Vector space of bit vectors

Ralf Poeppel
mailto:ralf@poeppel-familie.de

2026-02-06

Abstract

This article is a supplementary documentation to the Go package `gf2vs` [10]. The package implements data types and functions for the vector space of bit vectors.

1 Introduction

Bit vectors are very common in computer science. They are used for integers, combinatorial algorithms, coding theory, and for logical and arithmetic operations [7, 5]. All aspects of the vector space of bit vectors are examined relatively rarely.

Bits are based on the finite set of integers of order 2. This set $\{0, 1\}$ with the operations addition and multiplication modulo 2 satisfies the axioms of a field. This finite field is named Galois Field [8, 15][1] $GF(2) = \mathbb{F}_2$. Over this field there is the vector space $\mathbb{F}_2^n$.

The aim of this article is to document the properties of the vector space of bit vectors $\mathbb{F}_2^n$, as implemented in the Go package `gf2vs`. This is a brief reference of the properties collected from several sources.

2 Field $GF(2)$

2.1 Supporting Set

The supporting set of $GF(2) = \mathbb{F}_2$ is

$$\mathbb{Z}_2 = \mathbb{Z}/\mathbb{Z}2 = \{0, 1\} \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{R} \tag{1}$$

the subset $\mathbb{Z}_2$ of $\mathbb{Z}$, which is a subset of $\mathbb{R}$. This set is equal to $\mathbb{Z}/\mathbb{Z}2$, the cyclic group of order 2. This set holds the values of a bit in computer science. In logic we have the boolean values False $F = 0$ and True $T = 1$ [6].

2.2 Operations

The operations of $\mathbb{F}_2 = \mathbb{Z}/\mathbb{Z}2$ are defined modulo 2; see [2, ch. 2.2.6] and [5].

The operations addition and multiplication of the field $\mathbb{F}_2$ satisfies the group axioms [2, 8, 16], both operations are commutative.

[1] We cite Wikipedia for reused wordings.

Similarly the operations addition and multiplication of the field $\mathbb{R}$ satisfies the group axioms [2], both operations are commutative.

Please note the different definition of the addition in $\mathbb{F}_2$ and $\mathbb{R}$.

For reference we give here only the operations for $\mathbb{F}_2$.

2.2.1 Negation

$$- : \mathbb{F}_2 \to \mathbb{F}_2, \quad -x = x, \quad -0 = 0, \quad -1 = 1. \tag{2}$$

The negation of 1 in $\mathbb{F}_2$ is computed as $(-1) \mod 2 = 1$

2.2.2 Complement

$$\neg : \mathbb{F}_2 \to \mathbb{F}_2, \quad \neg x = 1 - x, \quad \neg 0 = 1, \quad \neg 1 = 0. \tag{3}$$

2.2.3 Absolute value

We define the mapping absolute value $|x|$ of an element x of $\mathbb{F}_2$ to $\mathbb{R}$:

$$|x| : \mathbb{F}_2 \to \mathbb{R}, \quad |0| = 0, \quad |1| = 1. \tag{4}$$

We use this mapping, when we need the default classical definition of the addition as in $\mathbb{R}$.

2.2.4 Addition

The addition is named exclusive disjunction in logic and XOR [6, 14] in computer science. The definition of addition is given in equation 5 obeying $(1 + 1) \mod 2 = 0$.

$$+ : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2, \quad 0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 0, \tag{5}$$

The group axioms [2, ch. 2.2.8] for the Group $G = \mathbb{F}_2$ and the operation addition are satisfied:

Associativity
$\forall a, b, c \in G : (a + b) + c = a + (b + c)$.

Identity element $e = 0$
$\exists e \in G, \forall a \in G : e + a = a$ and $a + e = a$, $e = 0$, e is unique.

Inverse element $(-a) = a$
$\forall a \in G \;\; \exists(-a) \in G : a + (-a) = e$ and $(-a) + a = e$, e identity element, $(-a) = a$ is unique for each a.

Commutativity
$a + b = b + a$.

So $\mathbb{F}_2$ with the operation addition is an abelian group.

2.2.5 Multiplication

The multiplication is named conjunction in logic and AND [6, 18] in computer science. The multiplication is identically defined as in $\mathbb{Z}$.

$$\cdot : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2, \quad 0 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1. \tag{6}$$

We may use the notation ab instead of $a \cdot b$, omitting the multiplication sign if there is no ambiguity.

The group axioms for the Group $G = \mathbb{F}_2$ and the operation multiplication are satisfied:

Associativity
$\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c)$.

Identity element $e = 1$
$\exists e \in G, \forall a \in G : e \cdot a = a$ and $a \cdot e = a$, $e = 1$, e is unique.

Inverse element a^{-1}
$\forall a \in G, a \neq 0, \ \exists a^{-1} \in G : a \cdot a^{-1} = e$ and $a^{-1} \cdot a = e$, e is the identity element; the only invertible element is 1, hence $a^{-1} = 1$ for $a = 1$.

Commutativity
$a \cdot b = b \cdot a$.

So $\mathbb{F}_2$ with the operation multiplication is an abelian group.

2.2.6 Disjunction

In boolean logic we have the operation disjunction, named OR in computer science [6, 19].

$$\vee : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2, \quad 0 \vee 0 = 0, \quad 0 \vee 1 = 1, \quad 1 \vee 0 = 1, \quad 1 \vee 1 = 1 \tag{7}$$

The operation $\vee$ does not satisfy the group axioms; there is no inverse element.

2.3 Field axioms

The set $K := \mathbb{F}_2$ with the operations addition and multiplication satisfies the field axioms [2, 5]. We use K as symbol for any field satisfying the field axioms.

K1 K with the addition $+$ is an abelian group.

K2 $K^* := K \setminus \{0\}$ with the multiplication $\cdot$ for every element of K^* is an abelian group.

K3 distributive property [1, 12] is satisfied $\forall a, b, c \in K$

$$\begin{aligned} a \cdot (b + c) &= a \cdot b + a \cdot c, \\ (a + b) \cdot c &= a \cdot c + b \cdot c. \end{aligned} \tag{8}$$

3 Vector space $\mathbb{F}_2^n$

3.1 Vectors

Bit vectors are the elements of the vector space. We define a bit vector x of size n as a tuple (x_i) of values $x_i \in \mathbb{F}_2$:

$$x := (x_i) := (x_1, x_2, \ldots, x_n), \quad \forall x_i \in \mathbb{F}_2 \tag{9}$$

We define the set of all bit vectors of size n see [2, ch 2.4.1]:

$$\mathbb{F}_2^n := \{x = (x_1, \ldots, x_n) : x_i \in \mathbb{F}_2\} \tag{10}$$

In addition we define two distinguished constant elements of $\mathbb{F}_2^n$:

Zero $\mathbb{0}$ zero vector, all components are 0.

Ones $\mathbb{1}$ ones vector, all components are 1.

3.2 Mapping of Integers to Bit Vectors

We define the mapping φ from $\mathbb{N}$ to $\mathbb{F}_2^n$. For mapping of an integer $a \in \mathbb{N}$ to a bit vector $x \in \mathbb{F}_2^n$ we need to compute the length n of x with equation (11).

$$n = \lfloor \log_2(a) \rfloor + 1. \tag{11}$$

If we need to map multiple integers a_i to the same vector space we compute n from the maximum of the a_i

$$n = \lfloor \log_2(\max(a_i)) \rfloor + 1. \tag{12}$$

Equation (13) defines the mapping for $a \in \mathbb{N}, a \geq 0$.

$$\varphi : \mathbb{N} \to \mathbb{F}_2^n, \quad x = \left(\lfloor \tfrac{a}{2^0} \rfloor \bmod 2,\ \lfloor \tfrac{a}{2^1} \rfloor \bmod 2,\ \ldots,\ \lfloor \tfrac{a}{2^{n-1}} \rfloor \bmod 2 \right). \tag{13}$$

The inverse mapping φ^{-1} is given by equation (14)

$$\varphi^{-1} : \mathbb{F}_2^n \to \mathbb{N}, \quad a = \textstyle\sum_{i=1}^n x_i 2^{i-1}. \tag{14}$$

In the programming language Go these mapping are used when parsing integer strings and formatting integers. Internally integers are stored as bit vectors.

3.3 Operations on vectors

We define bitwise operations on the bit vectors x, y, z see [2, ch. 2.4.1] and [7, (1), (2), (3)].

$$\left.\begin{array}{rrcr}
- : \mathbb{F}_2^n \to \mathbb{F}_2^n, & -x = x & \Leftrightarrow & -x_i = x_i \\
\sim : \mathbb{F}_2^n \to \mathbb{F}_2^n, & \sim x = y & \Leftrightarrow & \neg x_i = y_i \\
|\,| : \mathbb{F}_2^n \to \mathbb{R}^n, & |x| = y & \Leftrightarrow & x_i = y_i, \\
\oplus : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n, & x \oplus y = z & \Leftrightarrow & x_i + y_i = z_i, \\
\& : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n, & x \& y = z & \Leftrightarrow & x_i \cdot y_i = z_i, \\
| : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n, & x|y = z & \Leftrightarrow & x_i \vee y_i = z_i, \\
\cdot : \mathbb{F}_2 \times \mathbb{F}_2^n \to \mathbb{F}_2^n, & \lambda \cdot x = y & \Leftrightarrow & \lambda \cdot x_i = y_i,
\end{array}\right\} i = 1, \ldots, n. \tag{15}$$

We define the operation $| \, |$ for formal mapping of a bit vector from $\mathbb{F}_2^n$ to $\mathbb{R}^n$.

For the constants we have: $\sim \mathbb{0} = \mathbb{1}$ and $\sim \mathbb{1} = \mathbb{0}$.

We adopt the main identities from [7, (4), ..., (14)] for bit vectors of size n here:

$$x \oplus y = y \oplus x, \quad x\&y = y\&x, \quad x|y = y|x; \tag{16}$$
$$(x \oplus y) \oplus z = x \oplus (y \oplus z), \quad (x\&y)\&z = x\&(y\&z), \quad (x|y)|z = x|(y|z); \tag{17}$$
$$(x \oplus y)\&z = (x\&z) \oplus (y\&z); \tag{18}$$
$$(x\&y)|z = (x|z)\&(y|z), \quad (x|y)\&z = (x|z)\&(y|z); \tag{19}$$
$$x \oplus y = (x\&y) \oplus (x|y); \tag{20}$$
$$(x\&y)|x = x, \quad (x|y)\&x = x; \tag{21}$$
$$x \oplus \mathbb{0} = x, \quad x\&\mathbb{0} = \mathbb{0}, \quad x|\mathbb{0} = x; \tag{22}$$
$$x \oplus x = \mathbb{0}, \quad x\&x = x, \quad x|x = x; \tag{23}$$
$$x \oplus \mathbb{1} =\sim x, \quad x\&\mathbb{1} = x, \quad x|\mathbb{1} = \mathbb{1}; \tag{24}$$
$$x \oplus (\sim x) = \mathbb{1}, \quad x\&(\sim x) = \mathbb{0}, \quad x|(\sim x) = \mathbb{1}; \tag{25}$$
$$-(x \oplus y) = (\sim x) \oplus y = x \oplus (\sim y), \quad \sim (x\&y) = (\sim x)|(\sim y), \quad \sim (x|y) = (\sim x)\&(\sim y); \tag{26}$$

3.4 Axioms of vector space

The set $\mathbb{F}_2^n$ with the binary operation of vector addition $\oplus$ and the binary function of scalar multiplication $\cdot$, as given in (15), defines a vector space see [2, 21].

The axioms of a vector space are satisfied for $\mathbb{F}_2^n$:

Associativity of vector addition
$u \oplus (v \oplus w) = (u \oplus v) \oplus w, \;\; \forall u, v, w \in \mathbb{F}_2^n.$

Commutativity of vector addition
$u \oplus v = v \oplus u, \;\; \forall u, v \in \mathbb{F}_2^n.$

Identity element of vector addition
$\exists \mathbb{0} \in \mathbb{F}_2^n : v \oplus \mathbb{0} = v, \;\; \forall v \in \mathbb{F}_2^n.$

Inverse elements of vector addition
$\forall v \in \mathbb{F}_2^n \;\; \exists -v \in \mathbb{F}_2^n : v \oplus (-v) = \mathbb{0}$, and $-v = v$, i.e. each vector is its own additive inverse.

Compatibility of scalar multiplication with field multiplication
$\lambda(\eta v) = (\lambda\eta)v, \;\; \lambda, \eta \in \mathbb{F}_2, \;\; v \in \mathbb{F}_2^n.$

Identity element of scalar multiplication
$1v = v, \;\; 1 \in \mathbb{F}_2, \;\; v \in \mathbb{F}_2^n$, where 1 is the multiplicative identity of $\mathbb{F}_2$.

Distributivity of scalar multiplication with respect to vector addition
$\lambda(u \oplus v) = \lambda u \oplus \lambda v, \;\; \lambda \in \mathbb{F}_2, \;\; u, v \in \mathbb{F}_2^n.$

Distributivity of scalar multiplication with respect to field addition
$(\lambda + \eta)v = \lambda v + \eta v, \;\; \lambda, \eta \in \mathbb{F}_2, \;\; v \in \mathbb{F}_2^n.$

In the vector space $\mathbb{F}_2^n$ we are not limited to the operations vector addition and scalar multiplication. We can use the Boolean operations as well.

Negation, Complement, Not
$\sim v = \mathbb{1} - v = \mathbb{1} \oplus v$, swap all bits.

Disjunction, Or
$u|v = (u_i)|(v_i) = (u_i \vee v_i)$, element-wise Or.

Exclusive or, Xor
$u \oplus v = u \oplus v = (u_i) \oplus (v_i) = (u_i + v_i)$, element-wise Xor, equal to vector addition. »»»> doc

Conjunction, And
$u \wedge v = (u_i) \cdot (v_i) = (u_i \cdot v_i)$, element-wise And.

As we apply the operations element-wise $i = 1, \ldots n$, we satisfy the laws of associativity and commutativity inherited from the field.

3.5 Vector space base

We give here the definition of the base, the norm and the scalar product implemented in the Go package. The symbol K is used in definitions applicable by each of the fields $\mathbb{F}_2^n, \mathbb{Q}, \mathbb{R}, \mathbb{C}$.

3.5.1 Unit vector

We define the unit vectors e_i, $i = 1, \ldots, n$, of the vector space as the vectors where the ith element is $x_i = 1$ and all other elements are 0.

$$e_i = (x_k), \quad e_i \in K^n, \quad x_k \in K, \quad x_k = \begin{cases} 1, & k = i, \quad \text{identity element of multiplication,} \\ 0, & k \neq i, \quad \text{identity element of addition.} \end{cases} \tag{27}$$

Please note the e_i are linearly independent.

We observe the identity elements are identical for the fields and the unit vectors are identical for all vector spaces over a field K.

3.5.2 Generating system

We define the subset $\mathbb{E} := \{e_i\}$, $\mathbb{E} \subset \mathbb{F}_2^n$, of unit vectors e_i. The subset $\mathbb{E}$ forms a generating system. Each vector v of $\mathbb{F}_2^n$ is a linear combination of scalars a_i and the e_i:

$$v = \sum_{i=1}^{n} a_i e_i, \quad a_i \in \mathbb{F}_2, \quad e_i \in \mathbb{E}, \quad \forall v \in \mathbb{F}_2^n. \tag{28}$$

Here the addition is modulo 2.

Equation (28) is used equally for each vector space on any field K using the operation addition as defined for the field K and the 1 the identity element of the operation multiplication.

Thus the subset $\mathbb{E}$ spans $\mathbb{F}_2^n$. In this vector space it is one spanning set, and the decomposition of a vector v into a linear combination of unit vectors e_i is unique.

3.5.3 Base

The subset $\mathbb{E}$ is one base of the vector space $\mathbb{F}_2^n$. As it is a base of every vector space over a field K.

3.5.4 Index

We call $i = 1, \ldots, n$ the index of the unit vector e_i in the base.

3.5.5 Norm via Hamming weight

The addition in the field $\mathbb{F}_2$ is modulo 2. Hence each sum in $\mathbb{F}_2$ evaluates to either 0 or 1. In particular, for $x \in \mathbb{F}_2^n$ we have

$$\left(\sum_{i=1}^{n} x_i\right) \mod 2 = \begin{cases} 1, & |\{i \in \{1,\dots,n\} : x_i = 1\}| \text{ is odd}, \\ 0, & |\{i \in \{1,\dots,n\} : x_i = 1\}| \text{ is even}. \end{cases} \tag{29}$$

From this it follows that we cannot directly use the usual norm definitions (as over $\mathbb{R}$) to measure vector length in $\mathbb{F}_2^n$.

In coding theory [4, 17] the *Hamming weight* w_H (number of 1-entries) of $x \in \{0,1\}^n$ is defined as:

$$w_H : \mathbb{F}_2^n \to \mathbb{Z}, \quad w_H(x) = |\{i \in \{1,\dots,n\} : x_i = 1\}|, \tag{30}$$

and the associated *Hamming distance* d_H is:

$$d_H : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{Z}, \quad d_H(x,y) = w_H(x-y). \tag{31}$$

If we first apply operation $|\ |$ we map a vector from $\mathbb{F}_2^n \to \mathbb{R}^n$, by embedding $\mathbb{F}_2^n = \{0,1\}^n \subset \mathbb{R}^n$. On vectors in $\mathbb{R}^n$ norms are defined and we get:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| = w_H(x). \tag{32}$$

So the l_1-norm on $\mathbb{R}^n$ see [3, 11] equals the Hamming weight on bit vectors. This norm is sometimes called absolute-value norm [20]. The value of the norm is an element of the set $\{0,1,\dots,n\} \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{R}$.

In the programming languages C the function is named popcount() and in the language Go it is the function `OnesCount(uint x) uint` in package `math/bits`.

Obviously this norm satisfies the axioms of a norm:

Subadditivity / Triangle inequality
$w_H(x+y) \leq w_H(x) + w_H(y) \quad \forall x,y \in \mathbb{F}_2^n$,

Absolute homogeneity
$w_H(s \cdot x) = s \cdot w_H(x) \quad \forall s \in \mathbb{F}_2, x \in \mathbb{F}_2^n$,

Positive definiteness
$w_H(x) = 0 \Rightarrow x = \mathbb{0}$.

Please note the norm of the unit vectors $\|e_i\| = 1, \quad \forall i \in \{1,\dots n\}$.

3.5.6 Scalar product

We define the scalar product, dot product [2, 13] or inner product of two vectors as given in equation (33):

$$\langle,\rangle : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{R}, \quad \langle x,y\rangle = \textstyle\sum_{i=1}^{n} |x_i \cdot y_i| = \|x\&y\|_1 = w_H(x\&y), \tag{33}$$

The obtained value equals the standard scalar product of $x, y \in \{0,1\}^n \subset \mathbb{R}^n$

$$\langle,\rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \quad \langle x,y\rangle = \textstyle\sum_{i}^{n} x_i \cdot y_i, \tag{34}$$

We can easily verify by computation the properties of the scalar product:

Distributivity

$$
\begin{aligned}
\langle x \oplus x', y\rangle &= \langle x, y\rangle + \langle x', y\rangle,\\
\langle x, y \oplus y'\rangle &= \langle x, y\rangle + \langle x, y'\rangle,\\
\langle \lambda x, y\rangle &= \lambda\langle x, y\rangle,\\
\langle x, \lambda y\rangle &= \lambda\langle x, y\rangle,
\end{aligned}
$$

Commutativity

$$\langle x, y\rangle = \langle y, x\rangle,$$

Positive definiteness

$$\langle x, x\rangle \quad \geq 0, \text{ with } \langle x, x\rangle = 0 \text{ only if } x = \mathbb{0},$$

Orthogonality

$$\langle x, y\rangle = 0, \text{ we say the two vectors are orthogonal.}$$

In coding theory [9] the orthogonal bit vector is called the dual code.

3.5.7 Orthonormal Basis

With the properties of the norm and the scalar product it follows $\mathbb{E}$ is an orthonormal base, and this is the only orthonormal base of $\mathbb{F}_2^n$.

References

[1] Sheldon Jay Axler. *Linear Algebra Done Right*. 4th ed. Cham: Springer Nature; 2024. ISBN: 9783031410260. URL: `https://linear.axler.net/LADR4e.pdf` (visited on Feb. 3, 2026).

[2] Gerd Fischer and Boris Springborn. *Lineare Algebra: Eine Einführung für Studienanfänger*. de. 19th ed. Wiesbaden, Germany: Springer Spektrum, 2020.

[3] Otto Forster. *Analysis / Otto Forster ; 1: Differential- und Integralrechnung einer Veränderlichen*. Wiesbaden: Springer Spektrum; 2016. ISBN: 9783658115449.

[4] R. W. Hamming. "Error detecting and error correcting codes". In: *Bell System Technical Journal* 29.2 (1950), pp. 147–160. URL: `https://dn710109.ca.archive.org/0/items/bstj29-2-147/bstj29-2-147.pdf` (visited on Feb. 3, 2026).

[5] Raymond Hill. *A first course in coding theory*. Oxford: Clarendon Press; 2004. ISBN: 0198538030.

[6] Donald E. Knuth. *The Art of Computer Programming, Vol 4, Fasc 0. Introduction to Combinatorical Algorithms and Boolean Functions*. Vol. 4 Fascicle 0. Addison-Wesley, 2008.

[7] Donald E. Knuth. *The Art of Computer Programming, Vol 4, Fasc 1. Bitwise Tricks and Techniques*. Vol. 4 Fascicle 1. Addison-Wesley, 2008.

[8] Rudolf Lidl and Harald Niederreiter. *Finite fields*. Vol. 20. Encyclopedia of mathematics and its applications, volume 20. Cambridge: Cambridge University Press; 2000. ISBN: 9780511525926. DOI: `10.1017/CBO9780511525926`.

[9] Jacobus H. Lint. *Introduction to Coding Theory and Algebraic Geometry*. Vol. 12. Oberwolfach seminars, 12. Basel: Birkhäuser Basel; 1988. ISBN: 9783034892865. DOI: `10.1007/978-3-0348-9286-5`.

[10] Ralf Poeppel. *Go package documentation gf2vs*. `https://pkg.go.dev/github.com/rpoe/gf2vs`. [Online; accessed 10-January-2026]. Jan. 6, 2026.

[11] Eric W. Weisstein. *L^1 – Norm From MathWorld–A Wolfram Resource*. https://mathworld.wolfram.com/L1-Norm.html. [Online; accessed 09-October-2025]. July 27, 2025.

[12] Wikipedia contributors. *Distributive property — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Distributive_property&oldid=1329322251. [Online; accessed 28-January-2026]. 2025.

[13] Wikipedia contributors. *Dot product — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Dot_product&oldid=1328631745. [Online; accessed 2-February-2026]. 2025.

[14] Wikipedia contributors. *Exclusive or — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Exclusive_or&oldid=1316886803. [Online; accessed 12-January-2026]. 2025.

[15] Wikipedia contributors. *Finite field — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Finite_field&oldid=1330855394. [Online; accessed 12-January-2026]. 2026.

[16] Wikipedia contributors. *Group (mathematics) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Group_mathematics&oldid=1330839314. [Online; accessed 12-January-2026]. 2026.

[17] Wikipedia contributors. *Hamming weight — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Hamming_weight&oldid=1306107874. [Online; accessed 13-January-2026]. 2025.

[18] Wikipedia contributors. *Logical conjunction — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Logical_conjunction&oldid=1324909528. [Online; accessed 12-January-2026]. 2025.

[19] Wikipedia contributors. *Logical disjunction — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Logical_disjunction&oldid=1317551960. [Online; accessed 12-January-2026]. 2025.

[20] Wikipedia contributors. *Norm (mathematics) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Norm_mathematics&oldid=1326013131. [Online; accessed 14-January-2026]. 2025.

[21] Wikipedia contributors. *Vector space — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Vector_space&oldid=1326882436. [Online; accessed 12-January-2026]. 2025.