Los teoremas de DeMorgan para cualquier número de variables tienen una forma similar al caso de dos variables y se deducen por sustituciones sucesivas como se hizo en la deducción anterior. Estos teoremas se pueden generalizar así:

$$(A + B + C + D + \ldots + F)' = A'B'C'D' \ldots F'$$
$$(ABCD \ldots F)' = A' + B' + C' + D' + \ldots + F'$$

La forma generalizada del teorema de DeMorgan dice que el complemento de una función se obtiene intercambiando operadores AND y OR y complementando cada literal.

EJEMPLO 2-2

Halle el complemento de las funciones $F_1 = x'yz' + x'y'z$ y $F_2 = x(y'z' + yz)$. Aplicando el teorema de DeMorgan tantas veces como sea necesario, se obtienen los complementos como sigue:

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$$
$$F_2' = [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')'(yz)'$$
$$= x' + (y + z)(y' + z')$$

■

Un procedimiento más sencillo para deducir el complemento de una función consiste en obtener el dual de la función y complementar cada literal. Este método es consecuencia del teorema generalizado de DeMorgan. Recuerde que el dual de una función se obtiene intercambiando operadores AND y OR, y unos y ceros.

EJEMPLO 2-3

Determine el complemento de las funciones F_1 y F_2 del ejemplo 2-2 obteniendo sus duales y complementando cada literal.

1. $F_1 = x'yz' + x'y'z$.
   El dual de F_1 es $(x' + y + z')(x' + y' + z)$.
   Complementamos cada literal: $(x + y' + z)(x + y + z') = F_1'$.
2. $F_2 = x(y'z' + yz)$.
   El dual de F_2 es $x + (y' + z')(y + z)$.
   Complementamos cada literal: $x' + (y + z)(y' + z') = F_2'$.

■

2-5 FORMAS CANÓNICAS Y ESTÁNDAR

Minitérminos y maxitérminos

Una variable binaria podría aparecer en su forma normal (x) o en su forma complementada (x'). Considere ahora dos variables binarias x y y que se combinan con una operación AND. Puesto que cada variable podría aparecer en cualquiera de sus dos formas, hay cuatro combinaciones posibles: $x'y'$, $x'y$, xy' y xy. Cada uno de estos cuatro términos AND es un *minitérmi-*

Tabla 2-3
Minitérminos y maxitérminos para tres variables binarias

			Minitérminos		Maxitérminos	
x	**y**	**z**	**Términos**	**Designación**	**Términos**	**Designación**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

no, o *producto estándar*. De manera similar, podemos combinar n variables para formar 2^n minitérminos. Éstos podrían obtenerse con un método similar al que se muestra en la tabla 2-3 para tres variables. Se enumeran los números binarios de 0 a $2^n - 1$ bajo las n variables. Cada minitérmino se obtiene de un término AND de las n variables, poniendo un apóstrofo a cada variable si el bit correspondiente del número binario es un 0 y sin apóstrofo si es un 1. En la tabla también se muestra un símbolo para cada minitérmino, el cual tiene la forma m_j, donde j denota el equivalente decimal del número binario del minitérmino designado.

Asimismo, n variables que forman un término OR, donde cada variable puede tener apóstrofo o no, dan pie a 2^n posibles combinaciones, llamadas *maxitérminos* o *sumas estándar*. En la tabla 2-3 se presentan los ocho maxitérminos de tres variables, junto con su designación simbólica. Podemos obtener de manera similar cualesquier 2^n maxitérminos para n variables. Cada maxitérmino se obtiene de un término OR de las n variables, donde cada variable lleva un apóstrofo si el bit correspondiente del número binario es 1. Cabe señalar que cada maxitérmino es el complemento de su minitérmino correspondiente, y viceversa.

Se puede expresar algebraicamente una función booleana a partir de una tabla de verdad dada formando un minitérmino para cada combinación de las variables que produce un 1 en la función, y formando después el OR de todos esos términos. Por ejemplo, la función f_1 de la

Tabla 2-4
Funciones de tres variables

x	**y**	**z**	**Función f_1**	**Función f_2**
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

tabla 2-4 se obtiene expresando las combinaciones 001, 100 y 111 como $x'y'z$, $xy'z'$ y xyz, respectivamente. Puesto que cada uno de estos minitérminos hace que $f_1 = 1$, tenemos

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

De forma similar, es posible verificar fácilmente que

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

Estos ejemplos ilustran una propiedad importante del álgebra booleana: cualquier función booleana se puede expresar como una suma de minitérminos (donde "suma" se refiere al OR de los términos).

Considere ahora el complemento de una función booleana. Podría leerse de la tabla de verdad formando un minitérmino para cada combinación que produce un 0 en la función, y haciendo después el OR de esos términos. El complemento de f_1 se lee así:

$$f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$$

Si obtenemos el complemento de f_1', obtendremos la función f_1:

$$\begin{aligned} f_1 &= (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z) \\ &= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 \end{aligned}$$

Asimismo, podemos leer la expresión para f_2 de la tabla:

$$\begin{aligned} f_2 &= (x + y + z)(x + y + z')(x + y' + z)(x' + y + z) \\ &= M_0 M_1 M_2 M_4 \end{aligned}$$

Estos ejemplos ilustran una segunda propiedad del álgebra booleana: cualquier función booleana se puede expresar como un producto de maxitérminos (donde "producto" se refiere a hacer el AND de los términos). El procedimiento para obtener el producto de maxitérminos directamente de la tabla de verdad es el siguiente. Se forma un maxitérmino para cada combinación de las variables que produce un 0 en la función y luego se hace el AND de todos esos maxitérminos. Se dice que las funciones booleanas expresadas como suma de minitérminos o producto de maxitérminos están en *forma canónica*.

Suma de minitérminos

Dijimos antes que, para n variables binarias, podemos obtener 2^n minitérminos distintos, y que es posible expresar cualquier función booleana como una suma de minitérminos. Los minitérminos cuya suma define a la función booleana son los que producen los unos de la función en una tabla de verdad. Puesto que la función puede dar 1 o 0 con cada minitérmino, y dado que hay 2^n minitérminos, podemos calcular que el número de funciones que es posible formar con n variables es 2^{2n}. A veces es útil expresar la función booleana en su forma de suma de minitérminos. Si no está ya en esa forma, esto se logra expandiendo primero la expresión a una suma de términos AND. Luego se examina cada término para ver si contiene todas las variables. Si falta una o más variables, se le hace AND con una expresión como $x + x'$, donde x es una de las variables faltantes. El ejemplo que sigue aclarará el procedimiento.

EJEMPLO 2-4

Exprese la función booleana $F = A + B'C$ como suma de minitérminos. La función tiene tres variables, A, B y C. En el primer término, A, faltan dos variables; por tanto:

$$A = A(B + B') = AB + AB'$$

A esta función todavía le falta una variable:

$$\begin{aligned} A &= AB(C + C') + AB'(C + C') \\ &= ABC + ABC' + AB'C + AB'C' \end{aligned}$$

Al segundo término, $B'C$, le falta una variable:

$$B'C = B'C(A + A') = AB'C + A'B'C$$

Al combinar todos los términos, tenemos

$$\begin{aligned} F &= A + B'C \\ &= ABC + ABC' + AB'C + AB'C' + A'B'C \end{aligned}$$

Sin embargo, $AB'C$ aparece dos veces y, según el teorema 1 ($x + x = x$), podemos eliminar uno de ellos. Después de reacomodar los minitérminos en orden ascendente, obtenemos por fin

$$\begin{aligned} F &= A'B'C + AB'C' + AB'C + ABC' + ABC \\ &= m_1 + m_4 + m_5 + m_6 + m_7 \end{aligned}$$

■

Hay ocasiones en que conviene expresar la función booleana, en su forma de suma de minitérminos, con la siguiente notación abreviada:

$$F(A, B, C) = \sum(1, 4, 5, 6, 7)$$

El símbolo de sumatoria, $\sum$, representa el OR de los términos; los números que le siguen son los minitérminos de la función. Las letras entre paréntesis después de F son una lista de las variables en el orden que se usará al convertir un minitérmino en un término AND.

Otro procedimiento para obtener los minitérminos de una función booleana consiste en deducir la tabla de verdad directamente de la expresión algebraica y luego leer los minitérminos de esa tabla. Considere la función booleana del ejemplo 2-4:

$$F = A + B'C$$

La tabla de verdad que se muestra en la tabla 2-5 se deduce directamente de la expresión algebraica enumerando las ocho combinaciones binarias bajo las variables A, B y C e insertan-

Tabla 2-5
Tabla de verdad para $F = A + B'C$

A	*B*	*C*	*F*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

do unos bajo F para las combinaciones en las que $A = 1$ y $BC = 01$. Luego, se lee en la tabla de verdad que los cinco minitérminos de la función son 1, 4, 5, 6 y 7.

Producto de maxitérminos

Cada una de las 2^{2n} funciones de n variables binarias se puede expresar también como un producto de maxitérminos. Para expresar la función booleana como producto de maxitérminos, primero debe ponerse en formato de términos OR. Esto podría hacerse con la ayuda de la ley distributiva, $x + yz = (x + y)(x + z)$. Luego, se hace el OR de cualquier variable faltante x en cada término OR con xx'. El ejemplo que sigue aclarará el procedimiento.

EJEMPLO 2-5

Exprese la función booleana $F = xy + x'z$ en forma de producto de maxitérminos. Primero, se convierte la función en términos OR empleando la ley distributiva:

$$\begin{aligned} F &= xy + x'z = (xy + x')(xy + z) \\ &= (x + x')(y + x')(x + z)(y + z) \\ &= (x' + y)(x + z)(y + z) \end{aligned}$$

La función tiene tres variables, x, y y z. A cada término OR le falta una variable; por tanto:

$$\begin{aligned} x' + y &= x' + y + zz' = (x' + y + z)(x' + y + z') \\ x + z &= x + z + yy' = (x + y + z)(x + y' + z) \\ y + z &= y + z + xx' = (x + y + z)(x' + y + z) \end{aligned}$$

Después de combinar todos los términos y eliminar los que aparecen más de una vez, se obtiene:

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') \\ &= M_0 M_2 M_4 M_5 \end{aligned}$$

Una forma cómoda de expresar esta función es:

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

El símbolo de producto, Π, denota el AND de maxitérminos; los números son los maxitérminos de la función.

Conversión entre formas canónicas

El complemento de una función expresado como la suma de minitérminos es igual a la suma de los minitérminos que faltan en la función original. Ello se debe a que la función original se expresa con los minitérminos que hacen que la función sea igual a 1, mientras que su complemento es 1 para aquellos minitérminos que hacen que la función original sea 0. Por ejemplo, considere la función

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

Su complemento se expresa como

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

Ahora bien, si se determina el complemento de F' por el teorema de DeMorgan, se obtiene F en una forma distinta:

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3 = \Pi(0, 2, 3)$$

Esta última conversión es consecuencia de la definición de minitérminos y maxitérminos, como se muestra en la tabla 2-3. En esa tabla es evidente que se cumple la relación

$$m_j' = M_j$$

Es decir, el maxitérmino con subíndice j es el complemento del minitérmino que lleva ese subíndice, y viceversa.

El último ejemplo ilustra la conversión entre una función expresada como suma de minitérminos y su equivalente en producto de maxitérminos. Un argumento similar demuestra que la conversión entre el producto de maxitérminos y la suma de minitérminos es similar. Ahora plantearemos un procedimiento general de conversión. Para convertir de una forma canónica a otra, intercambiamos los símbolos Σ y Π e incluimos en la lista sólo los números que faltaban en la forma original. Para hallar los términos faltantes, debemos recordar que el número total de minitérminos o maxitérminos es 2^n, donde n es el número de variables binarias en la función.

Es posible convertir una función booleana de una expresión algebraica a un producto de maxitérminos utilizando una tabla de verdad y el procedimiento de conversión canónica. Considere, por ejemplo, la expresión booleana

$$F = xy + x'z$$

Primero, obtenemos la tabla de verdad de la función, la cual se muestra en la tabla 2-6. Los unos bajo F en la tabla se determinan a partir de las combinaciones de las variables en las que $xy = 11$ o $xz = 01$. De la tabla de verdad deducimos que los minitérminos de la función son 1, 3, 6 y 7. La función expresada como suma de minitérminos es

$$F(x, y, z) = \Sigma(1, 3, 6, 7)$$

Puesto que en total hay ocho minitérminos o maxitérminos en una función de tres variables, deducimos que los términos faltantes son 0, 2, 4 y 5. La función expresada como producto de maxitérminos es

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

Ésta es la misma respuesta que obtuvimos en el ejemplo 2-5.

Tabla 2-6
Tabla de verdad para $F = xy + x'z$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Formas estándar

Las dos formas canónicas del álgebra booleana son formas básicas que se obtienen al leer una función de su tabla de verdad, pero casi nunca son las que tienen el número mínimo de literales, porque cada minitérmino o maxitérmino debe contener, por definición, *todas* las variables, sea complementadas o sin complementar.

Otra forma de expresar funciones booleanas es en forma *estándar*. En esta configuración, los términos que forman la función podrían contener una, dos o cualquier número de literales. Hay dos tipos de formas estándar: la suma de productos y el producto de sumas.

La *suma de productos* es una expresión booleana que contiene términos AND, llamados *términos de producto*, con una o más literales cada uno. La *suma* denota el OR de esos términos. Un ejemplo de función expresada como suma de productos es

$$F_1 = y' + xy + x'yz'$$

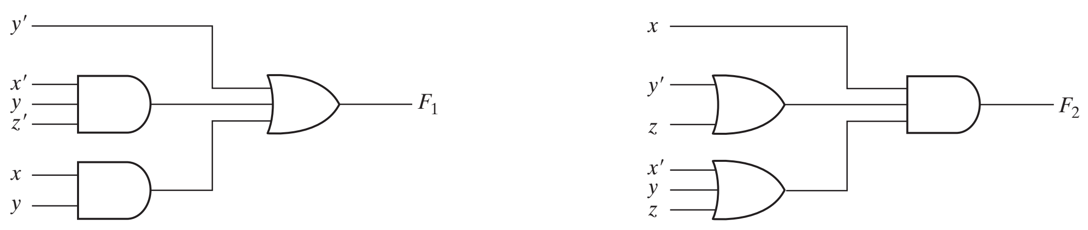
La expresión tiene tres términos de producto con una, dos y tres literales. Su suma es realmente una operación OR.

El diagrama de lógica de una expresión de suma de productos consiste en un grupo de compuertas AND seguidas de una sola compuerta OR. Este patrón de configuración se muestra en la figura 2-3a). Cada término de producto requiere una compuerta AND, salvo los términos que sólo tienen una literal. La suma lógica se forma con una compuerta OR cuyas entradas son las salidas de las compuertas AND y las literales solas. Suponemos que contamos directamente con las variables de entrada en forma de complemento, así que no se han incluido inversores en el diagrama. Esta configuración de circuito se denomina implementación de dos niveles.

Un *producto de sumas* es una expresión booleana que contiene términos OR, llamados *términos de suma*. Cada término puede tener cualquier cantidad de literales. El *producto* denota el AND de esos términos. Un ejemplo de función expresada como producto de sumas es

$$F_2 = x(y' + z)(x' + y + z)$$

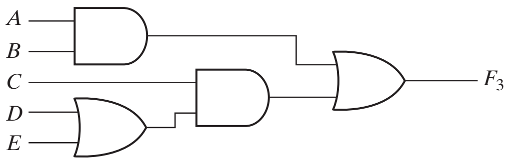
Esta expresión tiene tres términos de suma con una, dos y tres literales. El producto es una operación AND. El uso de las palabras *producto* y *suma* proviene de la similitud entre la operación AND y el producto aritmético (multiplicación), y de la similitud entre la operación OR y la suma aritmética (adición). La estructura de compuertas de la expresión de producto de sumas consiste en un grupo de compuertas OR para los términos de suma (excepto la literal



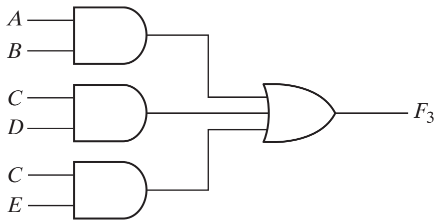
a) Suma de productos

b) Producto de sumas

FIGURA 2-3
Implementación de dos niveles



a) $AB + C(D + E)$



b) $AB + CD + CE$

FIGURA 2-4
Implementación de tres y dos niveles

sola) seguidas de una compuerta AND. Esto se observa en la figura 2-3b). Este tipo estándar de expresión produce una estructura de compuertas de dos niveles.

Las funciones booleanas se pueden expresar en forma no estándar. Por ejemplo, la función

$$F_3 = AB + C(D + E)$$

no es una suma de productos ni un producto de sumas. Su implementación se indica en la figura 2-4a), y requiere dos compuertas AND y dos compuertas OR. Este circuito tiene tres niveles de compuertas, y puede transformarse a una forma estándar utilizando la ley distributiva para eliminar los paréntesis:

$$F_3 = AB + C(D + E) = AB + CD + CE$$

La expresión de suma de productos se implementa en la figura 2-4b). En general, es preferible una implementación de dos niveles porque produce el mínimo de retardo en compuertas cuando la señal se propaga de las entradas a la salida.

2-6 OTRAS OPERACIONES LÓGICAS

Cuando colocamos los operadores binarios AND y OR entre dos variables, x y y, forman dos funciones booleanas, $x \cdot y$ y $x + y$, respectivamente. Se ha señalado ya que hay 2^{2^n} funciones para n variables binarias. En el caso de dos variables, $n = 2$, y el número de posibles funciones booleanas es 16. Por tanto, las funciones AND y OR son sólo dos de un total de 16 posibles funciones que se forman con dos variables binarias. Sería interesante encontrar las otras 14 funciones e investigar sus propiedades.

En la tabla 2-7 se presentan las tablas de verdad para las 16 funciones que se forman con dos variables binarias, x y y. Cada una de las 16 columnas, F_0 a F_{15}, representa una tabla de verdad de una posible función de las dos variables x y y. Observe que las funciones se determinan a partir de las 16 combinaciones binarias que se pueden asignar a F. Las 16 funciones se expresan algebraicamente con funciones booleanas, como se indica en la primera columna de la tabla 2-8. Las expresiones booleanas que se incluyen se han simplificado al número mínimo de literales.

Aunque cada función se puede expresar en términos de los operadores booleanos AND, OR y NOT, no hay motivo para no asignar símbolos de operador especiales que expresen las otras funciones. Dichos símbolos aparecen en la segunda columna de la tabla 2-8. Sin embargo, los