Capítulo 2

Conjuntos ordenados. Retículos y álgebras de Boole.

2.1. Conjuntos ordenados.

En este capítulo vamos a estudiar el concepto de retículo. Hay dos caminos para llegar a esta estructura. Uno es mediante un conjunto con dos operaciones binarias que satisfacen una serie de axiomas. El otro es a partir del concepto de conjunto ordenado. Nosotros aquí hemos adoptado el segundo. Si enriquecemos la estructura de retículo con unos axiomas adicionales obtenemos lo que se conoce como *Álgebra de Boole*. Analizaremos su estructura, y llegaremos a que en el caso finito, las álgebras de Boole tiene una forma muy particular. Estudiaremos el álgebra de Boole de las funciones booleanas, y el teorema de estructura nos conducirá a las formas normales. Terminaremos viendo algunas aplicaciones de las álgebras de Boole al diseño de circuitos lógicos.

Definición 5. *Sea X un conjunto, y $\leq$ una relación binaria en X. Se dice que $\leq$ es una relación de orden si se verifican las siguientes propiedades.*

Reflexiva: $x \leq x$ para todo $x \in X$.

Antisimétrica: Si $x \leq y$ e $y \leq x$ entonces $x = y$.

Transitiva: Si $x \leq y$ e $y \leq z$ entonces $x \leq z$.

Si X es un conjunto en el que tenemos definida una relación de orden $\leq$, se dice que $(X, \leq)$ es un conjunto ordenado (o, si está claro cual es la relación $\leq$ se dice simplemente que X es un conjunto ordenado).

Si $\leq$ es una relación de orden en X que satisface la propiedad adicional de que dados $x, y \in X$ entonces $x \leq y$ ó $y \leq x$, se dice entonces que $\leq$ es una relación de orden total, y que $(X, \leq)$ (o X) es un conjunto totalmente ordenado (en ocasiones, para destacar que $(X, \leq)$ es una relación de orden, pero que no es total se dice que $\leq$ es una relación de orden parcial y que $(X, \leq)$ es un conjunto parcialmente ordenado).

Ejemplo 2.1.1.

1. *El conjunto de los números naturales, con el orden natural ($m \leq n$ si existe $k \in \mathbb{N}$ tal que $n = m+k$) es un conjunto totalmente ordenado. De la misma forma, también lo son $(\mathbb{Z}, \leq)$, $(\mathbb{Q}, \leq)$ y $(\mathbb{R}, \leq)$.*

2. *Dado un conjunto X, entonces $\mathcal{P}(X)$, con el orden dado por la inclusión es un conjunto ordenado. Si X tiene más de un elemento, este orden no es total, pues dados $x, y \in X$ distintos se tiene que $\{x\} \not\subseteq \{y\}$ y $\{y\} \not\subseteq \{x\}$.*

3. *En el conjunto de los números naturales, la relación de divisibilidad es una relación de orden que no es total. Esta relación viene dada por* $a|b$ *si existe* $c \in \mathbb{N}$ *tal que* $b = a \cdot c$ *(compara con la relación de orden natural). Sin embargo, en el conjunto de los números enteros esta relación no es de orden pues no es antisimétrica, ya que* $2|-2$, $-2|2$ *y sin embargo* $2 \neq -2$.

4. *Para cualquier número natural* n *consideramos el conjunto*

$$D(n) = \{m \in \mathbb{N} : m|n\}$$

Entonces $(D(n), |)$ *es un conjunto (parcialmente) ordenado.*

5. *Sea* $(X, \leq)$ *es un conjunto ordenado, e* Y *un subconjunto de* X. *Definimos en* Y *el orden* $x \preceq y$ *si* $x \leq y$ *(vistos como elementos de* X*). Entonces,* $(Y, \preceq)$ *es un conjunto ordenado. De ahora en adelante, el orden en* Y *lo denotaremos igual que en* X.

Un ejemplo de esto último podría ser el caso de los divisores de un número natural n.

Si $(X, \leq)$ *es un conjunto totalmente ordenado, entonces, para cualquier* $Y \subseteq X$ *se tiene que* $(Y, \leq)$ *es un conjunto totalmente ordenado.*

La definición de conjunto ordenado puede hacerse también a partir de la noción de *orden estricto*.

Definición 6. *Sea* X *un conjunto, y* $<$ *una relación binaria en* X. *Se dice que* $<$ *es un orden estricto si se verifican las siguientes propiedades:*

Antirreflexiva Para cualquier $x \in X$ *se tiene que* $x \not< x$.

Transitiva Si $x < y$ *e* $y < z$ *entonces* $x < z$.

Es fácil comprobar que si $\leq$ es una relación de orden en un conjunto X, entonces si definimos

$$x < y \text{ si } x \leq y \text{ y } x \neq y$$

se tiene que $<$ es una relación de orden estricto en X.

De la misma forma, si $<$ es una relación de orden estricto en X entonces la relación siguiente:

$$x \leq y \text{ si } x < y \text{ o } x = y$$

es una relación de orden en X.

Vemos entonces que los conceptos de *relación de orden* y *relación de orden estricto* son equivalentes, pues dada una relación de orden tenemos determinada una relación de orden estricto y viceversa. Además, los caminos para pasar de orden a orden estricto, y de orden estricto a orden, son uno el inverso del otro.

A continuación vamos a construir un grafo (dirigido) asociado a una relación de orden. Aún cuando los grafos serán estudiados con posterioridad, la representación de una relación de orden mediante este grafo ayuda a visualizar mejor el orden dado.

Definición 7. *El diagrama de Hasse de un conjunto ordenado* $(X, \leq)$ *es un grafo dirigido cuyos vértices son los elementos de* X, *y existe un lado de* x *a* y *si* $x < y$ *y no existe* z *tal que* $x < z < y$.

El diagrama de Hasse está definido para cualquier conjunto ordenado. Sin embargo, en general dicho diagrama no permite recuperar el orden. Por ejemplo, en el caso del conjunto $(\mathbb{R}, \leq)$, dado cualquier $x \in \mathbb{R}$ no existe ningún $y \in \mathbb{R}$ que esté conectado a x por algún lado.

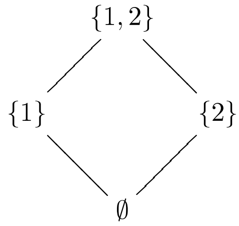
Sin embargo, si el conjunto X es finito, entonces dados $x, y \in X$ se tiene que $x \leq y$ si $x = y$ o existe algún camino que parta de x y termine en y.

Una forma habitual de representar el diagrama de Hasse es dibujar los lados como líneas ascendentes, lo que implica colocar los vértices de forma apropiada.

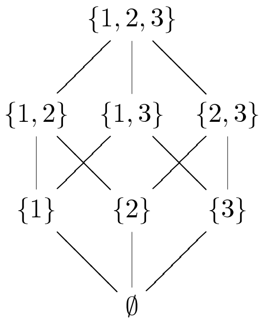
Ejemplo 2.1.2.

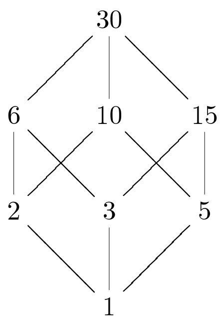
1. *El diagrama de Hasse del conjunto* $(\mathbb{N}, \leq)$ *sería*

2. *Consideramos el conjunto ordenado* $(\mathcal{P}(\{1,2\}, \subseteq)$. *Entonces el diagrama de Hasse sería:*



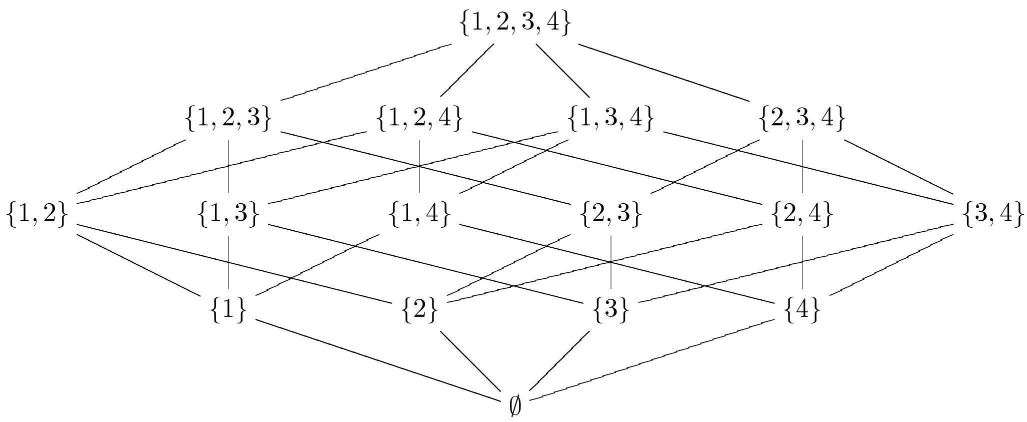
3. *Vamos a representar los diagramas de Hasse de los conjuntos ordenados* $\mathcal{P}(\{1,2,3\})$ *y* $D(30)$.





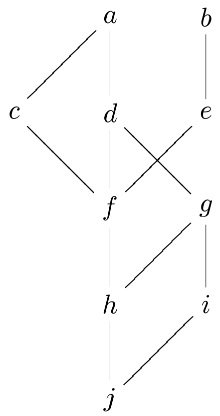
Observa como la estructura de conjunto ordenado es igual en ambos casos.

4. *Vamos a representar el diagrama de Hasse de* $\mathcal{P}(\{1,2,3,4\})$.



Prueba a dibujar el diagrama de Hasse de los divisores de 210 y compáralo con este último.

5. *Si tenemos un grafo dirigido que no contiene caminos cerrados, entonces podemos definir un orden en el conjunto de los vértices.* $x \leq y$ *si* $x = y$ *o existe un camino con inicio* x *y fin* y*. Si en el grafo no hay caminos entre dos vértices adyacentes, entonces el grafo es el diagrama de Hasse de un conjunto ordenado.*



tenemos definido un orden en el conjunto $X = \{a, b, c, d, e, f, g, h, i\}$*. Con este orden se tiene, por ejemplo que,*

$h \leq e$*, pues tenemos un camino* $h - f - e$ *que empieza en* h *y termina en* e*.*

$i \leq a$*, pues el camino* $i - g - d - a$ *empieza en* i *y termina en* a*.*

$i \not\leq e$*, pues ningún camino empieza en* i *y termina en* e*.*

Definición 8. *Sea* $(X, \leq)$ *un conjunto ordenado.*

1. *Un elemento* $x \in X$ *se dice que es maximal, si no existe* $y \in X$ *tal que* $x \leq y$ *y* $x \neq y$*.*

2. *Un elemento* $x \in X$ *se dice que es máximo, si para todo* $y \in X$ *se verifica que* $y \leq x$*.*

De la misma forma se puede definir lo que es un elemento minimal y lo que es un mínimo.

Ejemplo 2.1.3. *En el último conjunto ordenado del ejemplo anterior se tiene que* a *y* b *son elementos maximales, pues no hay ningún elemento que sea mayor que ellos. Sin embargo, el conjunto* X *no tiene máximo.*

El elemento j *es un elemento minimal, y además es mínimo.*

En el conjunto de los divisores de 30 (ver ejemplo anterior) tenemos que 10 *no es un elemento maximal, pues* $10 \leq 30$*. Sí se tiene que* 30 *es un elemento maximal, pues no hay ningún elemento que sea mayor que él. También se tiene que* 30 *es un máximo de ese conjunto.*

En el conjunto $(\mathbb{N}, \leq)$*, el cero es el mínimo y es el único elemento minimal. Este conjunto no tiene máximo ni elementos maximales.*

Si $(X, \leq)$ *es un conjunto ordenado finito, entonces* X *tiene al menos un elemento maximal y un elemento minimal.*

Nótese, que si un conjunto tiene máximo, entonces este es único. Además, en el caso de que tenga máximo, entonces tiene sólo un elemento maximal, que coincide con el máximo.

Idéntica observación vale para mínimo y elemento minimal.

Denotaremos por $\max(X)$ al máximo del conjunto X, en el caso de que exista, y por $\min(X)$ al mínimo.

En el ejemplo que hemos estudiado anteriormente no existe $\max(X)$, mientras que $\min(X) = j$.

Definición 9. *Sea* $(X, \leq)$ *un conjunto ordenado, e* Y *un subconjunto de* X*. Consideramos en* Y *el orden inducido de* X*.*

1. *Un elemento* $x \in X$ *se dice que es* cota superior *de* Y *si* $x \geq y$ *para todo* $y \in Y$*.*

2. *Un elemento $x \in X$ se dice que es* supremo *de Y si es el mínimo del conjunto de las cotas superiores de Y.*

De la misma forma se define lo que es una cota inferior y un ínfimo.

Ejemplo 2.1.4. *Si $X = \{a, b, c, d, e, f, g, h, i, j\}$ con el orden dado anteriormente, e $Y = \{c, d, f, g, h\}$ entonces:*

El conjunto de las cotas superiores de Y es $\{a\}$.

Puesto que este conjunto tiene mínimo, que es a, entonces a es el supremo de Y.

Los elementos c y d son elementos maximales de Y.

El conjunto de las cotas inferiores es $\{h, j\}$.

De éstas, h es el máximo, luego h es el ínfimo de Y.

h es además el único elemento minimal y el mínimo de Y.

Cuando un conjunto tiene supremo éste es único. Podemos entonces hablar de *el supremo de Y*, y lo representaremos mediante $\sup(Y)$.

De la misma forma, denotaremos por $\inf(Y)$ al ínfimo del conjunto Y cuando exista.

Cuando un conjunto tiene máximo, entonces también tiene supremo, y coincide con él. En el último ejemplo vemos como el recíproco no es cierto, pues Y tiene supremo pero no tiene máximo.

Cuando el supremo de un conjunto pertenezca al conjunto, entonces será también el máximo.

Definición 10 (Buen orden)**.** *Sea $(X, \leq)$ un conjunto ordenado. Se dice que $\leq$ es un buen orden si todo subconjunto no vacío de X tiene mínimo. En tal caso, se dice que $(X, \leq)$ (o X) es un conjunto bien ordenado.*

Observación: Todo conjunto bien ordenado es un conjunto totalmente ordenado, pues dados dos elementos $x, y \in X$ el subconjunto $\{x, y\}$ tiene mínimo. Si $\min(\{x, y\}) = x$ entonces $x \leq y$, mientras que si $\min(\{x, y\}) = y$ entonces $y \leq x$.

El recíproco no es cierto. Busca un ejemplo.

Ejemplo 2.1.5. *El conjunto de los números naturales, con el orden usual, es un conjunto bien ordenado, como demostramos en el Teorema 1.1.1.*

Definición 11. *Sean $(X_1, \leq_1)$ y $(X_2, \leq_2)$ dos conjuntos ordenados.*

Se define el orden producto *en $X_1 \times X_2$ como sigue:*

$$(x_1, x_2) \leq_{prod} (y_1, y_2) \text{ si } x_1 \leq_1 y_1 \text{ y } x_2 \leq_2 y_2.$$

Se define el orden lexicográfico *en $X_1 \times X_2$ como sigue:*

$$(x_1, x_2) \leq_{lex} (y_1, y_2) \overset{def}{\Longleftrightarrow} \begin{cases} x_1 <_1 y_1 & \text{ó} \\ x_1 = y_1 \text{ y } x_2 \leq_2 y_2. \end{cases}$$

Claramente, si $(x_1, x_2) \leq_{prod} (y_1, y_2)$ entonces $(x_1, x_2) \leq_{lex} (y_1, y_2)$.

Proposición 2.1.1. *Si* $(X_1, \leq_1)$ *y* $(X_2, \leq_2)$ *son dos conjuntos ordenados, entonces* $(X_1 \times X_2, \leq_{prod})$ *y* $(X_1 \times X_2, \leq_{lex})$ *son conjuntos ordenados.*

Además, si $\leq_1$ *y* $\leq_2$ *son órdenes totales (resp. buenos órdenes) entonces* $\leq_{lex}$ *es un orden total (resp. buen orden).*

Demostración: La demostración de que el orden producto es una relación de orden es fácil, y se deja como ejercicio. Centrémonos pues en el orden lexicográfico.

Notemos en primer lugar que si $(x_1, x_2) \leq_{lex} (y_1, y_2)$ entonces $x_1 \leq_1 y_1$.

Veamos que la relación es de orden.

Reflexiva: Si $(x_1, x_2) \in X_1 \times X_2$ entonces $(x_1, x_2) \leq_{lex} (x_1, x_2)$, pues se da la segunda opción ($x_1 = x_1$ y $x_2 \leq x_2$).

Simétrica: Supongamos que $(x_1, x_2) \leq_{lex} (y_1, y_2)$ y $(y_1, y_2) \leq_{lex} (x_1, x_2)$. Entonces se tiene que $x_1 \leq_1 y_1$ e $y_1 \leq_1 x_1$, de donde $x_1 = y_1$. Deducimos entonces que $x_2 \leq_2 y_2$ e $y_2 \leq_2 x_2$, lo que implica que $x_2 = y_2$.

Transitiva: Supongamos ahora que $(x_1, x_2) \leq_{lex} (y_1, y_2)$ y $(y_1, y_2) \leq_{lex} (z_1, z_2)$. Pueden darse entonces tres opciones(no excluyentes):

- $x_1 <_1 y_1$, en cuyo caso $x_1 <_1 z_1$, luego $(x_1, x_2) \leq_{lex} (z_1, z_2)$.
- $y_1 <_1 z_1$, en cuyo caso $x_1 <_1 z_1$ y concluimos como en la opción anterior.
- $x_1 = y_1$ e $y_1 = z_1$. En tal caso, $x_2 \leq_2 y_2$ e $y_2 \leq_2 z_2$, de donde $x_1 = z_1$ y $x_2 \leq_2 z_2$, es decir, $(x_1, x_2) \leq_{lex} (z_1, z_2)$.

Supongamos ahora que $\leq_1$ y $\leq_2$ son órdenes totales. Sean $(x_1, x_2), (y_1, y_2) \in X_1 \times X_2$. Aquí pueden darse tres opciones (mutuamente excluyentes):

- $x_1 <_1 y_1$. En tal caso $(x_1, x_2) \leq_{lex} (y_1, y_2)$.
- $y_1 <_1 x_1$. En este caso $(y_1, y_2) \leq_{lex} (x_1, x_2)$.
- $x_1 = y_1$. Entonces dependiendo de que $x_2 \leq_2 y_2$ o $y_2 \leq_2 x_2$ se tendrá que $(x_1, x_2) \leq_{lex} (y_1, y_2)$ o que $(y_1, y_2) \leq_{lex} (x_1, x_2)$.

Por último, supongamos que $\leq_1$ y $\leq_2$ son buenos órdenes, y sea $Y \subseteq X_1 \times X_2$ un subconjunto no vacío.

Nos quedamos con el conjunto de todas las primeras coordenadas de los elementos de A, es decir, tomamos

$$Y_1 = \{x_1 \in X_1 : (x_1, x_2) \in A \text{ para algún } x_2 \in X_2\}.$$

Sea $a = \text{mín}(Y_1)$. Tomamos entonces $Y_2 = \{x_2 \in X_2 : (a, x_2) \in A\}$. Como $Y_2 \neq \emptyset$, tiene mínimo. Sea éste b. Entonces $(a, b) = \text{mín}(A)$. ■

Observación: Si tenemos n conjuntos ordenados $X_1, X_2, \ldots, X_n$, podemos definir recursivamente el orden producto y el orden lexicográfico en $X_1 \times X_2 \times \cdots \times X_n$.

Supuesto definido el orden producto $\leq_{prod}$ en $X_1 \times \cdots \times X_{n-1}$ se define en $X_1 \times \cdots \times X_n$:

$$(x_1, \ldots x_{n-1}, x_n) \leq_{prod} (y_1, \ldots, y_{n-1}, y_n) \; si \; (x_1, \ldots, x_{n-1}) \leq_{prod} (y_1, \ldots, y_{n-1}) \text{ y } x_n \leq y_n,$$

es decir, definimos el orden producto en $(X_1 \times \cdots \times X_{n-1}) \times X_n$.

Supuesto definido el orden lexicográfico $\leq_{lex}$ en $X_1 \times \cdots \times X_{n-1}$ se define en $X_1 \times \cdots \times X_n$:

$$(x_1, \ldots x_{n-1}, x_n) \leq_{lex} (y_1, \ldots, y_{n-1}, y_n) \overset{\text{def}}{\Longleftrightarrow} \begin{cases} (x_1, \ldots, x_{n-1}) <_{lex} (y_1, \ldots, y_{n-1}) & \text{ó} \\ (x_1, \ldots, x_{n-1}) = (y_1, \ldots, y_{n-1}) \; y \; x_n \leq y_n. \end{cases}$$

Sea el conjunto

$$\mathcal{A} = \{ \ , a, b, c, d, e, f, g, h, i, j, l, l, m, n, \tilde{n}, o, p, q, r, s, t, u, v, w, x, y, z\},$$

es decir, las 27 letras del alfabeto junto con el espacio en blanco.

Claramente, $\mathcal{A}$ tiene un orden total de todos conocido.

Supongamos que n es el número de letras de la palabra más larga de la lengua española. Entonces, cada palabra puede representarse como un elemento de $\mathcal{A}^n$ (poniendo tantos espacios al final como sea necesario).

Cuando ordenamos las palabras, tal y como vienen en un diccionario, nos fijamos en la primera letra, y es la que nos da el orden. Cuando ésta coincide, pasamos a la segunda, y es ésta entonces la que nos da el orden. De coincidir también, nos fijamos en la tercera, y así sucesivamente. Es decir, las palabras de la lengua están ordenadas siguiendo el orden lexicográfico.

Ejemplo 2.1.6.

Consideramos en $\mathbb{N}\times\mathbb{N}$ *los órdenes producto* ($\leq$) *y lexicográfico* $\leq_{lex}$ *deducidos a partir del orden usual en* $\mathbb{N}$. *Entonces:*

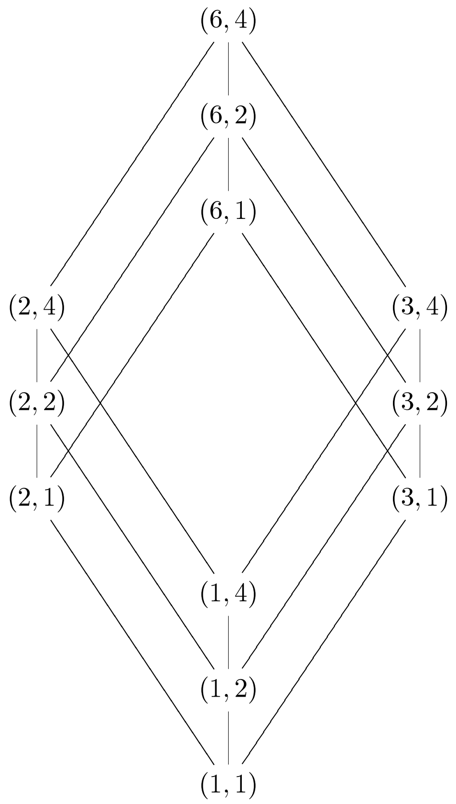
Los elementos $(0,n),(1,n-1),\ldots,(n-1,1),(n,0)$ *están ordenados según el orden lexicográfico, mientras que con el orden producto ninguna pareja de ellos es comparable.*

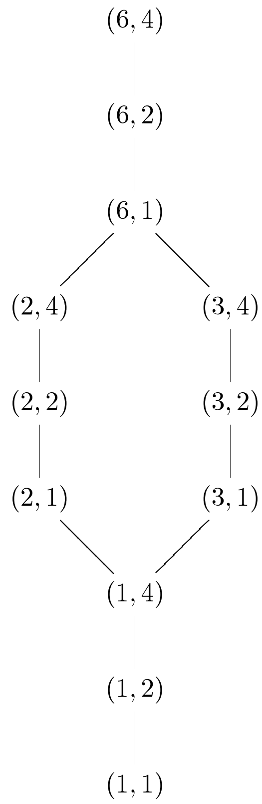
Se puede ver entonces que la propiedad de ser orden total o buen orden no se mantiene al tomar el orden producto.

Si $X=\{(0,n),(1,n-1),\ldots,(n-1,1),(n,0)\}$ *entonces:*

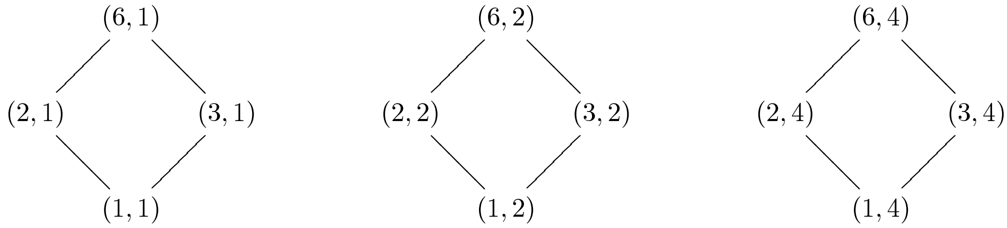
- *El conjunto de cotas inferiores con respecto al orden lexicográfico es* $\{(0,0),(0,1),\ldots,(0,n)\}$, *mientras que con respecto al orden producto tiene una única cota inferior, que es* $(0,0)$.
- *El ínfimo, respecto al orden lexicográfico es* $(0,n)$, *que es también el mínimo. Con respecto al orden producto es* $(0,0)$, *y no tiene mínimo.*
- *Con respecto al orden lexicográfico tiene un elemento minimal, que es* $(0,n)$ *y un elemento maximal, que es* $(n,0)$. *Con respecto al orden producto, todos los elementos son maximales y minimales.*

Sean ahora los conjuntos ordenados $(D(6),|)$ *y* $(D(4),|)$. *Entonces los diagramas de Hasse de* $D(6)\times D(4)$ *con el orden producto y el orden lexicográfico son respectivamente:*

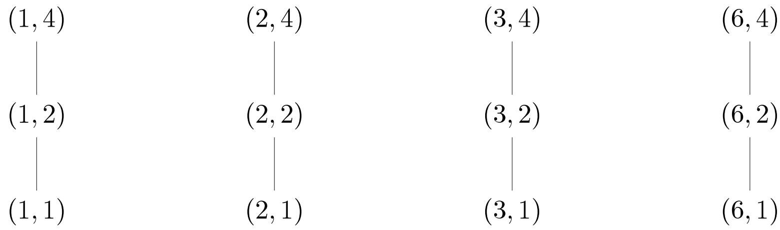




Nótese como el diagrama de Hasse de $D(6) \times D(4)$ con el orden producto consiste en "pegar" tres diagramas como el de $D(6)$

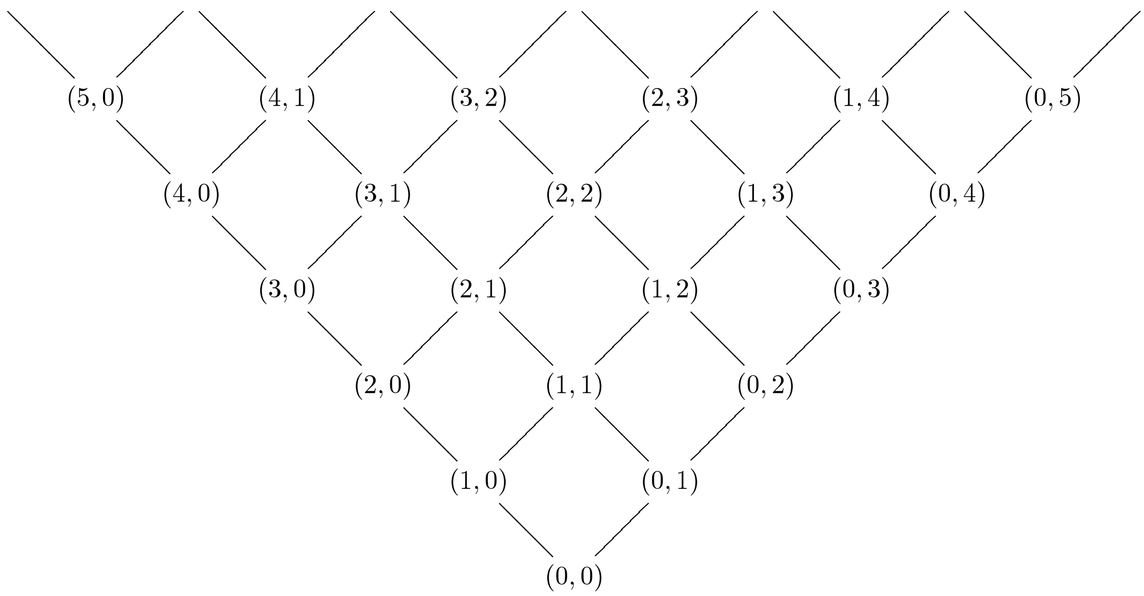


y cuatro diagramas como el de $D(4)$



mientras que el diagrama de Hasse de $D(6) \times D(4)$ con el orden lexicográfico tiene la "misma forma" que el de $D(6)$, salvo que en cada vértice tenemos un diagrama de $D(4)$.

El diagrama de Hasse de $(\mathbb{N}^2, \leq_{prod})$ sería como sigue:



2.2. Retículos.

Definición 12. *Un retículo es un conjunto ordenado, $(L, \leq)$ en el que cualquier conjunto finito tiene supremo e ínfimo.*

Si $(L, \leq)$ es un retículo y $x, y \in L$, denotaremos por $x \vee y$ al supremo del conjunto $\{x, y\}$ y por $x \wedge y$ al ínfimo del conjunto $\{x, y\}$.

Nótese que $x \vee y$ está definido por la propiedad:

$$x \leq x \vee y;\ y \leq x \vee y \qquad (x \leq z \text{ e } y \leq z) \Longrightarrow x \vee y \leq z$$

La primera parte dice que $x \vee y$ es una cota superior del conjunto $\{x, y\}$, mientras que la segunda dice que es la menor de las cotas superiores.

Proposición 2.2.1. *Si $(L, \leq)$ es un retículo, las operaciones $\vee$ y $\wedge$ satisfacen las siguienes propiedades:*

$$\begin{array}{rl}
\textit{Conmutativa} & \begin{cases} x \vee y = y \vee x \\ x \wedge y = y \wedge x. \end{cases} \\
\textit{Asociativa} & \begin{cases} x \vee (y \vee z) = (x \vee y) \vee z \\ x \wedge (y \wedge z) = (x \wedge y) \wedge z. \end{cases} \\
\textit{Absorción} & \begin{cases} x \vee (x \wedge y) = x \\ x \wedge (x \vee y) = x. \end{cases} \\
\textit{Idempotencia} & \begin{cases} x \vee x = x \\ x \wedge x = x. \end{cases}
\end{array}$$

Demostración: La demostración de la propiedad conmutativa, así como la de idempotencia es inmediata. Para demostrar la propiedad asociativa basta comprobar que tanto $x \vee (y \vee z)$ como $(x \vee y) \vee z$ representa el supremo del conjunto $\{x, y, z\}$, y lo mismo para el ínfimo. Veamos que $\sup(\{x, y, z\}) = x \vee (y \vee z)$.

Es claro que $x \leq x \vee (y \vee z)$, $y \leq x \vee (y \vee z)$ y $z \leq x \vee (y \vee z)$. Por otra parte,

$$\left.\begin{array}{l} x \leq u \\ \left.\begin{array}{l} y \leq u \\ z \leq u \end{array}\right\} \Longrightarrow y \vee z \leq u \end{array}\right\} \Longrightarrow x \vee (y \vee z) \leq u.$$

Por tanto, $x \vee (y \vee z)$ es el supremo del conjunto $\{x, y, z\}$.

En cuanto a la absorción, la primera se deduce fácilmente del hecho de que $x \wedge y \leq x$ y la segunda de que $x \leq x \vee y$. $\blacksquare$

Ejemplo 2.2.1.

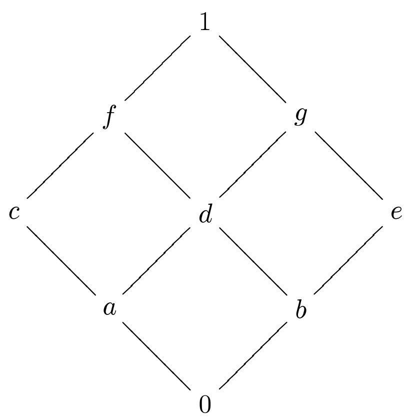
1. *Si X es un conjunto totalmente ordenado, entonces X es un retículo. Dados $x, y \in X$ se tiene que $x \vee y = \text{máx}(\{x, y\})$ mientras que $x \wedge y = \text{mín}(\{x, y\})$.*

2. *El conjunto ordenado $(\mathbb{N}, |)$ es un retículo. En este caso se tiene que $x \vee y = mcm(x, y)$ mientras que $x \wedge y = mcd(x, y)$. De la misma forma, si $n \in \mathbb{N}$ entonces $D(n)$, con el orden dado por la divisibilidad es un retículo. Supremo e ínfimo vienen dado por el mínimo común múltiplo y el máximo común divisor respectivamente.*

3. *Si X es un conjunto, entonces $\mathcal{P}(X)$ es un retículo. En este caso supremo e ínfimo vienen dados por la unión y la intersección respectivamente; es decir, $A \vee B = A \cup B$ y $A \wedge B = A \cap B$.*

4. *Si V es un K-espacio vectorial, el conjunto de los subespacios vectoriales de V es un retículo, con el orden dado por la inclusión. Aquí, dado dos subespacios vectoriales V_1 y V_2 se tiene que $V_1 \vee V_2 = V_1 + V_2$ mientras que $V_1 \wedge V_2 = V_1 \cap V_2$.*

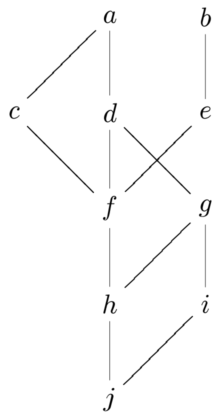
5. *El conjunto ordenado cuyo diagrama de Hasse es*



es un retículo.

Se tiene, por ejemplo: $c \vee d = f$, $c \wedge d = a$, $b \vee c = f$, $b \wedge c = 0$, $c \vee e = 1$, $c \wedge e = 0$.

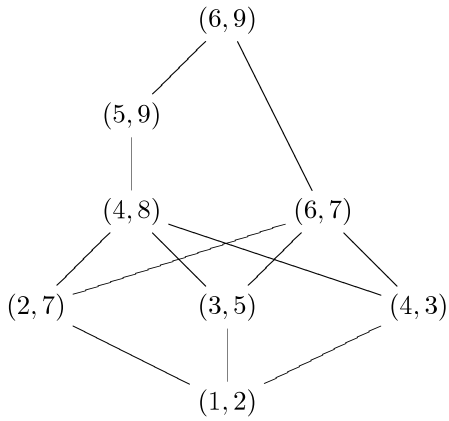
6. *El conjunto ordenado cuyo diagrama de Hasse es*



no es un retículo, pues por ejemplo, no existe el supremo del conjunto $\{a, e\}$. *Sin embargo, el conjunto* $\{f, i\}$ *sí tiene supremo (*d*) e ínfimo (*j*).*

7. *Dado el conjunto* $A = \{(1,2);\ (4,3);\ (3,5);\ (2,7);\ (4,8);\ (6,9);\ (6,7); (5,9)\} \subseteq \mathbb{N}^2$, *consideramos en* A *el orden inducido del orden producto en* $\mathbb{N}^2$.

El diagrama de Hasse de A *sería:*



Este conjunto no es un retículo. Por ejemplo, tomamos $x = (2,7)$ *e* $y = (3,5)$. *El conjunto de las cotas superiores de* x *e* y *es* $\{(4,8);\ (6,7);\ (5,9);\ (6,9)\}$. *Y ese conjunto no tiene mínimo. Por tanto, no existe la menor de las cotas superiores, luego no existe el supremo de* x *e* y.

Nótese que si $(L, \leq)$ es un retículo, entonces dados $x, y \in L$ se verifica que $x \leq y$ si, y sólo si, $x \vee y = y$, o si queremos, $x \leq y$ si, y sólo si, $x \wedge y = x$. Es decir, podemos recuperar el orden dentro del retículo a partir del conocimiento de las operaciones supremo o ínfimo.

La siguiente proposición nos da condiciones suficientes para que dos operaciones definidas en un conjunto puedan ser el supremo y el ínfimo de alguna relación de orden en ese conjunto.

Proposición 2.2.2. *Sea* L *un conjunto en el que tenemos definidas dos operaciones* $\vee$ *y* $\wedge$ *que satisfacen las propiedades conmutativa, asociativa, idempotencia y de absorción. Supongamos que en* L *definimos la relación*

$$x \leq y \qquad \text{si } x \vee y = y.$$

Entonces, $(L, \leq)$ *es un retículo donde las operaciones supremo e ínfimo vienen dadas por* $\vee$ *y* $\wedge$ *respectivamente.*

Demostración:

1. Veamos en primer lugar que $(L, \leq)$ es un conjunto ordenado. Para esto, comprobemos que la relación $\leq$ es reflexiva, antisimétrica y transitiva.

   Reflexiva. Puesto que $x \vee x = x$ se tiene que $x \leq x$ para cualquier $x \in L$.

   Antisimétrica. Supongamos que $x \leq y$ e $y \leq x$. Esto implica que $x \vee y = y$ y que $y \vee x = x$. Puesto que $\vee$ es conmutativa deducimos que $x = y (= x \vee y)$.

   Transitiva. Supongamos ahora que $x \leq y$ y que $y \leq z$, es decir, $x \vee y = y$ e $y \vee z = z$. Entonces:

   $$x \vee z = x \vee (y \vee z) = (x \vee y) \vee z = y \vee z = z$$

   luego $x \leq z$.

2. Comprobemos ahora que dados $x, y \in L$ se verifica que $\sup(\{x, y\}) = x \vee y$.

   Puesto que $x \vee (x \vee y) = (x \vee x) \vee y = x \vee y$ se tiene que $x \leq x \vee y$. De la misma forma se comprueba que $y \leq x \vee y$.

   Si $x \leq u$ e $y \leq u$ (es decir, $x \vee u = u$ e $y \vee u = u$). Entonces:

   $$(x \vee y) \vee u = x \vee (y \vee u) = x \vee u = u,$$

   de donde se deduce que $x \vee y \leq u$.

3. Por último, veamos que $\inf(\{x, y\}) = x \wedge y$.

   $(x \wedge y) \vee x = x \vee (x \wedge y) = x$ luego $x \wedge y \leq x$.

   De la misma forma se comprueba que $x \wedge y \leq y$.

   Si $u \leq x$ y $u \leq y$ (es decir, $u \vee x = x$ y $u \vee y = y$) se tiene que:

   $$u \wedge x = u \wedge (u \vee x) = u \qquad u \wedge y = u \wedge (u \vee y) = u,$$

   $$u \wedge (x \wedge y) = (u \wedge x) \wedge y = u \wedge y = u,$$

   $$u \vee (x \wedge y) = (u \wedge (x \wedge y)) \vee (x \wedge y) = (x \wedge y) \vee ((x \wedge y) \wedge u) = x \wedge y,$$

   luego $u \leq x \wedge y$.

■

Nótese que se tiene que $x \vee y = y$ si, y sólo si, $x \wedge y = x$, luego podría haberse hecho la demostración definiendo la relación

$$x \leq y \quad \text{si } x \wedge y = x.$$

Nótese también que la propiedad de idempotencia se puede deducir a partir de la de absorción, pues

$$x \vee x = x \vee [x \wedge (x \vee x)] = x,$$

luego podemos demostrar la proposición anterior partiendo de que las operaciones $\vee$ y $\wedge$ satisfacen las propiedades asociativa, conmutativa y de absorción.

Esta proposición permite definir un retículo, bien dando la relación de orden, bien dando las operaciones $\vee$ y $\wedge$.

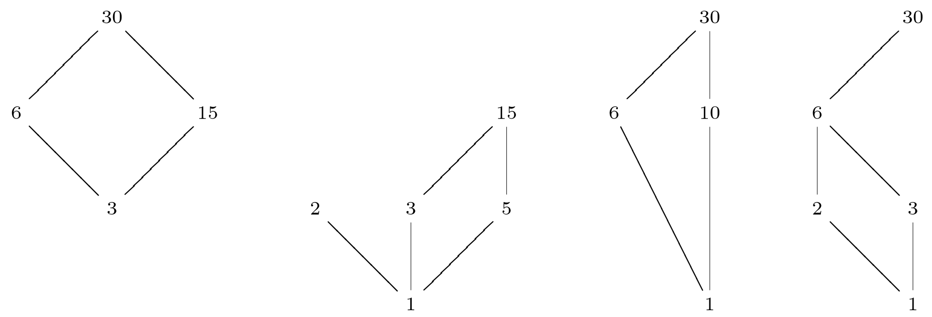
Si $(L, \leq)$ es un retículo y L tiene máximo, denotaremos a éste por 1, mientras que si tiene mínimo lo denotaremos por 0. Se tiene entonces, $x \vee 1 = 1$, $x \wedge 1 = x$, $x \vee 0 = x$ y $x \wedge 0 = 0$.

Un retículo finito siempre tiene máximo y mínimo. Si el retículo es infinito, puede tenerlo o no. Así, por ejemplo, $(\mathbb{N}, \leq)$ tiene mínimo pero no tiene máximo; $(\mathbb{Z}, \leq)$ no tiene ni mínimo ni máximo. El retículo $(\mathbb{N}, |)$ es infinito y tiene máximo y mínimo. En este caso, el máximo es 0 mientras que el mínimo es 1.

Definición 13. *Sea $(L, \leq)$ un retículo, y $L' \subseteq L$ un subconjunto de L. Entonces L' es un subrretículo si para cualesquiera $x, y \in L'$ se verifica que $x \vee y \in L'$ y $x \wedge y \in L'$.*

Ejemplo 2.2.2. *Consideramos el retículo $D(30)$.*

Sean $L_1 = \{3, 6, 15, 30\}$, $L_2 = \{1, 2, 3, 5, 15\}$, $L_3 = \{1, 6, 10, 30\}$ y $L_4 = \{1, 2, 3, 6, 30\}$. Sus diagramas de Hasse son:



L_1 L_2 L_3 L_4

Entonces L_1 y L_4 son subrretículos de $D(30)$, mientras que L_2 y L_3 no lo son. L_2 no es subrretículo porque el supremo de 2 y 3 es 6, que no pertenece a L_2. L_3 no es subrretículo porque el ínfimo de 6 y 10 vale 2, que no pertenece a L_3. Nótese que L_3, con el orden que hereda de $D(30)$, es un retículo, pero no es subrretículo de L_3.

Definición 14. *Sea L un retículo. Se dice que L es distributivo si para cualesquiera $x, y, z \in L$ se verifica que*

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \quad \text{y} \quad x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$$

En general, si L es un retículo se tiene que $x \vee (y \wedge z) \leq (x \vee y) \wedge (x \vee z)$.

$$\left.\begin{array}{l} \left.\begin{array}{lcl} x & \leq & x \vee y \\ x & \leq & x \vee z \end{array}\right\} \Longrightarrow \quad x \leq (x \vee y) \wedge (x \vee z) \\ \\ \left.\begin{array}{l} y \wedge z \leq x \vee y \\ y \wedge z \leq x \vee z \end{array}\right\} \Longrightarrow y \wedge z \leq (x \vee y) \wedge (x \vee z) \end{array}\right\} \Longrightarrow x \vee (y \wedge z) \leq (x \vee y) \wedge (x \vee z),$$

y de la misma forma se tiene que $(x \wedge y) \vee (x \wedge z) \leq x \wedge (y \vee z)$. Por tanto, se tiene que un retículo es distributivo si $(x \vee y) \wedge (x \vee z) \leq x \vee (y \wedge z)$ y $(x \wedge (y \vee z) \leq x \wedge y) \vee (x \wedge z)$.

Por otra parte, si $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ para cualesquiera $x, y, z \in L$ se tiene que

$$\begin{array}{rcll} (x \wedge y) \vee (x \wedge z) & = & [(x \wedge y) \vee x] \wedge [(x \wedge y) \vee z] & \text{propiedad distributiva} \\ & = & [x \vee (x \wedge y)] \wedge [z \vee (x \wedge y)] & \text{pues } \vee \text{ es conmutativa} \\ & = & [(x \vee x) \wedge (x \vee y)] \wedge [(z \vee x) \wedge (z \vee y)] & \text{propiedad distributiva} \\ & = & (x \vee x) \wedge (x \vee y) \wedge (x \vee z) \wedge (y \vee z) & \text{propiedad asociativa y conmutativa} \\ & = & [x \wedge (x \vee y) \wedge (x \vee z)] \wedge (y \vee z) & \text{idempotencia y propiedad asociativa} \\ & = & x \wedge (y \vee z) & \text{Absorción} \end{array}$$

mientras que si $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ para cualesquiera $x, y, z \in L$ entonces se verifica que $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ también para cualesquiera $x, y, z \in L$.

Es decir, basta con que se dé una de las dos posibles propiedades distributivas para que se dé la otra.

Ejemplo 2.2.3.

1. *Si L es un conjunto totalmente ordenado, entonces L es un retículo distributivo. Basta comprobar que para cualesquiera $x, y, z \in L$ se verifica que*

$$máx\{x, mín\{y, z\}\} = mín\{máx\{x, y\}, máx\{x, z\}\},$$

lo cual puede hacerse fácilmente comprobando que se da la igualdad en cualquiera de los seis casos siguientes:

$$x \leq y \leq z; \quad x \leq z \leq y; \quad y \leq x \leq z; \quad y \leq z \leq x; \quad z \leq x \leq y; \quad z \leq y \leq x,$$

y puesto que en la igualdad el papel que juegan y y z es el mismo, bastaría con comprobarlo en los casos

$$x \leq y \leq z; \qquad y \leq x \leq z; \qquad y \leq z \leq x.$$

2. *El retículo $(\mathbb{N}, |)$ es un retículo distributivo. Basta ver que en este caso, el cálculo del supremo y el ínfimo se reduce al cálculo del máximo y el mínimo de los exponentes, y entonces reducirse al caso anterior.*

   Por el mismo motivo, para cada número natural $n \in \mathbb{N}$ el retículo $D(n)$ es distributivo.

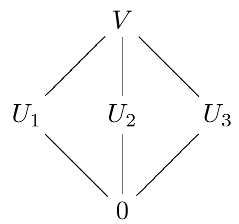
3. *Si X es un conjunto, entonces $(\mathcal{P}(X), \subseteq)$ es un retículo distributivo, pues la unión y la intersección de conjuntos son distributivas la una con respecto de la otra.*

4. *Si V es un K-espacio de dimensión mayor que 1, entonces el retículo de los subespacios vectoriales de V es un retículo que no es distributivo.*

   Como ejemplo, sea $K = \mathbb{Z}_2$ y $V = \mathbb{Z}_2^2$. Entonces V tiene 5 subespacios vectoriales que son:

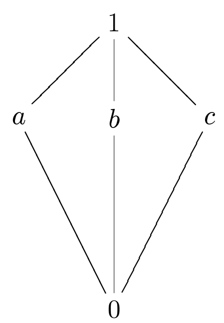
$$V; \quad U_1 = \{(0,0),(1,0)\}; \quad U_2 = \{(0,0),(0,1)\}; \quad U_3 = \{(0,0),(1,1)\}; \quad 0.$$

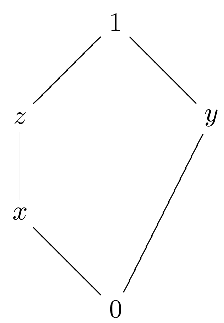
   y se tiene que $U_2 \cap U_3 = 0$, luego $U_1 + (U_2 \cap U_3) = U_1$, mientras que $(U_1 + U_2) \cap (U_1 + U_3) = V \cap V = V$. El diagrama de Hasse de este retículo es:



   En general, si V es un K-espacio de dimensión mayor o igual que 2, y u, v son dos vectores linealmente independientes, consideramos $U_1 = L\{u\}$, $U_2 = L\{v\}$ y $U_3 = L\{u+v\}$ y se verifica que $U_1 + (U_2 \cap U_3) = U_1$, mientras que $(U_1 + U_2) \cap (U_1 + U_3) = L\{u, v\}$.

5. *Consideramos los siguientes retículos:*





   denominados respectivamente diamante *y* pentágono. *En el ejemplo anterior hemos visto que el diamente no es distributivo. En cuanto al pentágono, se tiene que*

$$x \vee (y \wedge z) = x \vee 0 = x, \qquad (x \vee y) \wedge (x \vee z) = 1 \wedge z = z.$$

luego tampoco es distributivo.

En general, se tiene que un retículo es distributivo si no contiene como subrretículos ni al pentágono ni al diamante. En el apartado anterior hemos visto como el retículo de subespacios vectoriales de un espacio vectorial tiene al diamante como subrretículo.

Proposición 2.2.3. *Sea L un retículo distributivo, y sen $x, y, z \in L$ tales que $x \vee y = x \vee z$ y $x \wedge y = x \wedge z$. Entonces $y = z$.*

Demostración: Se tiene que

$$y = y \vee (x \wedge y) = y \vee (x \wedge z) = (y \vee x) \wedge (y \vee z) = (z \vee x) \wedge (z \vee y) = z \vee (x \wedge y) = z \vee (x \wedge z) = z.$$

■

Ejemplo 2.2.4. *En el diamante se tiene que $a \vee b = a \vee c = 1$, y $a \wedge b = a \wedge c = 0$, y sin embargo, $b \neq c$.*
En el pentágono, $y \vee x = y \vee z = 1$ e $y \wedge x = y \wedge z = 0$, y sin embargo, $x \neq z$.

Definición 15. *Sea L un retículo que tiene máximo y mínimo (a los que denotaremos por 1 y 0 respectivamente), y $x \in L$. Se dice que $y \in L$ es un complemento de x si $x \vee y = 1$ y $x \wedge y = 0$.*
Un reticulo en el que todo elemento tiene complemento se dice complementado.

Obviamente, si y es un complemento de x entonces x es un complemento de y.
Por otra parte, si L es un retículo distributivo y x un elemento de L que tiene complemento, entonces el complemento es único (ver Proposicion 2.2.3).
Si L es un retículo distributivo y x es un elemento que tiene complemento, denotaremos por x' o $\overline{x}$ al único complemento de x.

Ejemplo 2.2.5.

1. *Si L tiene máximo (1) y mínimo (0), entonces 0 es un complemento de 1.*
2. *El retículo $(\mathcal{P}(X), \subseteq)$ es un retículo complementado. Dado $A \in \mathcal{P}(X)$ se verifica que $A \cup (X \setminus A) = X$ y $A \cap (X \setminus A) = \emptyset$. Por ser un retículo distributivo, el complemento de cada elemento es único.*
3. *El pentágono y el diamente son retículos complementados. Vemos sin embargo, que los complementos de algunos elementos no son únicos.*

   Así, en el diamante, tanto b como c son complementos de a; tanto a como c son complementos de b y tanto a como b son complementos de c.

   En el pentágono, tanto x como z son complementos de y. Sin embargo, x tiene un único complemento, que es y, al igual que z.
4. *Si L es un conjunto totalmente ordenado con más de dos elementos, entonces es un retículo distributivo, pero no es complementado.*
5. *Si V es un K-espacio vectorial de dimensión finita (esto último no es necesario) entonces el retículo de los subespacios vectoriales de V es un retículo complementado.*

   Para ver esto, tomamos U un subespacio vectorial de V. Supongamos que $\mathcal{B}_U = \{u_1, \ldots, u_m\}$ es una base de U. Esta base puede ser ampliada hasta una base de V. Si dicha base ampliada es $\mathcal{B} = \{u_1, \ldots, u_m, u_{m+1}, \ldots, u_n\}$ entonces el subespacio generado por $\{u_{m+1}, \ldots, u_n\}$ es un complemento de U.

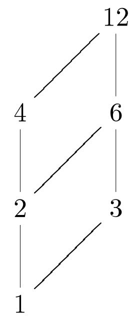
   Puesto que en general hay muchas formas de completar una base de V a partir de una base de U, el subespacio U puede tener muchos complementos.

   Así, si tomamos $U = \mathbb{R}^2$ y $U = L[(1,0)]$ (es decir, el eje OX) entonces cualquier recta que pase por el origen distinta del eje OX es un complemento de U.

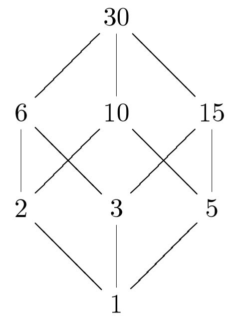
6. *Dado un número natural $D(n)$, el retículo $D(n)$ no tiene por qué ser un retículo complementado. Por ejemplo, $D(4)$ no es complementado (es un conjunto totalmente ordenado con 3 elementos), mientras que $D(6)$ sí lo es.*

   Se pide, determinar qué elementos de $D(n)$ tienen complemento, y a partir de ahí, determinar para qué valores de n es $D(n)$ un retículo complementado.

   Así, por ejemplo, en $D(12)$ tienen complemento $1, 3, 4, 12$ mientras que no tienen $2, 6$. En $D(30)$ todos los elementos tienen complemento.



$D(12)$



$D(30)$

Proposición 2.2.4. *Sean $(L_1, \leq)$ y $(L_2, \leq)$ dos conjuntos ordenados. Consideramos en $L_1 \times L_2$ el orden producto. Entonces:*

- *Si L_1 y L_2 son retículos, también lo es $L_1 \times L_2$. Las operaciones supremo e ínfimo en $L_1 \times L_2$ vienen dadas por*

$$(x_1, x_2) \vee (y_1, y_2) = (x_1 \vee y_1, x_2 \vee y_2) \qquad (x_1, x_2) \wedge (y_1, y_2) = (x_1 \wedge y_1, x_2 \wedge y_2)$$

- *Si L_1 y L_2 son retículos distributivos, también lo es $L_1 \times L_2$.*
- *Si L_1 y L_2 son retículos complementados, también lo es $L_1 \times L_2$.*

2.3. Álgebras de Boole

2.3.1. Generalidades sobre álgebras de Boole

Definición 16. *Un* álgebra de Boole *es un retículo distributivo y complementado.*

Ejemplo 2.3.1.

1. *Dado un conjunto X, el conjunto $\mathcal{P}(X)$, con el orden dado por la inclusión es un álgebra de Boole.*
2. *$D(6)$, o $D(30)$ son álgebras de Boole. No es álgebra de Boole $D(4)$ o $D(12)$.*

Al igual que los retículos se pueden definir sin mencionar el orden, sino únicamente las operaciones supremo e ínfimo, con las respectivas propiedades, un álgebra de Boole puede definirse también a partir de las operaciones $\vee$ y $\wedge$.

Definición 17 (Segunda definición de álgebra de Boole)**.** *Sea B un conjunto. Supongamos que en B tenemos definidas dos operaciones, $\vee$ y $\wedge$ tales que:*

1. $x \vee (y \vee z) = (x \vee y) \vee z, \qquad x \wedge (y \wedge z) = (x \wedge y) \wedge z.$
2. $x \vee y = y \vee x, \qquad x \wedge y = y \wedge z.$

3. *$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z), \qquad x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.*

4. *$x \vee (x \wedge y) = x, \qquad x \wedge (x \vee y) = x$.*

5. *Existen $0, 1 \in B$ tales que $x \vee 0 = x, \quad x \wedge 0 = 0, \quad x \vee 1 = 1, \quad x \wedge 1 = x$.*

6. *Para cada $x \in B$ existe $\overline{x} \in B$ tal que $x \vee \overline{x} = 1$ y $x \wedge \overline{x} = 0$.*

Es fácil comprobar que las definiciones 16 y 17 son equivalentes.

Proposición 2.3.1 (Leyes de De Morgan)**.** *Sea B un álgebra de Boole, y $x, y \in B$. Entonces:*

$$\overline{x \vee y} = \overline{x} \wedge \overline{y}, \qquad \overline{x \wedge y} = \overline{x} \vee \overline{y}.$$

Demostración: Se verifica que:

$$(x \vee y) \vee (\overline{x} \wedge \overline{y}) = [(x \vee y) \vee \overline{x}] \wedge [(x \vee y) \vee \overline{y}] = (x \vee \overline{x} \vee y) \wedge (x \vee y \vee \overline{y}) = (1 \vee y) \wedge (x \vee 1) = 1 \wedge 1 = 1,$$

$$(x \vee y) \wedge (\overline{x} \wedge \overline{y}) = [x \wedge (\overline{x} \wedge \overline{y})] \vee [y \wedge (\overline{x} \wedge \overline{y})] = (0 \wedge \overline{y}) \vee (\overline{x} \wedge 0) = 0 \vee 0 = 0.$$

■

Ejemplo 2.3.2.

1. *Consideremos el conjunto $\mathbb{Z}_2$. En él, consideramos las operaciones*

$$x \wedge y = xy, \qquad x \vee y = x + y + xy.$$

   Entonces $\mathbb{Z}_2$, con estas operaciones es un álgebra de Boole. De hecho, es el álgebra de Boole más simple (a excepción de un álgebra de Boole con un elemento). Representaremos a este álgebra de Boole como $\mathbb{B}$.

   Nótese que este álgebra de Boole se corresponde con el orden $0 \leq 1$.

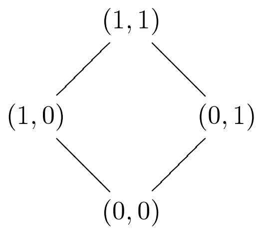
2. *Puesto que el producto de álgebras de Boole es un álgebra de Boole, tenemos, para cada número natural n el álgebra de Boole $\mathbb{B}^n$ que tiene 2^n elementos. En este caso, las operaciones del álgebra de Boole vienen dadas por:*

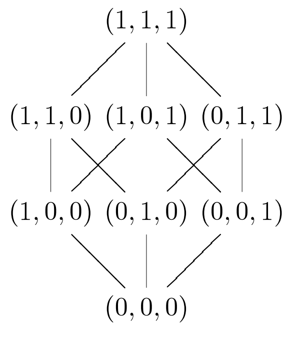
$$(x_1, x_2, \ldots, x_n) \vee (y_1, y_2, \ldots, y_n) = (x_1 \vee y_1, x_2 \vee y_2, \ldots, x_n \vee y_n),$$

$$(x_1, x_2, \ldots, x_n) \wedge (y_1, y_2, \ldots, y_n) = (x_1 \wedge y_1, x_2 \wedge y_2, \ldots, x_n \wedge y_n),$$

$$\overline{(x_1, x_2, \ldots x_n)} = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}).$$

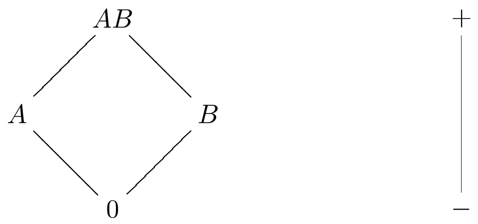
   Veamos los diagramas de Hasse de $\mathbb{B}^2$ y $\mathbb{B}^3$.



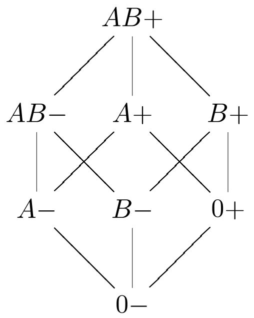


   Podemos comparar las estructuras de álgebra de Boole de $\mathbb{B}^2$ y $\mathbb{B}^3$ con las de $\mathcal{P}(\{a, b\})$ y $\mathcal{P}(\{a, b, c\})$.

3. Consideramos las álgebras de Boole siguientes:



que como podemos ver tienen una estructura semejante a $\mathbb{B}^2$ y $\mathbb{B}$ respectivamente. Su producto, tendrá entonces la misma estructura que $\mathbb{B}^3$. El diagrama de Hasse de dicho álgebra sería



y vemos que los elementos que la forman son los ocho grupos sanguíneos. En este caso, ser menor o igual *significa* puede donar. *Así, el grupo $0-$ es el donante universal, mientras que el grupo $AB+$ es el receptor universal.*

Definición 18. *Sea B un álgebra de Boole y $x \in B$. Se dice que x es un átomo si x es un elemento minimal de $B \setminus \{0\}$.*

Nota: Si $(X, \leq)$ es un conjunto ordenado que tiene mínimo (que llamaremos 0), podemos también definir los átomos de X como los elementos minimales del conjunto $X \setminus \{0\}$.

Ejemplo 2.3.3. *Si X es un conjunto, los átomos del álgebra de Boole $\mathcal{P}(X)$ son los subconjuntos unitarios.*

Los átomos del álgebra de Boole $\mathbb{B}^n$ son aquellos que tienen todas las coordenadas nulas salvo una.

En el álgebra de Boole $D(30)$ los átomos son los divisores primos de 30.

Teorema 2.3.1. *Sea B un álgebra de Boole finita, y $x \in B \setminus \{0\}$. Entonces, x se expresa de forma única como supremo de átomos.*

Antes de demostrar el teorema, veamos el siguiente lema:

Lema 2.3.1. *Sea B un álgebra de Boole finita y $x \in B \setminus \{0\}$. Entonces existe $a \in B$, átomo, y tal que $a \leq x$.*

Demostración: Basta tomar el conjunto $A_x = \{y \in B : 0 < y \leq x\}$, que es distinto del vacío (pues x es un elemento suyo). Se tiene que un elemento minimal de A_x (que existe por ser A_x finito) es un átomo de B. ■

Dado cualquier elemento $x \in B \setminus \{0\}$, denotaremos por $\mathcal{A}_x$ al conjunto de todos los átomos de B que son menores o iguales que x.

Demostración:(teorema 2.3.1)

Supongamos que $\mathcal{A}_x = \{a_1, a_2, \ldots, a_m\}$. Sea entonces $z = a_1 \vee a_2 \vee \ldots \vee a_m$. Comprobemos que $z = x$.

Puesto que $a_i \leq x$ se tiene que $z \leq x$. Supongamos que $z \neq x$.

Consideramos $\overline{z}$. Se tiene entonces que $1 = z \vee \overline{z} \leq x \vee \overline{z}$ de donde $x \vee \overline{z} = 1$. Por tanto, $x \wedge \overline{z} \neq 0$ (si valiera 0 tendríamos que $\overline{z} = \overline{x}$, lo que implicaría que $z = x$).

Sea a un átomo menor o igual que $x \wedge \overline{z}$. Entonces, $a \leq x$, luego $a = a_i$ para algún i. Supongamos que $a = a_1$. En ese caso, se tiene que:

$$0 = \overline{z} \wedge z = \overline{z} \wedge (a_1 \vee \cdots \vee a_m) \geq a \wedge (a_1 \vee \cdots \vee a_m) = (a \wedge a_1) \vee (a \wedge a_2) \vee \cdots \vee (a \wedge a_m) = a_1,$$

lo cual no es posible.

Deducimos por tanto que $z = x$, es decir, x se expresa como supremo de átomos.

Supongamos ahora que podemos expresar x como supremo de átomos de la forma $x = b_1 \vee \cdots \vee b_k$. Entonces:

$$b_i = b_i \wedge x = b_i \wedge (a_1 \vee \cdots \vee a_m) = (b_i \wedge a_1) \vee (b_i \wedge a_2) \vee \cdots \vee (b_i \wedge a_m),$$

y puesto que el ínfimo de dos átomos vale cero salvo que los dos átomos coincidan deducimos que $b_i = a_j$ para algún j. Por tanto, se tiene que

$$\{b_1, \ldots, b_k\} \subseteq \{a_1, \ldots, a_m\}.$$

De forma análoga se demuestra la otra inclusión. ∎

Este teorema nos dice que si B es un álgebra de Boole finita, y $X = \{a_1, \ldots, a_n\}$ son sus átomos (es decir, $X = \mathcal{A}_1$) entonces los elementos de B son:

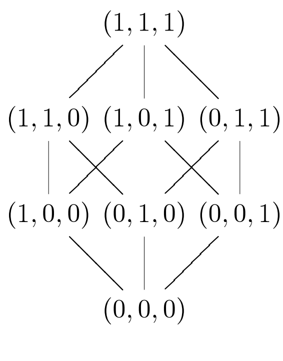
$$B = \left\{ \bigvee_{x \in A} x : \ A \in \mathcal{P}(X) \right\},$$

donde se ha empleado la notación $0 = \bigvee_{x \in \emptyset} x$.

Vemos entonces que B tiene tantos elementos como $\mathcal{P}(X)$. Por tanto, el número de elementos de B es 2^n, donde n es el número de átomos. Es más, tenemos que las álgebras de Boole B, $\mathbb{B}^n$ y $\mathcal{P}(X)$ con $X = \{1, 2, \ldots, n\}$ son isomorfas.

Ejemplo 2.3.4.

1. *Consideramos el álgebra de Boole $\mathbb{B}^3$.*

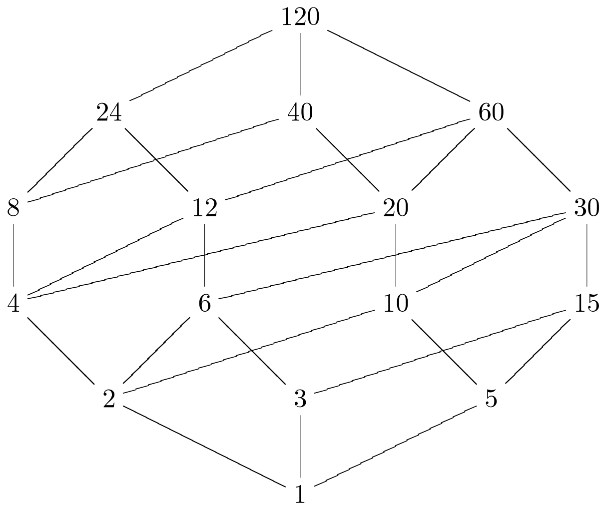


Vemos que los átomos son $(1, 0, 0)$, $(0, 1, 0)$ *y* $(0, 0, 1)$. *Si tomamos cualquier elemento distinto de* $(0, 0, 0)$, *por ejemplo,* $(1, 0, 1)$ *podemos comprobar fácilmente que se puede expresar como supremo de átomos. En este caso,* $(1, 0, 1) = (1, 0, 0) \vee (0, 0, 1)$.

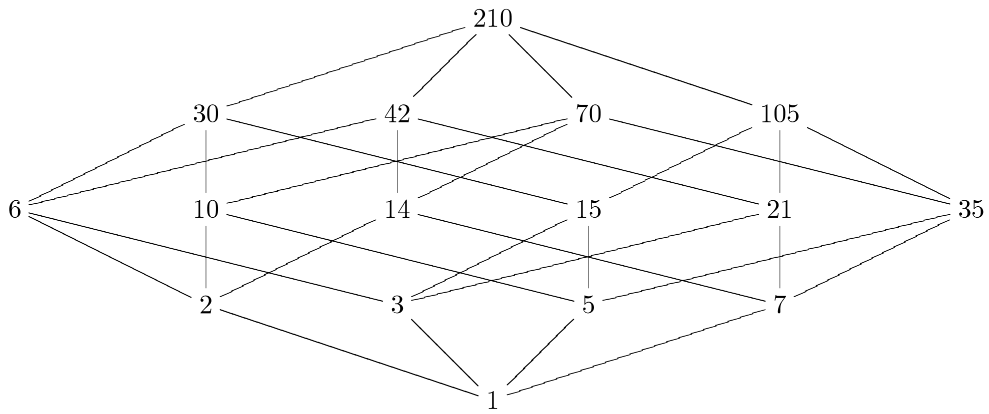
Con cualquier otro elemento podríamos hacer lo mismo.

2. *Sean ahora los conjuntos ordenados* $D(120)$ *y* $D(210)$*, cuyos diagramas de Hasse son:*

$D(120)$



$D(210)$



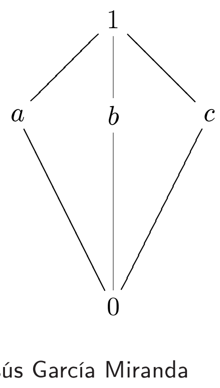
Los átomos del primer conjunto ordenado serían los elementos $2, 3, 5$*. Si tomamos por ejemplo el elemento* 30 *vemos que podemos ponerlo como supremo de átomos, pues* $30 = 2 \vee 3 \vee 5$*. Pero, por ejemplo, si tomamos* $x = 20$ *vemos que no podemos ponerlo como supremo de átomos. Los átomos que son menores que 20 son 2 y 5, y como podemos comprobar,* $2 \vee 5 = 10 \neq 20$.

Por tanto, aquí hay elementos que no se pueden poner como supremo de átomos. Esto nos dice que $D(120)$ *no es un álgebra de Boole.*

Los átomos del segundo conjunto son $2, 3, 5, 7$*. Podemos tomar cualquier elemento distinto de 1, y comprobar que puede expresarse como supremo de átomos. Por ejemplo:* $42 = 2 \vee 3 \vee 7$*,* $35 = 5 \vee 7$*,* $105 = 3 \vee 5 \vee 7$*.*

El conjunto de los divisores de 210 es un álgebra de Boole.

3. *Consideramos el diamante, que sabemos que no es un álgebra de Boole, pues es distributivo.*



Los átomos aquí serían a, b, c. La situación aquí es que todo elemento (salvo 0) se expresa como supremo de átomos. Pero la forma no es única, pues $1 = a \vee b = a \vee c = b \vee c = a \vee b \vee c$.

2.3.2. Funciones y expresiones booleanas

Definición 19. *Una función booleana con n variables es una aplicación $f : \mathbb{B}^n \to \mathbb{B}$.*

Denotaremos por $\mathcal{F}_n$ al conjunto de las funciones booleanas con n variables. Es decir:

$$\mathcal{F}_n = \{f : \mathbb{B}^n \to \mathbb{B} \quad f \textit{ es aplicación}\}.$$

Ejemplo 2.3.5.

1. *La aplicación $f : \mathbb{B} \to \mathbb{B}$ dada por $f(0) = 1$; $f(1) = 0$ es una función booleana en 1 variable (es decir, un elemento de $\mathcal{F}_1$. Esta aplicación responde a la expresión $f(x) = \overline{x}$.*

2. *Sea $f : \mathbb{B}^2 \to \mathbb{B}$ la aplicación $f(x, y) = x \vee y$. Entonces f es una aplicación booleana en 2 variables ($f \in \mathcal{F}_2$). Esta aplicación, elemento a elemento es:*

$$(0,0) \mapsto 0, \qquad (1,0) \mapsto 1, \qquad (0,1) \mapsto 1, \qquad (1,1) \mapsto 1.$$

Definición 20. *Dadas $f, g : \mathbb{B}^n \to \mathbb{B}$ se dice que $f \leq g$ si $f(x_1, x_2, \ldots, x_n) \leq g(x_1, x_2, \ldots x_n)$ para todo $(x_1, x_2, \ldots, x_n) \in \mathbb{B}^n$.*

Es fácil comprobar que esta relación convierte a $\mathcal{F}_n$ en un álgebra de Boole. Las operaciones *supremo* e *ínfimo*, así como el complementario vienen dador por

$$f \vee g(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \vee g(x_1, \ldots x_n),$$

$$f \wedge g(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \wedge g(x_1, \ldots x_n),$$

$$\overline{f}(x_1, \ldots, x_n) = \overline{f(x_1, \ldots, x_n)},$$

mientras que el máximo y el mínimo son las aplicaciones constantes 1 y 0 respectivamente.

Los átomos de este álgebra de Boole son las aplicaciones que valen 1 en un elemento de $\mathbb{B}^n$, y 0 en el resto. Puesto que en $\mathbb{B}^n$ hay 2^n elementos, tenemos que $\mathcal{F}_n$ tiene 2^n átomos, lo que nos dice que $\mathcal{F}_n$ tiene 2^{2^n} elementos.

Ejemplo 2.3.6.

1. *El álgebra $\mathcal{F}_1$ tiene $2^2 = 4$ elementos. Éstos son:*

$$\begin{array}{cccc} 0 \mapsto 0 & 0 \mapsto 0 & 0 \mapsto 1 & 0 \mapsto 1 \\ 1 \mapsto 0 & 1 \mapsto 1 & 1 \mapsto 0 & 1 \mapsto 1. \end{array}$$

Los átomos son las aplicaciones segunda y tercera.

2. *El álgebra $\mathcal{F}_2$ tiene 4 átomos, y por tanto 16 elementos. Los átomos son:*

$$\begin{array}{cccc} (0,0) \mapsto 1 & (0,0) \mapsto 0 & (0,0) \mapsto 0 & (0,0) \mapsto 0 \\ (1,0) \mapsto 0 & (1,0) \mapsto 1 & (1,0) \mapsto 0 & (1,0) \mapsto 0 \\ (0,1) \mapsto 0 & (0,1) \mapsto 0 & (0,1) \mapsto 1 & (0,1) \mapsto 0 \\ (1,1) \mapsto 0 & (1,1) \mapsto 0 & (1,1) \mapsto 0 & (1,1) \mapsto 1. \end{array}$$

2.3.3. Expresiones booleanas

Definición 21. *Sea S un conjunto. Se definen las expresiones booleanas sobre el conjunto S de forma recursiva como sigue:*

1. *Si $x \in S \cup \{0,1\}$ entonces x es una expresión booleana.*
2. *Si e_1, e_2 son expresiones booleanas, entonces también lo son $e_1 \vee e_2$, $e_1 \wedge e_2$ y $\overline{e_1}$.*

A las expresiones booleanas que sean elementos de S, o complementos suyos, los denominaremos literales.

Ejemplo 2.3.7. *Si $S = \{x, y, z\}$ son expresiones booleanas x, $x \vee z$, $\overline{x \wedge \overline{y}}$, 1.*
Son literales, x, $\overline{z}$, z.

A la hora de representar las expresiones booleanas, emplearemos la notación $x\,y$ o $x{\cdot}y$ para la expresión $x \wedge y$, mientras que usaremos la notación $x + y$ para la expresión $x \vee y$.
Así, la expresión booleana $x \vee (y \wedge \overline{z})$ la representaremos como $x + (y\,\overline{z})$.

Supongamos que tenemos un conjunto S con n elementos, es decir, $S = \{x_1, x_2, \ldots, x_n\}$. A cada elemento de S le vamos a asignar un elemento de $\mathcal{F}_n$. Concretamente, al elemento x_i le asignamos la función $x_i : \mathbb{B}^n \to \mathbb{B}$ dada por $x_i(a_1, \ldots, a_i, \ldots, a_n) = a_i$. De esta forma, a cada expresión booleana sobre el conjunto S le podemos hacer corresponder una función $\mathbb{B}^n \to \mathbb{B}$.
Por ejemplo, si $S = \{x, y, z\}$ y consideramos la expresión booleana $x \vee (\overline{y} \wedge z)$, le corresponde la función booleana

$$\begin{array}{ll}
(0,0,0) \mapsto 0 \vee (1 \wedge 0) = 0, & (0,0,1) \mapsto 0 \vee (1 \wedge 1) = 1, \\
(0,1,0) \mapsto 0 \vee (0 \wedge 0) = 0, & (0,1,1) \mapsto 0 \vee (0 \wedge 1) = 0, \\
(1,0,0) \mapsto 1 \vee (1 \wedge 0) = 1, & (1,0,1) \mapsto 1 \vee (1 \wedge 1) = 1, \\
(1,1,0) \mapsto 1 \vee (0 \wedge 0) = 1, & (1,1,1) \mapsto 1 \vee (0 \wedge 1) = 1.
\end{array}$$

Puesto que cada expresión booleana determina una función booleana, podremos referirnos a las funciones mencionando las expresiones que las representan. Así, la función que acabamos de ver podría definirse como $f(x, y, z) = x \vee (\overline{y} \wedge z)$. Ahora, para calcular la imagen de un elemento de $\mathbb{B}^3$ basta sustituir en la expresión booleana x, y y z por los valores en los que queremos evaluar, y efectuar las operaciones en el álgebra de Boole $\mathbb{B}$. Por ejemplo

$$f(0,0,1) = 0 \vee (\overline{0} \wedge 1) = 0 \vee (1 \wedge 1) = 0 \vee 1 = 1.$$

Nótese que en el Ejemplo 2.3.5 ya se ha empleado esta forma de definir una función booleana.
Si ahora quisiéramos emplear la notación introducida anteriormente, la función f adoptaría la forma $f(x, y, z) = x + (\overline{y}\,z)$.
A la hora de emplear esta notación hemos de tener cuidado en no confundir con las operaciones suma y producto hecho en $\mathbb{Z}_2$. En relación al producto no hay problema, pues vimos como la operación $\wedge$ se corresponde con el producto en $\mathbb{Z}_2$. Sin embargo, en $\mathbb{Z}_2$ se tiene que $x \vee y = x + y + x\,y$, lo cual hace que la operación $+$ difiera de la operación $\vee$, pues $1 + 1 = 0$ mientras que $1 \vee 1 = 1$. Para el resto de parejas, ambas operaciones coinciden ($0 + 0 = 0 \vee 0$; $0 + 1 = 0 \vee 1$; $1 + 0 = 1 \vee 0$). El contexto nos aclarará en cada caso si al emplear el símbolo $+$ nos estamos refiriendo a la suma (en $\mathbb{Z}_2$) o al supremo (en $\mathbb{B}$).
Por ejemplo, si decimos *sea f la función booleana dada por* $f(x, y, z) = x\,y + y\,\overline{z}$ está claro que nos referimos al supremo. En tal caso, se tiene que

$$f(0,1,1) = 0 \cdot 1 + 1 \cdot 0 = 0 + 0 = 0 \qquad f(1,1,0) = 1 \cdot 1 + 1 \cdot 1 = 1 + 1 = 1$$

Definición 22. *Dos expresiones booleanas son equivalentes si las correspondientes funciones booleanas son iguales. Si e_1 e e_2 son expresiones booleanas equivalentes emplearemos el símbolo $e_1 = e_2$.*

Ejemplo 2.3.8. *Las expresiones booleanas* $\overline{x}\,\overline{y}$ *y* $\overline{x+y}$ *son equivalentes. También lo son las expresiones* $x+y+\overline{x}\,\overline{y}$ *y* 1.

A continuación vamos a dar una tabla de expresiones equivalentes.

Proposición 2.3.2. *Sean* e_1, e_2 *y* e_3 *tres expresiones booleanas en* n *variables. Entonces:*

1.	$e_1+(e_2+e_3)=(e_1+e_2)+e_3$	$e_1\cdot(e_2\cdot e_3)=(e_1\cdot e_2)\cdot e_3$
2.	$e_1+e_2=e_2+e_1$	$e_1\cdot e_2=e_2\cdot e_1$
3.	$e_1+e_1=e_1$	$e_1\cdot e_1=e_1$
4.	$e_1\cdot(e_2+e_3)=e_1\cdot e_2+e_1\cdot e_3$	$e_1+(e_2\cdot e_3)=(e_1+e_2)\cdot(e_1+e_3)$
5.	$\overline{e_1+e_2}=\overline{e_1}\cdot\overline{e_2}$	$\overline{e_1\cdot e_2}=\overline{e_1}+\overline{e_2}$
6.	$e_1+\overline{e_1}=1$	$e_1\cdot\overline{e_1}=0$
7.	$e_1+1=1$	$e_1\cdot 0=0$
8.	$e_1+0=e_1$	$e_1\cdot 1=e_1$
9.	$\overline{1}=0$	$\overline{0}=1$

Definición 23. *Sea* $S=\{x_1,x_2,\ldots,x_n\}$. *Un* minterm *en* n *variables es el producto de* n *literales, cada uno con una variable diferente.*

Ejemplo 2.3.9. *Si* $S=\{x,y,z\}$, *entonces son minterm* $x\,y\,z$, $x\,\overline{y}\,\overline{z}$, $\overline{x}\,y\,z$. *No son minterm* $x\,y$, $x\,y\,\overline{y}$ *ni* $x\,z\,x$.

Lema 2.3.2. *Sea* m *un minterm en* n *variables. Entonces* m *determina una función booleana* $f:\mathbb{B}^n\to\mathbb{B}$ *que vale* 1 *en un elemento de* $\mathbb{B}^n$ *y* 0 *en el resto.*

Ejemplo 2.3.10. *Sea* $f:\mathbb{B}^2\to\mathbb{B}$ *la función booleana dada por* $f(x,y)=x\,\overline{y}$. *Claramente* $x\,\overline{y}$ *es un minterm. Se tiene que* $f(1,0)=1$, *mientras que* $f(0,0)=f(0,1)=f(1,1)=0$.

Corolario 2.3.1. *Los minterm son los átomos del álgebra* $\mathcal{F}_n$.

Corolario 2.3.2. *Toda función booleana se expresa de forma única (salvo el orden) como suma (supremo) de minterm.*

La expresión de una función booleana como suma de minterm recibe el nombre de *forma normal disyuntiva.* Para hallar la forma normal disyuntiva de una función booleana podemos emplear dos métodos.

El primero consiste en evaluar la función en todos los elementos de $\mathbb{B}^n$, y observar en cuales de ellos toma el valor 1. Cada uno de esos elementos se corresponde con un minterm.

El segundo consiste en, a partir de una expresión booleana que nos defina a f, utilizar las equivalencias dadas en la proposición 2.3.2 para transformar la expresión en una suma de minterm.

Ejemplo 2.3.11. *Vamos a expresar como supremo de minterm la función booleana dada por* $f(x,y)=x+y$.

1. *Si queremos emplear el primer método, evaluamos la función en los cuatro elementos de* $\mathbb{B}^2$. *Nos queda:*

$$f(0,0)=0+0=0 \qquad f(0,1)=0+1=1 \qquad f(1,0)=1+0=1 \qquad f(1,1)=1+1=1$$

El elemento $(0,1)$ *se corresponde con el minterm* $\overline{x}\,y$, *el* $(1,0)$ *con* $x\,\overline{y}$ *mientras que* $(1,1)$ *se corresponde con* $x\,y$. *Por tanto tenemos que* $f(x,y)=\overline{x}\,y+x\,\overline{y}+x\,y$.

2. Empleamos ahora el segundo método. En este caso

$$
\begin{aligned}
f(x,y) &= x+y \\
&= x\cdot 1 + 1\cdot y && \textit{(equivalencias 8 y 2)} \\
&= x(y+\overline{y}) + (x+\overline{x})y && \textit{(equivalencia 6)} \\
&= x\,y + x\,\overline{y} + x\,y + \overline{x}\,y && \textit{(equivalencias 4 y 2)} \\
&= x\,y + x\,y + x\,\overline{y} + \overline{x}\,y && \textit{(equivalencia 2)} \\
&= x\,y + x\,\overline{y} + \overline{x}\,y && \textit{(equivalencia 3).}
\end{aligned}
$$

Cada elemento de $\mathbb{B}^n$ es una secuencia de n dígitos *ceros* o *unos*. Es por tanto, la expresión en binario de un número entre 0 y 2^n-1. Por otra parte, a cada elemento de $\mathbb{B}^n$ le corresponde un minterm (aquél para el que toma el valor 1). Por tanto, cada minterm está determinado por un número comprendido entre 0 y 2^n-1. Denotaremos por *el minterm a*, donde $0 \leq a \leq 2^{n-1}$, y lo representaremos como $m(a)$ o m_a, al minterm determinado por el número a siguiendo el criterio anterior.

Por ejemplo, el minterm $x\,y\,\overline{z}\,\overline{t}$ toma el valor 1 en $(1,1,0,0)$. Puesto que $12 = (1100)_2$ tenemos que $x\,y\,\overline{z}\,\overline{t}$ es el minterm 12, o dicho de otra forma, $x\,y\,\overline{z}\,\overline{t} = m_{12} = m(12)$.

Ejemplo 2.3.12. *La función booleana del ejemplo anterior $f(x,y) = x+y$ hemos visto que se expresa como suma de minterm de la forma $f(x,y) = x\,y + x\,\overline{y} + \overline{x}\,y$. Empleando la notación recién introducida nos quedaría $f(x,y) = m_3 + m_2 + m_1$, o si preferimos $f(x,y) = m_1 + m_2 + m_3$.*

También se suele emplear la notación $f(x,y) = \sum m(1,2,3)$.

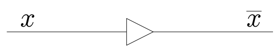
De la misma forma que toda función booleana se expresa de forma única como suma de minterm, se puede probar que toda función booleana se expresa de forma única como producto de maxterm. Una expresión de esta forma se denomina *forma canónica conjuntiva.*

Un maxtern es una suma de n literales. Se corresponde con una función booleana que vale 1 en todos los elementos de $\mathbb{B}^n$ salvo en uno, en el que vale 0. Este elemento determina al maxterm.

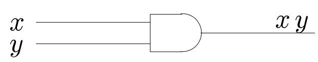
2.3.4. Puertas lógicas

En este apartado utilizaremos las funciones booleanas para el diseño de circuitos lógicos. Los elementos básicos de estos circuitos se llaman *puertas lógicas.* Aquí emplearemos tres tipos de puertas lógicas, cada una correspondiente a una operación booleana, y las combinaremos para diseñar circuitos que realicen una serie de tareas. Las puertas básicas a emplear son:

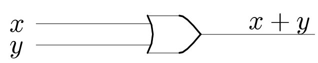
La puerta **NOT** que tiene como entrada el valor de una variable booleana y produce como salida el complementario de dicho valor. Para una puerta NOT emplearemos el siguiente símbolo.



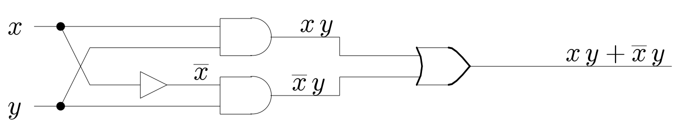
La puerta **AND** que tiene como entrada el valor de dos o más variables booleanas, y como salida el producto de éstas. Las entradas se muestran a la izquierda y la salida a la derecha. Emplearemos el siguiente símbolo para esta puerta.



La puerta **OR** tiene como entrada el valor de dos o más variables booleanas, y como salida la suma de éstas. La representaremos mediante el siguiente símbolo.



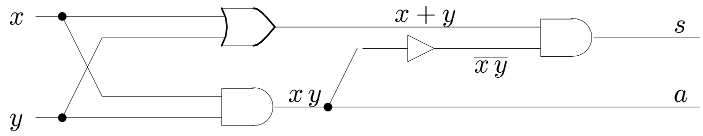
Ejemplo 2.3.13. *Vamos a diseñar un circuito con dos entradas y que produzca la salida $x\,y + \overline{x}\,y$.*



Nótese que puesto que $x\,y + \overline{x}\,y = (x+\overline{x})y = 1\cdot y = y$ podría haberse diseñado un circuito mucho más simple que tuviera el mismo efecto.

A continuación vamos a diseñar un circuito que, introducidos dos números en binario nos devuelve su suma.

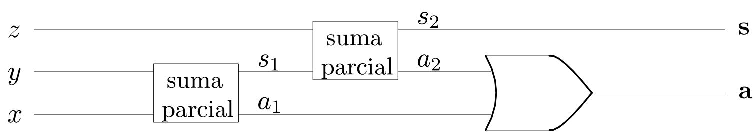
Para esto, comenzamos en primer lugar diseñando un circuito que, dados dos dígitos binarios nos devuelva la suma. Puesto que los posibles resultados de la suma son 0, 1 y 10 necesitamos un circuito que tenga dos salidas. Denotaremos el bit de la derecha como s (suma) y el de la izquierda como a (acarreo). Se tiene entonces que $s = \overline{x}\,y + x\,\overline{y} = (x+y)\overline{x\,y}$ mientras que $a = x\,y$. Un circuito podría ser entonces:



Denominaremos a este circuito como *suma parcial*.

Construyamos ahora un circuito que nos sume dos dígitos binarios más el posible acarreo de una suma anterior. Podemos ver fácilmente que esto es equivalente a sumar tres dígitos binarios x, y y z. El resultado será, como antes una salida doble. A las dos salidas las denotaremos de la misma forma que en el caso anterior: s y a.

Para obtener la salida s, obtenemos la suma de x e y, y al resultado le sumamos z. Puede verse entonces que un circuito que nos da la suma de tres dígitos sería:

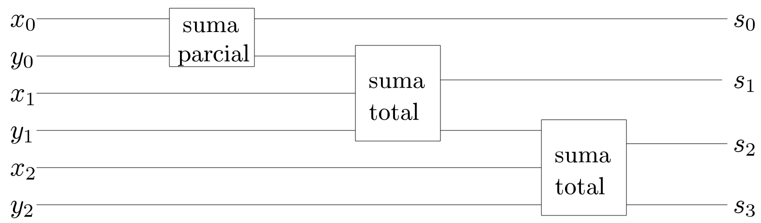


x	y	z	s_1	a_1	s_2	a_2	$\mathbf{s}$	$\mathbf{a}$	$a_1 + a_2$
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	1	0	0
0	1	0	1	0	1	0	1	0	0
0	1	1	0	1	0	0	0	1	1
1	0	0	0	0	1	0	1	0	0
1	0	1	1	0	0	1	0	1	1
1	1	0	1	0	0	1	0	1	1
1	1	1	0	1	1	0	1	1	1

pues como podemos apreciar, $\mathbf{s} = s_2$ y $\mathbf{a} = a_1 + a_2$.

Denotaremos a este circuito como *suma total*.

Veamos ahora como calcular la suma de dos números entre 0 y 7. Supongamos que estos números se escriben en binario como $x_2x_1x_0$ e $y_2y_1y_0$. Su suma, escrita en binario es $s_3s_2s_1s_0$.

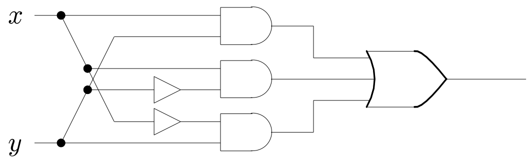


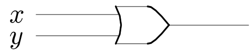
2.3.5. Optimización de funciones booleanas

Hemos visto en la subsección anterior como a partir de la representación de una función booleana en n variables haciendo uso de una expresión booleana podemos diseñar un circuito que nos devuelva el resultado de aplicar la función a las n variables.

Puesto que cualquier función booleana puede expresarse como suma de minterm, podemos diseñar cualquier circuito empleando las tres puertas NOT, OR y AND. Sin embargo, la expresión de una función como suma de minterm no es en general la más apropiada pues requiere de muchas operaciones, lo que se traduce en la necesidad de emplear gran cantidad de puertas.

Así, por ejemplo, los siguientes circuitos producen el mismo efecto sobre las entrada x, y.





pues el primer circuito responde a la función $x\,y + x\,\overline{y} + \overline{x}\,y$, mientras que el segundo a $x + y$, que vimos anteriormente que son iguales.

Lo que pretendemos es, a partir de una expresión de suma de minterm, transformarla en otra expresión equivalente con menos sumandos y menor productos en los sumandos. Vamos a estudiar dos métodos para este propósito. Por una parte, los mapas de Karnaugh, y por otra parte el método de Quine-McCluskey.

Referente al primero, decir que es un método eficiente para funciones de no más de cuatro variables, mientras que puede utilizarse hasta funciones de seis variables.

En cuanto al segundo, aunque realiza la optimización de forma automática, y podría implementarse como un programa informático, el algoritmo, para un número grande de variables booleanas, es computacionalmente muy costoso.

Mapas de Karnaugh

En primer lugar veremos qué es un mapa de Karnaugh, y posteriormente veremos cómo utilizarlos para optimizar las expresiones booleanas.

Un mapa de Karnaugh para una función booleana de dos, tres o cuatro variables es una tabla con tantas celdas como posibles minterm (4 para dos variables, 8 para tres variables y 16 para cuatro variables). Cada celda va asociada a un minterm, y dos celdas adyacentes se diferencian únicamente en un literal, así como dos celdas opuestas en una fila o una columna.

	$\overline{x}$	x
$\overline{y}$		
y		

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}$				
z				

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}\,\overline{t}$				
$\overline{z}\,t$				
$z\,t$				
$z\,\overline{t}$				

Por ejemplo, en el mapa correspondiente a 4 variables, las cuatro celdas adyacentes a $\overline{x}\,y\,\overline{z}\,t$ son: por la derecha, $x\,y\,\overline{z}\,t$, por arriba, $\overline{x}\,y\,\overline{z}\,\overline{t}$, por la izquierda, $\overline{x}\,\overline{y}\,\overline{z}\,t$ y por abajo, $\overline{x}\,y\,z\,t$. Vemos como cada una de estas celdas se diferencia de $\overline{x}\,y\,\overline{z}\,t$ en sólo un literal ($\overline{x} - x$ en la primera, $t - \overline{t}$ en la segunda, $y - \overline{y}$ en la tercera y $\overline{z} - z$ en la cuarta).

Vemos también como las celdas opuestas de la misma fila se diferencian también en sólo un literal (en la segunda fila, estas celdas opuestas son $\overline{x}\,\overline{y}\,\overline{z}\,t$ y $x\,\overline{y}\,\overline{z}\,t$), así como las celdas opuestas de una columna.

Si ahora tenemos una función booleana en dos, tres o cuatro variables, su mapa de Karnaugh consiste en la tabla antes descrita, en la que se han destacado aquellas celdas correspondientes a los minterm que aparecen en la forma normal disyuntiva de la función. Nosotros aquí las marcaremos con un 1.

Ejemplo 2.3.14. *Vamos a dibujar los mapas de Karnaugh de las funciones booleanas:* $f(x, y) = x\,\overline{y} + \overline{x}\,\overline{y}$; $f(x, y, z) = (\overline{\overline{x} + \overline{y}})(y\,\overline{z}) + \overline{y}\,\overline{z}$; $f(x, y, z, t) = x\,y\,\overline{z}\,\overline{t} + x\,\overline{y}\,z\,t + x\,y\,z\,\overline{t} + x\,y\,z\,t + \overline{x}\,\overline{y}\,z\,\overline{t} + x\,\overline{y}\,z\,\overline{t} + x\,y\,\overline{z}\,t.$

	$\overline{x}$	x
$\overline{y}$	1	1
y		

$x\,\overline{y}+\overline{x}\,\overline{y}$

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}$	1		1	1
z	1	1	1	1

$(\overline{\overline{x}+\overline{y}})(y\,\overline{z})+\overline{y}\,\overline{z}$

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}\,\overline{t}$			1	
$\overline{z}\,t$			1	
$z\,t$			1	1
$z\,\overline{t}$	1		1	1

$x\,y\,\overline{z}\,\overline{t}+x\,\overline{y}\,z\,t+x\,y\,z\,\overline{t}+x\,y\,z\,t+\overline{x}\,\overline{y}\,z\,\overline{t}+x\,\overline{y}\,z\,\overline{t}+x\,y\,\overline{z}\,t$

Una vez dibujado el mapa de Karnaugh de una función booleana, se buscan los 1 que aparezcan en celdas adyacentes (u opuestas en una misma fila o columna). Dos de estas celdas se transforman en un único producto en el que ha desaparecido el literal en que difieren. Así, en el ejemplo del mapa de Karnaugh para la función de dos variables, tenemos dos "unos" adyacentes, situados en las celdas $x\,\overline{y}$ y $\overline{x}\,\overline{y}$. Estas dos celdas dan lugar a un producto en el que desaparece el literal diferente (x, $\overline{x}$), quedando entonces la expresión booleana $\overline{y}$. Obviamente, lo único que estamos haciendo es la transformación $x\,\overline{y}+\overline{x}\,\overline{y} = (x+\overline{x})\overline{y} = 1\cdot\overline{y} = \overline{y}$, donde se ha empleado la propiedad distributiva, la definición de complementario y de 1. Los mapas de Karnaugh constituyen una representación gráfica de una función booleana que ayuda a encontrar los minterm que podemos agrupar.

De la misma forma, si encontramos cuatro "unos" adyacentes, formando, bien un cuadrado, bien una línea (fila o columna), podemos sustituirlos por un solo producto en el que se eliminan los dos litereles que difieren en esas cuatro celdas.

El objetivo es tratar de agrupar los "unos" en el menor número posible de bloques, y de mayor tamaño.

Vamos a optimizar las dos funciones que hemos representado mediante mapas de Karnaugh en el ejemplo anterior. En primer lugar consideramos la función $f:\mathbb{B}^3\to\mathbb{B}$ dada por $f(x,y,z) = (\overline{\overline{x}+\overline{y}})(y\,\overline{z})+\overline{y}\,\overline{z}$.

Su mapa de Karnaugh es:

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}$	1		1	1
z	1	1	1	1

y vemos que podemos agrupar en 3 bloques, lo que da lugar a tres sumandos, que son $\overline{x}\,\overline{y}$, $\overline{x}\,y\,z$ y x. Es decir, $f(x,y,z) = \overline{x}\,\overline{y}+\overline{x}\,y\,z+x$.

Vemos también que podemos hacer otras agrupaciones en 3 bloques. Por ejemplo,

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}$	1		1	1
z	1	1	1	1

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}$	1		1	1
z	1	1	1	1

que dan lugar a las expresiones $f(x, y, z) = \overline{x}\,\overline{y} + \overline{x}\,z + x$ o a $f(x, y, z) = \overline{x}\,\overline{y} + z + x$.

Sin embargo, no olvidemos que también se consideran adyacentes las celdas opuestas de una misma fila. Podemos entonces agrupar en los siguientes bloques:

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}$	1		1	1
z	1	1	1	1

lo que da lugar a la expresión $f(x, y, z) = x + \overline{y} + z$.

Podemos apreciar como en todas las optimizaciones obtenidas hemos obtenido tres sumandos. Sin embargo, esta última parece mejor, pues es la que tiene menos productos en cada sumando. Esto viene de haber obtenido los bloques más grandes.

Por último, vamos a optimizar la función $f : \mathbb{B}^4 \to \mathbb{B}$ dada por

$$f(x, y, z, t) = x\,y\,\overline{z}\,\overline{t} + x\,\overline{y}\,z\,t + x\,y\,z\,\overline{t} + x\,y\,z\,t + \overline{x}\,\overline{y}\,z\,\overline{t} + x\,\overline{y}\,z\,\overline{t} + x\,y\,\overline{z}\,t$$

Para esto, agrupamos los "unos" del mapa de Karnaugh por bloques:

	$\overline{x}\,\overline{y}$	$\overline{x}\,y$	$x\,y$	$x\,\overline{y}$
$\overline{z}\,\overline{t}$			1	
$\overline{z}\,t$			1	
$z\,t$			1	1
$z\,\overline{t}$	1		1	1

lo que da lugar a la expresión $f(x, y, z, t) = x\,y + x\,z + \overline{y}\,z\,\overline{t}$.

Método de Quine - McCluskey

Acabamos de ver cómo los mapas de Karnaugh nos ayudan a minimizar el desarrollo de una función booleana como suma de productos. Sin embargo, este método se basa en la visualización de la función en un diagrama, y es poco eficiente para funciones de más de cuatro variables. Sería conveniente tener un proceso que pudiera automatizarse. El método de Quine-McCluskey se ajusta a esta condición. El método consta de dos partes. En una primera, se determinan que términos son candidatos a que aparezcan en un desarrollo minimal. En la segunda se seleccionan de estos candidatos los que intervienen en dicho desarrollo.

Describamos a continuación el método.

Sabemos que cada minterm en n variables va unido a una secuencia de n bits. En primer lugar, dada una función booleana como suma de minterm, ordenamos las cadenas de bits en una columna, agrupando aquellos en los que aparecen igual cantidad de "unos".

Comparamos las cadenas de un grupo con las del grupo inmediatamente inferior. Si encontramos dos cadenas que difieren únicamente en un bit, las marcamos y, en una columna situada a la derecha, representamos estas dos cadenas por una nueva en la que sustituimos el bit diferente por $-$. Si aparecieran dos cadenas iguales, se deja únicamente una.

Una vez realizadas todas las comparaciones posibles, en esta nueva columna repetimos el proceso.

Se continúa así hasta que no podamos obtener una nueva columna.

Se seleccionan aquellas cadenas que no hayan sido marcadas.

Veamos esto con un ejemplo.

Ejemplo 2.3.15. *Sea la expresión booleana* $x\,y\,z + \overline{x}\,y\,z + \overline{x}\,\overline{y}\,z + \overline{x}\,\overline{y}\,\overline{z} + x\,\overline{y}\,z$

Cada minterm lo representamos mediante una cadena de tres dígitos binarios. Estos son 111, 011, 001, 000 *y* 101. *Ordenamos las cadenas en una columna, situando en primer lugar la que tiene* 3 *"unos", a continuación las que tienen* 2 *"unos" y así sucesivamente.*

111
011
101
001
000

Comparamos la cadena del primer nivel con las de segundo. Resulta que hay 2 *de las que se diferencia en un único dígito. Sustituimos este dígito por* $-$, *luego nos queda* -11 *y* $1-1$.

Comparamos las cadenas del segundo y tercer nivel, y vemos que 101 *y* 001 *se diferencian en un único dígito. Esto da lugar a la cadena* -01. *Por último, comparamos la del tercer nivel con el cuarto, lo que nos da* $00-$.

Ordenamos todos estos datos en una nueva columna y todas las cadenas de esta columna que han intervenido en alguna de la segunda las marcamos.

~ 111	-11
~ 011	$1{-}1$
~ 101	$0{-}1$
~ 001	-01
~ 000	$00-$

Repetimos aquí el proceso con la segunda columna

~ 111	~ -11	$--1$
~ 011	$\sim 1{-}1$	
~ 101	$\sim 0{-}1$	
~ 001	~ -01	
~ 000	$00-$	

Las cadenas a seleccionar son entonces las no marcadas, es decir, $--1$ *y* $00-$, *que se corresponden con los términos* z *y* $\overline{x}\,\overline{y}$.

La segunda parte de este método consiste en encontrar, de todos los productos booleanos, el menor conjunto de ellos que represente a la expresión booleana dada. Para ello, hacemos una tabla en la que, en el eje horizontal situamos los minterm que nos definían la expresión booleana, mientras que en el eje vertical situamos los productos booleanos que hemos seleccionado en la primera parte. A continuación señalamos las celdas que se correspondan con un producto booleano y un minterm con la condición de que todos los literales que intervienen en el producto booleano también se encuentren en el minterm.

Una vez hecho esto, elegimos la menor cantidad de productos booleanos de forma que uniendo las celdas que están señaladas en sus filas podamos completar una fila completa de la tabla. De haber varias posibles elecciones, nos quedamos con aquellas en que los productos booleanos tengan la menor cantidad posible de literales.

Ejemplo 2.3.16. *En el ejemplo anterior, la tabla nos quedaría como sigue:*

	$x\,y\,z$	$x\,\overline{y}\,z$	$\overline{x}\,y\,z$	$\overline{x}\,\overline{y}\,z$	$\overline{x}\,\overline{y}\,\overline{z}$
z	X	X	X	X	
$\overline{x}\,\overline{y}$				X	X

Vemos como aquí necesitamos los dos productos booleanos para rellenar una fila.
Después de todo esto deducimos que:

$$x\,y\,z+\overline{x}\,y\,z+\overline{x}\,\overline{y}\,z+\overline{x}\,\overline{y}\,\overline{z}+x\,\overline{y}\,z=\overline{x}\,\overline{y}+z$$

Veamos a continuación un ejemplo completo.

Ejemplo 2.3.17. *Dada la expresión booleana* $x\,\overline{y}\,\overline{z}\,\overline{t}+\overline{x}\,y\,\overline{z}\,t+x\,\overline{y}\,z\,\overline{t}+\overline{x}\,\overline{y}\;z\,t+\overline{x}\,\overline{y}\,\overline{z}\,t+\overline{x}\,y\,z\,t+\overline{x}\,\overline{y}\,z\,\overline{t}$ *vamos a tratar de encontrar una expresión óptima mediante el método de Quine-McCluskey.*

Las cadenas de bits correspondientes a cada uno de los minterm son 0111, 0101, 1010, 0011, 0001, 0111 *y* 0010. *A partir de ellas construimos la tabla:*

a b	$\sim$ *0111*	*1*	$\sim$ *01-1*	*0- -1*
a c	$\sim$ *0101*	*2*	$\sim$ *0-11*	
d e	$\sim$ *1010*	*2*	$\sim$ *0-01*	
b f g	$\sim$ *0011*		*10-0*	
d	$\sim$ *1000*		*-010*	
c f	$\sim$ *0001*	*1*	$\sim$ *00-1*	
e g	$\sim$ *0010*		*001-*	

A la izquierda de cada una de las cadenas marcadas hemos colocado unas letras (columna de la izquierda) o números (columna central) que nos indican cada cadena con cual se empareja para formar una cadena en la columna que tiene más a la derecha.

A partir de aquí seleccionamos las cadenas que no están marcadas, que se corresponden con los productos $x\,\overline{y}\,\overline{t}$*;* $\overline{y}\,z\,\overline{t}$*;* $\overline{x}\,\overline{y}\,z$*;* $\overline{x}\,t$*. Esto nos da la siguiente tabla:*

	$x\,\overline{y}\,\overline{z}\,\overline{t}$	$\overline{x}\,y\,\overline{z}\,t$	$x\,\overline{y}\,z\,\overline{t}$	$\overline{x}\,\overline{y}\;z\,t$	$\overline{x}\,\overline{y}\,\overline{z}\,t$	$\overline{x}\,y\,z\,t$	$\overline{x}\,\overline{y}\,z\,\overline{t}$
$x\,\overline{y}\,\overline{t}$	X		X				
$\overline{y}\,z\,\overline{t}$			X				X
$\overline{x}\,\overline{y}\,z$				X			X
$\overline{x}\,t$		X		X	X	X	

Y a partir de la tabla se ve cómo los términos $x\,\overline{y}\,\overline{t}$ *y* $\overline{x}\,t$ *tienen que aparecer en la expresión simplificada. Si el primero no lo pusiéramos, no quedaría cubierto el minterm* $x\,\overline{y}\,\overline{z}\,\overline{t}$*, mientras que si no lo hiciéramos con el segundo, sería el minterm* $\overline{x}\,y\,\overline{z}\,t$ *(y otros dos más) quien no quedaría cubierto. Aquellos términos que son los únicos que cubren a alguno de los minterms, se les llama* implicantes primos.

En este caso, los implicantes primos son $x\,\overline{y}\,\overline{t}$ *y* $\overline{x}\,t$*.*

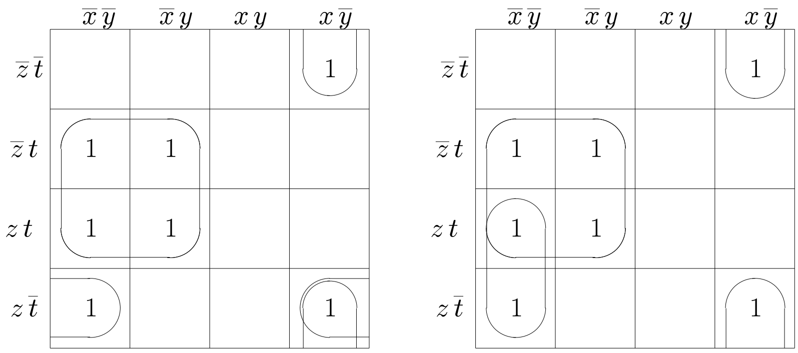
Lo que hacemos ahora es, de la tabla anterior, eliminamos las filas donde están los implicantes primos, y las columnas donde están los minitérminos que quedan cubiertos por esos implicantes primos. Es decir, suprimimos las filas primera (correspondiente al término $x\,\overline{y}\,\overline{t}$*) y cuarta (correspondiente al término* $\overline{x}\,t$*). También suprimimos las columnas primera, segunda, tercera, cuarta, quinta y sexta. Nos queda entonces:*

	$\overline{x}\,\overline{y}\,z\,\overline{t}$
$\overline{y}\,z\,\overline{t}$	X
$\overline{x}\,\overline{y}\,z$	X

y vemos que eligiendo cualquiera de los dos, podemos tener cubiertos todos los minterms. Por tanto,

$$x\,\overline{y}\,\overline{z}\,\overline{t}+\overline{x}\,y\,\overline{z}\,t+x\,\overline{y}\,z\,\overline{t}+\overline{x}\,\overline{y}\;z\,t+\overline{x}\,\overline{y}\,\overline{z}\,t+\overline{x}\,y\,z\,t+\overline{x}\,\overline{y}\,z\,\overline{t}=x\,\overline{y}\,\overline{t}+\overline{x}\,t+\overline{y}\,z\,\overline{t}=x\,\overline{y}\,\overline{t}+\overline{x}\,t+\overline{x}\,\overline{y}\,z$$

Si para optimizar esta expresión booleana empleáramos los mapas de Karnaugh tendríamos dos formas diferentes de agrupar las celdas con "unos":



lo que nos da las dos expresiones que acabamos de ver.

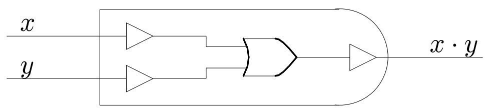
2.3.6. Conjuntos funcionalmente completos

En secciones precedentes hemos visto cómo a partir de tres puertas lógicas, las puertas AND, OR y NOT, podemos diseñar circuitos lógicos, y en la sección anterior hemos estudiado cómo minimizar el número de puertas necesarias para diseñar tales circuitos. Sin embargo, para construir tales circuitos es necesario emplear tres tipos de puertas lógicas diferentes. En esta sección vamos a intentar construir cualquier circuito lógico empleando menos puertas diferentes, aunque sea a costa de aumentar el número de éstas.

Comenzamos reduciendo el número de puertas distintas necesarias a 2.

Proposición 2.3.3. *Las puertas lógicas OR y NOT, o las puertas lógicas AND y NOT son suficientes para la construcción de cualquier circuito lógico.*

Demostración: Para ver que las puertas OR y NOT son suficientes, basta comprobar que $xy = \overline{\overline{x} + \overline{y}}$, lo que nos dice que podemos construir una puerta AND usando las puertas OR y NOT. Esta puerta podría quedar como sigue:



Y por tanto, usando únicamente puertas OR y NOT podemos construir cualquier circuito.

De la misma forma, y puesto que $x + y = \overline{\overline{x}\,\overline{y}}$ se puede ver que usando únicamente las puetas AND y NOT se puede diseñar cualquier circuito.

■

Definición 24. *Sean x, y variables booleanas. Se definen las funciones booleanas $-\uparrow-$ y $-\downarrow-$ como sigue:*

$$x \uparrow y = \overline{x \cdot y}, \qquad x \downarrow y = \overline{x + y}.$$

Es decir:

$$0 \uparrow 0 = 1, \quad 0 \uparrow 1 = 1, \quad 1 \uparrow 0 = 1, \quad 1 \uparrow 1 = 0.$$
$$0 \downarrow 0 = 1, \quad 0 \downarrow 1 = 0, \quad 1 \downarrow 0 = 0, \quad 1 \downarrow 1 = 0.$$

Estos operadores se denotan como NAND (NOT AND) y NOR (NOT OR) respectivamente.

Proposición 2.3.4. *Cualquier función booleana se puede expresar usando únicamente el operador NAND (resp. NOR).*

Demostración:

Para comprobar esto, escribimos en primer lugar:

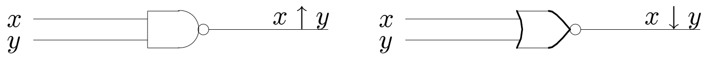
$\overline{x} = \overline{x \cdot x} = x \uparrow x$,

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

es decir, los operadores NOT y OR pueden expresarse utilizando únicamente NAND. La Proposición 2.3.3 nos dice que cualquier función booleana la podemos expresar únicamente con el operador NAND.

De la misma forma, puesto que $\overline{x} = x \downarrow x$ y $x \cdot y = (x \downarrow x) \downarrow (y \downarrow y)$ deducimos que el operador NOR es suficiente para expresar cualquier función booleana. ■.

Las puertas correspondientes NAND y NOR se suelen representar como sigue:



Cualquiera de los circuitos que vimos en la Sección 3.3.4, o cualquier otro que se nos ocurra podemos ahora diseñarlo usando únicamente la puerta NAND (o la puerta NOR). Esta puerta se construye de forma sencilla con transistores, tanto con la tecnología de semiconductores como con las técnicas más recientes de fabricación de microcircuitos.