidentes.

El método consiste en acomodar la tabla de verdad de la función en un mapa cuadriculado, de tal forma que en el mapa queden contiguas las entradas de la tabla cuyo índice difiere en sólo un bit y posteriormente agrupar las celdas de acuerdo con ciertas reglas. Por supuesto en un arreglo bidimensional, conforme el número de variables crece, se vuelve difícil hacer que sean contiguas las celdas que difieren de otras en sólo un bit. Se comienza considerando que las celdas en el extremo derecho son contiguas a las del extremo izquierdo y las de arriba a las de abajo, luego se complica. Describiremos aquí las que se deben usar para obtener sumas de productos, por supuesto si se usan las reglas duales, se obtendrán productos de sumas.

- Los grupos no pueden incluir celdas con valor 0.
- Los grupos pueden crecer horizontal o verticalmente, pero no en diagonal.
- Los grupos deben contener un número de celdas de la forma 2^k.
- Cada grupo debe ser tan grande como sea posible.
- Toda celda con valor 1, debe estar en, al menos un grupo.
- Los grupos pueden tener intersección no vacía.
- Se debe procurar minimizar el número de grupos.

Ciertamente es más fácil comprender esto mediante un ejemplo.

■ **EJEMPLO 3.5**

El mapa de Karnaugh para la función F del ejemplo, se muestra en la figura 3.3. Cómo se puede apreciar en el mapa:

$$F(x_0, x_1, x_2) = \overline{x_2}\ \overline{x_0} + x_1 + x_2 x_0$$

que es justamente la expresión obtenida previamente por manipulación algebraica.

■

x_2 \ $x_1\,x_0$	00	01	11	10
0	1	0	1	1
1	0	1	1	1

Figura 3.3. Mapa de Karnaugh para la función del ejemplo.

$\mathbf{x_0}$	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{F_2}$	$\mathbf{x_0}$	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{F_2}$
0	0	0	0	1	1	0	0	0	1
0	0	0	1	0	1	0	0	1	0
0	0	1	0	1	1	0	1	0	1
0	0	1	1	0	1	0	1	1	0
0	1	0	0	1	1	1	0	0	1
0	1	0	1	1	1	1	0	1	1
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0

Tabla 3.5. Tabla de verdad de la función F_2.

■ **EJEMPLO 3.6**

Sea $F_2(x_0, x_1, x_2, x_3)$ una función descrita por la tabla de verdad 3.5.

El mapa de Karnaugh para F_2 se puede ver en la figura 3.4, de allí se desprende que:

$$F_2(x_0, x_1, x_2, x_3) = \overline{x_1}\ \overline{x_3} + x_1\ \overline{x_2}$$

■

Cuando el número de variables crece, el mapa de Karnaugh ya no resulta tan práctico. Los patrones ya no son tan evidentes. La noción de celdas contiguas en términos del código binario ya no se mapea en una

$x_0\ x_1 \backslash x_2\ x_3$	00	01	11	10
00	1	0	0	1
01	1	1	0	0
11	1	1	0	0
10	1	0	0	1

Figura 3.4. Mapa de Karnaugh para la F_2.

contigüidad espacial. Hay, de hecho, dos alternativas para acomodar una tabla de 5 variables, Hay quienes prefieren una o la otra.

■ **EJEMPLO 3.7**

En la figura 3.5 se muestran las dos alternativas usuales para un mapa de Karnaugh de 5 variables. Como puede observarse los patrones que dan lugar a simplificaciones se ven diferentes en ambos mapas.

La función representada es:

$$\begin{aligned}
F_3(x_0, x_1, x_2, x_3, x_4) \quad = \quad & \overline{x_0}\ \overline{x_1}\ \overline{x_2}\ \overline{x_3}\ \overline{x_4} + \overline{x_0}\ \overline{x_1}\ \overline{x_2}\ x_3\ \overline{x_4} + \\
& \overline{x_0}\ \overline{x_1}\ x_2\ x_3\ \overline{x_4} + \overline{x_0}\ \overline{x_1}\ x_2\ \overline{x_3}\ \overline{x_4} + \\
& \overline{x_0}\ x_1\ \overline{x_2}\ \overline{x_3}\ \overline{x_4} + \overline{x_0}\ x_1\ \overline{x_2}\ \overline{x_3}\ x_4 + \\
& \overline{x_0}\ x_1\ \overline{x_2}\ x_3\ x_4 + \overline{x_0}\ x_1\ x_2\ \overline{x_3}\ \overline{x_4} + \\
& x_0\ x_1\ \overline{x_2}\ \overline{x_3}\ \overline{x_4} + x_0\ x_1\ \overline{x_2}\ \overline{x_3}\ x_4 + \\
& x_0\ x_1\ \overline{x_2}\ x_3\ x_4 + x_0\ x_1\ x_2\ x_3\ x_4 + \\
& x_0\ x_1\ x_2\ \overline{x_3}\ x_4 + x_0\ x_1\ x_2\ \overline{x_3}\ \overline{x_4} + \\
& x_0\ \overline{x_1}\ \overline{x_2}\ \overline{x_3}\ \overline{x_4} + x_0\ \overline{x_1}\ x_2\ x_3\ x_4 + \\
& x_0\ \overline{x_1}\ x_2\ \overline{x_3}\ x_4 + x_0\ \overline{x_1}\ x_2\ \overline{x_3}\ \overline{x_4}
\end{aligned}$$

De acuerdo con el mapa:

$$F_3(x_0, x_1, x_2, x_3, x_4) = \overline{x_3}\ \overline{x_4} + \overline{x_0}\ \overline{x_1}\ x_3\ \overline{x_4} + x_1\ \overline{x_2}\ x_4 + x_0\ x_2\ x_4$$

■

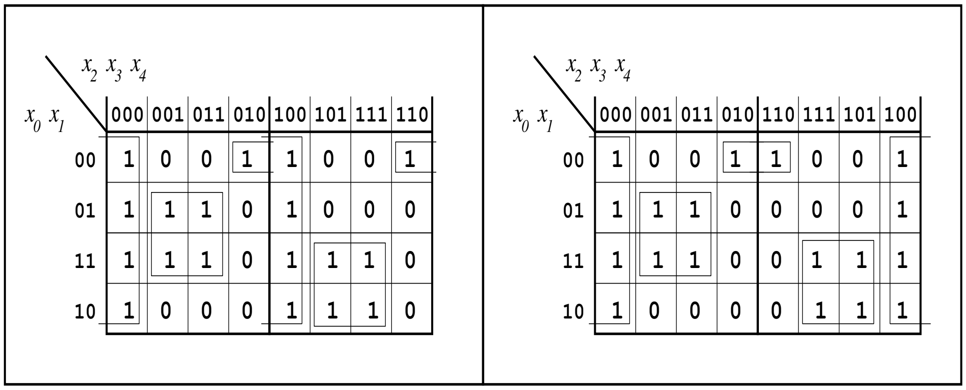

Figura 3.5. Alternativas de mapa de 5 variables..

3.4.3 Método de Quine–McCluskey

Entre 1952 y 1956 Willard Van Orman Quine y Edward McCluskey elaboraron un método algorítmico para la minimización de una función de conmutación. Dado que es un algoritmo, se puede programar y entonces se eliminan los problemas inherentes al método de Karnaugh: la expresión obtenida es ciertamente mínima y correcta y podemos pensar en minimizar funciones de más de 5 variables, cuando la capacidad humana para reconocer patrones se ve seriamente afectada por el orden del mapa de Karnaugh. Podemos olvidarnos pues, de los posibles errores humanos.

El algoritmo encuentra lo que se denominan los *implicantes primos* de la expresión original dada en mintérminos. Se parte de una tabla en la que aparecen los renglones de la tabla de verdad de la función en los que esta vale 1, pero agrupados por su *peso de Hamming*, o sea por el número de unos en su expresión binaria. Se procede luego como sigue:

1. Se empata cada miembro de un grupo con todos y cada uno de los del grupo contiguo siguiente con los que difiera en el valor de una sola variable.

2. Cada pareja genera un nuevo patrón de empate que es registrado en una nueva entrada de la que llamaremos tabla siguiente. En este patrón se elimina aquella variables cuyo valor cambió en los términos que se empataron.

3. Se marcan como usadas aquellas entradas de la tabla actual que se pudieron empatar.

4. Si la tabla siguiente tiene sólo un patrón o patrones imposibles de empatar entonces ya se ha terminado el proceso, si no es así la tabla actual se reemplaza por la tabla nueva y se regresa al primer paso.

Los patrones que quedan al final y aquellos que nunca fueron marcados durante el proceso definen los implicantes primos de la función.

■ **EJEMPLO 3.8**

Recordemos la función F de nuestro ejemplo:

$$F(x_2, x_1, x_0) = \sum(0, 2, 3, 5, 6, 7)$$

Las tablas del proceso de Quine–McCluskey son las siguientes, se han marcado con un "*" los patrones usados.

m	x_2	x_1	x_0	
0	0	0	0	*
2	0	1	0	*
3	0	1	1	*
5	1	0	1	*
6	1	1	0	*
7	1	1	1	*

m	x_2	x_1	x_0	
0,2	0	–	0	
2,3	0	1	–	*
2,6	–	1	0	*
3,7	–	1	1	*
5,7	1	–	1	
6,7	1	1	–	*

m	x_2	x_1	x_0
2,3,6,7	–	1	–
2,6,3,7	–	1	–

Como se puede observar los únicos patrones no usados son el primero y el quinto de la segunda tabla y el único de la última.

Estos patrones corresponden, respectivamente a:

$$F(x_2, x_1, x_0) = \overline{x_2}\ \overline{x_0} + x_2\ x_0 + x_1$$

lo que coincide con nuestras expresines previas para esa función.

■

Por ser algorítmico, el método de Quine–McCluskey fue utilizado en el software usado para diseñar hardware. Pero dado que el problema de minimizar una expresión booleana es, en general intratable, desde hace tiemp se prefiere una heurística bastante efectiva comunmente llamada *Espresso*, desarrollada en los ochenta por Robert Brayton, de IBM y más tarde por Richard Rudell y que resulta mucho más eficiente en términos de consumo de recursos y, aunque no garantiza el mínimo global, ofrece muy buenas aproximaciones.