



Disjunctive normal form

In boolean logic, a **disjunctive normal form** (DNF) is a canonical normal form of a logical formula consisting of a disjunction of conjunctions; it can also be described as an **OR of ANDs**, a sum of products, or — in philosophical logic — a *cluster concept*.^[1] As a normal form, it is useful in automated theorem proving.

Definition

A logical formula is considered to be in DNF if it is a disjunction of one or more conjunctions of one or more literals.^{[2][3][4]} A DNF formula is in **full disjunctive normal form** if each of its variables appears exactly once in every conjunction and each conjunction appears at most once (up to the order of variables). As in conjunctive normal form (CNF), the only propositional operators in DNF are and (\wedge), or (\vee), and not (\neg). The *not* operator can only be used as part of a literal, which means that it can only precede a propositional variable.

The following is a context-free grammar for DNF:

1. $DNF \rightarrow (Conjunction) \vee DNF$
2. $DNF \rightarrow (Conjunction)$
3. $Conjunction \rightarrow Literal \wedge Conjunction$
4. $Conjunction \rightarrow Literal$
5. $Literal \rightarrow \neg Variable$
6. $Literal \rightarrow Variable$

Where *Variable* is any variable.

For example, all of the following formulas are in DNF:

- $(A \wedge \neg B \wedge \neg C) \vee (\neg D \wedge E \wedge F \wedge D \wedge F)$
- $(A \wedge B) \vee (C)$
- $(A \wedge B)$
- (A)

The formula $A \vee B$ is in DNF, but not in full DNF; an equivalent full-DNF version is $(A \wedge B) \vee (A \wedge \neg B) \vee (\neg A \wedge B)$.

The following formulas are **not** in DNF:^[5]

- $\neg(A \vee B)$, since an OR is nested within a NOT
- $\neg(A \wedge B) \vee C$, since an AND is nested within a NOT

- $A \vee (B \wedge (C \vee D))$, since an OR is nested within an AND

Conversion to DNF

In classical logic each propositional formula can be converted to DNF^[6] ...

... by syntactic means

The conversion involves using logical equivalences, such as double negation elimination, De Morgan's laws, and the distributive law. Formulas built from the primitive connectives $\{\wedge, \vee, \neg\}$ ^[7] can be converted to DNF by the following canonical term rewriting system:^[8]

$$\begin{aligned}
 (\neg\neg x) &\rightsquigarrow x \\
 (\neg(x \vee y)) &\rightsquigarrow ((\neg x) \wedge (\neg y)) \\
 (\neg(x \wedge y)) &\rightsquigarrow ((\neg x) \vee (\neg y)) \\
 (x \wedge (y \vee z)) &\rightsquigarrow ((x \wedge y) \vee (x \wedge z)) \\
 ((x \vee y) \wedge z) &\rightsquigarrow ((x \wedge z) \vee (y \wedge z))
 \end{aligned}$$

... by semantic means

The full DNF of a formula can be read off its truth table.^[9] For example, consider the formula

$$\phi = ((\neg(p \wedge q)) \leftrightarrow (\neg r \uparrow (p \oplus q))).^{[10]}$$

The corresponding truth table is

AB \ CD	00	01	11	10
10	1	1		
11		1	1	
01			1	1
00	1			1

Karnaugh map of the disjunctive normal form $(\neg A \wedge \neg B \wedge \neg D) \vee (\neg A \wedge B \wedge C) \vee (A \wedge B \wedge D) \vee (A \wedge \neg B \wedge \neg C)$

AB \ CD	00	01	11	10
10	1	1		
11		1	1	
01			1	1
00	1			1

Karnaugh map of the disjunctive normal form $(\neg A \wedge C \wedge \neg D) \vee (B \wedge C \wedge D) \vee (A \wedge \neg C \wedge D) \vee (\neg B \wedge \neg C \wedge \neg D)$. Despite the different grouping, the same fields contain a "1" as in the previous map.

p	q	r		(\neg	$(p \wedge q)$)	\leftrightarrow	($\neg r$	\uparrow	$(p \oplus q)$)
T	T	T			F	T		F		F	T	F	
T	T	F			F	T		F		T	T	F	
T	F	T			T	F		T		F	T	T	
T	F	F			T	F		F		T	F	T	
F	T	T			T	F		T		F	T	T	
F	T	F			T	F		F		T	F	T	
F	F	T			T	F		T		F	T	F	
F	F	F			T	F		T		T	T	F	

- The full DNF equivalent of ϕ is

$$(p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$$

- The full DNF equivalent of $\neg\phi$ is

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r)$$

Remark

A propositional formula can be represented by one and only one full DNF.^[12] In contrast, several *plain* DNFs may be possible. For example, by applying the rule $((a \wedge b) \vee (\neg a \wedge b)) \rightsquigarrow b$ three times, the full DNF of the above ϕ can be simplified to $(\neg p \wedge \neg q) \vee (\neg p \wedge r) \vee (\neg q \wedge r)$. However, there are also equivalent DNF formulas that cannot be transformed one into another by this rule, see the pictures for an example.

Disjunctive Normal Form Theorem

It is a theorem that all consistent formulas in propositional logic can be converted to disjunctive normal form.^{[13][14][15][16]} This is called the **Disjunctive Normal Form Theorem**.^{[13][14][15][16]} The formal statement is as follows:

Disjunctive Normal Form Theorem: Suppose X is a sentence in a propositional language \mathcal{L} with n sentence letters, which we shall denote by A_1, \dots, A_n . If X is not a contradiction, then it is truth-functionally equivalent to a disjunction of conjunctions of the form $\pm A_1 \wedge \dots \wedge \pm A_n$, where $+A_i = A_i$, and $-A_i = \neg A_i$.^[14]

The proof follows from the procedure given above for generating DNFs from truth tables. Formally, the proof is as follows:

Suppose X is a sentence in a propositional language whose sentence letters are A, B, C, \dots . For each row of X 's truth table, write out a corresponding conjunction $\pm A \wedge \pm B \wedge \pm C \wedge \dots$, where $\pm A$ is defined to be A if A takes the value **T** at that row, and is $\neg A$ if A takes the value **F** at that row; similarly for $\pm B, \pm C$, etc. (the alphabetical

ordering of A, B, C, \dots in the conjunctions is quite arbitrary; any other could be chosen instead). Now form the disjunction of all these conjunctions which correspond to T rows of X 's truth table. This disjunction is a sentence in $\mathcal{L}[A, B, C, \dots; \wedge, \vee, \neg]$,^[17] which by the reasoning above is truth-functionally equivalent to X . This construction obviously presupposes that X takes the value T on at least one row of its truth table; if X doesn't, i.e., if X is a contradiction, then X is equivalent to $A \wedge \neg A$, which is, of course, also a sentence in $\mathcal{L}[A, B, C, \dots; \wedge, \vee, \neg]$.^[14]

This theorem is a convenient way to derive many useful metalogical results in propositional logic, such as, trivially, the result that the set of connectives $\{\wedge, \vee, \neg\}$ is functionally complete.^[14]

Maximum number of conjunctions

Any propositional formula is built from n variables, where $n \geq 1$.

There are $2n$ possible literals: $L = \{p_1, \neg p_1, p_2, \neg p_2, \dots, p_n, \neg p_n\}$.

L has $(2^{2n} - 1)$ non-empty subsets.^[18]

This is the maximum number of conjunctions a DNF can have.^[12]

A full DNF can have up to 2^n conjunctions, one for each row of the truth table.

Example 1

Consider a formula with two variables p and q .

The longest possible DNF has $2^{(2 \times 2)} - 1 = 15$ conjunctions.^[12]

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

The longest possible full DNF has 4 conjunctions: they are underlined.

This formula is a tautology.

Example 2

Each DNF of the e.g. formula $(X_1 \vee Y_1) \wedge (X_2 \vee Y_2) \wedge \dots \wedge (X_n \vee Y_n)$ has 2^n conjunctions.

Computational complexity

The Boolean satisfiability problem on conjunctive normal form formulas is NP-complete. By the duality principle, so is the falsifiability problem on DNF formulas. Therefore, it is co-NP-hard to decide if a DNF formula is a tautology.

Conversely, a DNF formula is satisfiable if, and only if, one of its conjunctions is satisfiable. This can be decided in polynomial time simply by checking that at least one conjunction does not contain conflicting literals.

Variants

An important variation used in the study of computational complexity is *k-DNF*. A formula is in *k-DNF* if it is in DNF and each conjunction contains at most k literals.^[19]

See also

- Algebraic normal form – an XOR of AND clauses
- Blake canonical form – DNF including all prime implicants
 - Quine–McCluskey algorithm – algorithm for calculating prime implicants
- Conjunction/disjunction duality
- Propositional logic
- Truth table

Notes

1. Post 1921.
2. Davey & Priestley 1990, p. 153.
3. Gries & Schneider 1993, p. 67.
4. Whitesitt 2012, pp. 33–37.
5. However, they are in negation normal form.
6. Davey & Priestley 1990, p. 152-153.
7. Formulas with other connectives can be brought into negation normal form first.
8. Dershowitz & Jouannaud 1990, p. 270, Sect.5.1.
9. Sobolev 2020.
10. $\phi = ((\text{NOT } (p \text{ AND } q)) \text{ IFF } ((\text{NOT } r) \text{ NAND } (p \text{ XOR } q)))$
11. like $(a \wedge b) \vee (b \wedge a) \vee (a \wedge b \wedge b)$
12. It is assumed that repetitions and variations^[11] based on the commutativity and associativity of \vee and \wedge do not occur.
13. Halbeisen, Lorenz; Kraph, Regula (2020). *Gödel's theorems and zermelo's axioms: a firm foundation of mathematics*. Cham: Birkhäuser. p. 27. ISBN 978-3-030-52279-7.
14. Howson, Colin (1997). *Logic with trees: an introduction to symbolic logic*. London ; New York: Routledge. p. 41. ISBN 978-0-415-13342-5.
15. Cenzer, Douglas; Larson, Jean; Porter, Christopher; Zapletal, Jindřich (2020). *Set theory and foundations of mathematics: an introduction to mathematical logic*. New Jersey: World Scientific. pp. 19–21. ISBN 978-981-12-0192-9.

16. Halvorson, Hans (2020). *How logic works: a user's guide*. Princeton Oxford: Princeton University Press. p. 195. ISBN 978-0-691-18222-3.
17. That is, the language with the propositional variables A, B, C, \dots and the connectives $\{\wedge, \vee, \neg\}$.
18. $|\mathcal{P}(L)| = 2^{2^n}$
19. Arora & Barak 2009.

References

- Arora, Sanjeev; Barak, Boaz (20 April 2009). *Computational Complexity: A Modern Approach*. Cambridge University Press. p. 579. doi:10.1017/CBO9780511804090 (<https://doi.org/10.1017%2FCBO9780511804090>). ISBN 9780521424264.
 - Davey, B.A.; Priestley, H.A. (1990). *Introduction to Lattices and Order*. Cambridge Mathematical Textbooks. Cambridge University Press.
 - Dershowitz, Nachum; Jouannaud, Jean-Pierre (1990). "Rewrite Systems". In Van Leeuwen, Jan (ed.). *Formal Models and Semantics*. Handbook of Theoretical Computer Science. Vol. B. Elsevier. pp. 243–320. ISBN 0-444-88074-7.
 - Gries, David; Schneider, Fred B. (22 October 1993). *A Logical Approach to Discrete Math* (<https://books.google.com/books?id=ZWTDQ6H6gsUC&q=%22disjunctive+normal+form%22&pg=PA67>). Springer Science & Business Media. ISBN 978-0-387-94115-8.
 - Hilbert, David; Ackermann, Wilhelm (1999). *Principles of Mathematical Logic* (<https://books.google.com/books?id=45ZGMjV9vfcC&q=%22disjunctive+normal+form%22>). American Mathematical Soc. ISBN 978-0-8218-2024-7.
 - Howson, Colin (11 October 2005) [1997]. *Logic with trees: an introduction to symbolic logic* (<https://books.google.com/books?id=Y4WGAqAAQBAJ&q=%22disjunctive+normal+form%22>). Routledge. ISBN 978-1-134-78550-6.
 - Post, Emil (July 1921). "Introduction to a General Theory of Elementary Propositions". *American Journal of Mathematics*. **43** (3): 163–185. doi:10.2307/2370324 (<https://doi.org/10.2307%2F2370324>). JSTOR 2370324 (<https://www.jstor.org/stable/2370324>).
 - Sobolev, S.K. (2020) [1994], "Disjunctive normal form" (https://www.encyclopediaofmath.org/index.php?title=Disjunctive_normal_form&oldid=54535), *Encyclopedia of Mathematics*, EMS Press
 - Whitesitt, J. Eldon (24 May 2012) [1961]. *Boolean Algebra and Its Applications* (<https://books.google.com/books?id=20Un1T78GIMC&q=%22disjunctive+normal+form%22>). Courier Corporation. ISBN 978-0-486-15816-7.
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Disjunctive_normal_form&oldid=1235215860"