WIKIPEDIA
The Free Encyclopedia

Navigation

Main page
Contents
Featured content
Current events
Random article
Donate

Interaction

About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Help

Toolbox

What links here
Related changes
Upload file
Special pages
Permanent link
Cite this page

Print/export

Create a book
Download as PDF
Printable version

Languages

Česky
Deutsch
Esperanto
فارسی
Français
한국어
Italiano
עברית
Македонски
Nederlands
日本語
Norsk (bokmål)
Piemontèis
Polski
Português
Română
Русский
Simple English
Slovenčina
Türkçe
中文

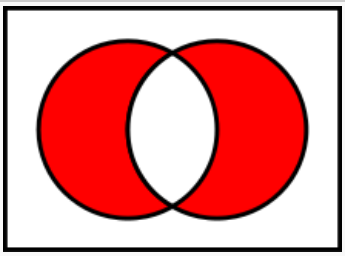
Exclusive or

From Wikipedia, the free encyclopedia

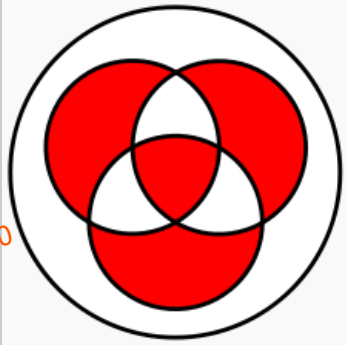
"XOR" redirects here. For other uses, see XOR (disambiguation), XOR gate.
"Either or" redirects here. For Kierkegaard's philosophical work, see Either/Or.

The logical operation **exclusive disjunction**, also called **exclusive or** (symbolized **XOR**, **EOR**, **EXOR**, ⊻ or ⊕, pronounced either /ɛksˈɔːr/ or /zˈɜːr/), is a type of logical disjunction on two operands that results in a value of true if exactly one of the operands has a value of true.[1] A simple way to state this is "one or the other but not both."

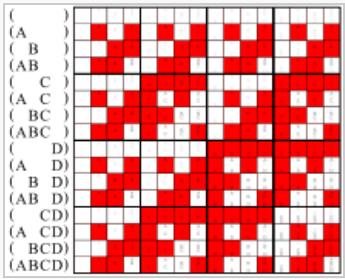
Put differently, **exclusive disjunction** is a logical operation on two logical values, typically the values of two propositions, that produces a value of *true* only in cases where the truth value of the operands differ.



Venn diagram of $A \oplus B$



Venn diagram of $A \oplus B \oplus C$



Arguments on the left combined by XOR
This is a binary Hadamard matrix
(compare: Hadamard code)

Contents

1 Truth table
2 Equivalencies, elimination, and introduction
3 Relation to modern algebra
4 Exclusive "or" in natural language
5 Alternative symbols
6 Properties
6.1 Associativity and commutativity
6.2 Other properties
7 Computer science
7.1 Bitwise operation
8 See also
9 Notes
10 External links

Gonzalez v. Arizona, No. 08-17094 archived on November 4, 2010

Truth table [edit]

The truth table of $A \oplus B$ (also written as A XOR B or $A \neq B$) is as follows:

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Equivalencies, elimination, and introduction [edit]

The following equivalents can then be deduced, written with logical operators, in mathematical and engineering notation:

$$\begin{aligned} p \oplus q &= (p \wedge \neg q) \vee (\neg p \wedge q) &&= p\overline{q} + \overline{p}q \\ &= (p \vee q) \wedge (\neg p \vee \neg q) &&= (p+q)(\overline{p}+\overline{q}) \\ &= (p \vee q) \wedge \neg(p \wedge q) &&= (p+q)(\overline{pq}) \end{aligned}$$

Generalized or **n-ary** XOR is true when the number of 1-bits is odd.

The exclusive disjunction $p \oplus q$ can be expressed in terms of the logical conjunction ($\wedge$), the disjunction ($\vee$), and the negation ($\neg$) as follows:

$$p \oplus q = (p \wedge \neg q) \vee (\neg p \wedge q)$$

The exclusive disjunction $p \oplus q$ can also be expressed in the following way:

$$p \oplus q = \neg(p \wedge q) \wedge (p \vee q)$$

This representation of XOR may be found useful when constructing a circuit or network, because it has only one $\neg$ operation and small number of $\wedge$ and $\vee$ operations. The proof of this identity is given below:

$$
\begin{aligned}
p \oplus q &= (p \wedge \neg q) && \vee && (\neg p \wedge q) \\
&= ((p \wedge \neg q) \vee \neg p) && \wedge && ((p \wedge \neg q) \vee q) \\
&= ((p \vee \neg p) \wedge (\neg q \vee \neg p)) && \wedge && ((p \vee q) \wedge (\neg q \vee q)) \\
&= (\neg p \vee \neg q) && \wedge && (p \vee q) \\
&= \neg(p \wedge q) && \wedge && (p \vee q)
\end{aligned}
$$

It is sometimes useful to write $p \oplus q$ in the following way:

$$p \oplus q = \neg((p \wedge q) \vee (\neg p \wedge \neg q))$$

This equivalence can be established by applying De Morgan's laws twice to the fourth line of the above proof.

The exclusive or is also equivalent to the negation of a logical biconditional, by the rules of material implication (a material conditional is equivalent to the disjunction of the negation of its antecedent and its consequence) and material equivalence.

Relation to modern algebra [edit]

Although the operators $\wedge$ (conjunction) and $\vee$ (disjunction) are very useful in logic systems, they fail a more generalizable structure in the following way:

The systems $(\{T, F\}, \wedge)$ and $(\{T, F\}, \vee)$ are monoids. This unfortunately prevents the combination of these two systems into larger structures, such as a mathematical ring.

However, the system using exclusive or $(\{T, F\}, \oplus)$ **is** an abelian group. The combination of operators $\wedge$ and $\oplus$ over elements $\{T,F\}$ produce the well-known field F_2. This field can represent any logic obtainable with the system $(\wedge, \vee)$ and has the added benefit of the arsenal of algebraic analysis tools for fields.

Exclusive "or" in natural language [edit]

The Oxford English Dictionary explains "either … or" as follows:

> *The primary function of either, etc., is to emphasize the indifference of the two (or more) things or courses … but a secondary function is to emphasize the mutual exclusiveness, = either of the two, but not both.*

The exclusive-or explicitly states "one or the other, but not neither nor both."

Following this kind of common-sense intuition about "or", it is sometimes argued that in many natural languages, English included, the word "or" has an "exclusive" sense. The **exclusive disjunction** of a pair of propositions, (*p*, *q*), is supposed to mean that *p* is true or *q* is true, but not both. For example, it might be argued that the normal intention of a statement like "You may have coffee, or you may have tea" is to stipulate that exactly one of the conditions can be true. Certainly under many circumstances a sentence like this example should be taken as forbidding the possibility of one's accepting both options. Even so, there is good reason to suppose that this sort of sentence is not disjunctive at all. If all we know about some disjunction is that it is true overall, we cannot be sure that either of its disjuncts is true. For example, if a woman has been told that her friend is either at the snack bar or on the tennis court, she cannot validly infer that he is on the tennis court. But if her waiter tells her that she may have coffee or she may have tea, she can validly infer that she may have tea. Nothing classically thought of as a disjunction has this property. This is so even given that she might reasonably take her waiter as having denied her the possibility of having both coffee and tea.

Gonzalez v. Arizona, No. 08-17094 archived on November 4, 2010

(Note: If the waiter intends that choosing neither tea nor coffee is an option i.e. ordering nothing, the appropriate operator is NAND: p NAND q.)

In English, the construct "either ... or" is usually used to indicate exclusive or and "or" generally used for inclusive. But in Spanish, the word "o" (or) can be used in the form p o q (exclusive) or the form o p o q (inclusive). Formalists may contend that any binary or other n-ary exclusive "or" is true if and only if it has an odd number of true inputs, and there is no word in English that can conjoin a list of two or more options has this general property. For example, Barrett and Stenner contend in the 1971 article "The Myth of the Exclusive 'Or'" (Mind, 80 (317), 116–121) that no author has produced an example of an English or-sentence that appears to be false because both of its inputs are true, and brush off or-sentences such as "The light bulb is either on or off" as reflecting particular facts about the world rather than the nature of the word "or". However, the "barber paradox" – Everybody in town shaves himself or is shaved by the barber, who shaves the barber? – would not be paradoxical if "or" could not be exclusive (although a purist could say that "either" is required in the statement of the paradox).

Whether these examples can be considered "natural language" is another question. Certainly when one sees a menu stating "Lunch special: sandwich and soup or salad", one would not expect to be permitted to order both soup and salad. Nor would one expect to order neither soup nor salad, because that belies the nature of the "special", that ordering the two items together is cheaper than ordering them a la carte. Similarly, a lunch special consisting of one meat, french fries or mashed potatoes and vegetable would consist of three items, only one of which would be a form of potato. If one wanted to have meat and both kinds of potatoes, one would ask if it were possible to substitute a second order of potatoes for the vegetable. And, one would not expect to be permitted to have both types of potato and vegetable, because the result would be a vegetable plate rather than a meat plate.

Alternative symbols [edit]

The symbol used for exclusive disjunction varies from one field of application to the next, and even depends on the properties being emphasized in a given context of discussion. In addition to the abbreviation "XOR", any of the following symbols may also be seen:

- A plus sign ($+$). This makes sense mathematically because exclusive disjunction corresponds to addition modulo 2, which has the following addition table, clearly isomorphic to the one above:

Addition Modulo 2

p	q	$p+q$

0	0	0
0	1	1
1	0	1
1	1	0

- The use of the plus sign has the added advantage that all of the ordinary algebraic properties of mathematical rings and fields can be used without further ado. However, the plus sign is also used for Inclusive disjunction in some notation systems.
- A plus sign that is modified in some way, such as being encircled ($\oplus$). This usage faces the objection that this same symbol is already used in mathematics for the *direct sum* of algebraic structures.
- An inclusive disjunction symbol ($\vee$) that is modified in some way, such as being underlined ($\underline{\vee}$) or with dot above ($\dot{\vee}$).
- In several programming languages, such as C, C++, C#, Java, Perl, and Python, a caret (^) is used to denote the bitwise XOR operator. This is not used outside of programming contexts because it is too easily confused with other uses of the caret.
- The symbol ⋊, sometimes written as >< or as >-<.
- In IEC symbology, an exclusive or is marked "=1".

Properties [edit]

This section uses the following symbols:

$$\begin{aligned} 0 &= \text{false} \\ 1 &= \text{true} \\ \neg p &= \text{not } p \\ p + q &= p \text{ xor } q \\ p \wedge q &= p \text{ and } q \\ p \vee q &= p \text{ or } q \end{aligned}$$

The following equations follow from logical axioms:

$$\begin{aligned} p + 0 &= p \\ p + 1 &= \neg p \\ p + p &= 0 \\ p + \neg p &= 1 \end{aligned}$$

$$\begin{aligned} p + q &= q + p \\ p + q + p &= q \\ p + (q + r) &= (p + q) + r \\ p + q &= \neg p + \neg q \\ \neg(p + q) &= \neg p + q = p + \neg q \end{aligned}$$

$$\begin{aligned} p + (\neg p \wedge q) &= p \vee q \\ p + (p \wedge \neg q) &= p \wedge q \\ p + (p \vee q) &= \neg p \wedge q \\ \neg p + (p \vee \neg q) &= p \vee q \\ p \wedge (p + \neg q) &= p \wedge q \\ p \vee (p + q) &= p \vee q \end{aligned}$$

Gonzalez v. Arizona, No. 08-17094 archived on November 4, 2010

Associativity and commutativity [edit]

In view of the isomorphism between addition modulo 2 and exclusive disjunction, it is clear that XOR is both an associative and a commutative operation. Thus parentheses may be omitted in successive operations and the order of terms makes no difference to the result. For example, we have the following equations:

$$p + q = q + p$$

$$(p + q) + r = p + (q + r) = p + q + r$$

Other properties [edit]

- **falsehood preserving**: The interpretation under which all variables are assigned a truth value of 'false' produces a truth value of 'false' as a result of exclusive disjunction.
- linear

Computer science [edit]

Bitwise operation [edit]

Main article: Bitwise operation

Traditional symbolic representation of an XOR logic gate

Exclusive disjunction is often used for bitwise operations. Examples:

- 1 xor 1 = 0

- 1 xor 0 = 1
- 0 xor 0 = 0
- 1110 xor 1001 = 0111 (this is equivalent to addition without carry)

As noted above, since exclusive disjunction is identical to addition modulo 2, the bitwise exclusive disjunction of two *n*-bit strings is identical to the standard vector of addition in the vector space $(\mathbb{Z}/2\mathbb{Z})^n$.

In computer science, exclusive disjunction has several uses:

- It tells whether two bits are unequal.
- It is an optional bit-flipper (the deciding input chooses whether to invert the data input).
- It tells whether there is an odd number of 1 bits ($A \oplus B \oplus C \oplus D \oplus E$ is true iff an odd number of the variables are true).

In logical circuits, a simple adder can be made with an XOR gate to add the numbers, and a series of AND, OR and NOT gates to create the carry output.

On some computer architectures, it is more efficient to store a zero in a register by xor-ing the register with itself (bits xor-ed with themselves are always zero) instead of loading and storing the value zero.

In simple threshold activated neural networks, modeling the 'xor' function requires a second layer because 'xor' is not a linearly-separable function.

Exclusive-or is sometimes used as a simple mixing function in cryptography, for example, with one-time pad or Feistel network systems.

XOR is used in RAID 3–6 for creating parity information. For example, RAID can "back up" bytes `10011100` and `01101100` from two (or more) hard drives by XORing (`11110000`) and writing to another drive. Under this method, if any one of the three hard drives are lost, the lost byte can be re-created by XORing bytes from the remaining drives. If the drive containing `01101100` is lost, `10011100` and `11110000` can be XORed to recover the lost byte.

XOR is also used to detect an overflow in the result of a signed binary arithmetic operation. If the leftmost retained bit of the result is not the same as the infinite number of digits to the left, then that means overflow occurred. XORing those two bits will give a "1" if there is an overflow.

XOR can be used to swap two numeric variables in computers, using the XOR swap algorithm; however this is regarded as more of a curiosity and not encouraged in practice.

In computer graphics, XOR-based drawing methods are often used to manage such items as bounding boxes and cursors on systems without alpha channels or overlay planes.

See also

[edit]

Gonzalez v. Arizona, No. 08-17094 archived on November 4, 2010

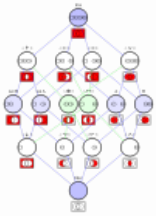
- Affirming a disjunct
- Ampheck
- Boolean algebra (logic)
- List of Boolean algebra topics
- Boolean domain
- Boolean function
- Boolean-valued function
- Controlled NOT gate
- Disjunctive syllogism
- First-order logic
- Inclusive or
- Involution
- Logical graph
- Logical value
- Minimal negation operator
- Multigrade operator
- Operation
- Parametric operator
- Parity bit
- Propositional calculus
- Symmetric difference
- XOR linked list
- XOR gate
- XOR cipher

Notes

[edit]

1. ^ See *Stanford Encyclopedia of Philosophy*, article *Disjunction*

v · d · e	**Logical connectives**
	Tautology ($\top$)
	NAND ($\uparrow$) · Converse implication ($\leftarrow$) · Implication ($\rightarrow$) · OR ($\vee$)
	Negation ($\neg$) · **XOR** ($\oplus$) · Biconditional ($\leftrightarrow$) · Statement
	NOR ($\downarrow$) · Nonimplication ($\not\rightarrow$) · Converse nonimplication ($\not\leftarrow$) · AND ($\wedge$)
	Contradiction ($\bot$)



External links

[edit]

- An example of XOR being used in cryptography

Categories: Logic | Boolean algebra | Binary operations | Propositional calculus | Logical connectives

This page was last modified on 28 October 2010 at 22:52.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.
Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Contact us

Privacy policy About Wikipedia Disclaimers