Disjunctive Normal Form (DNF)

What Does Disjunctive Normal Form (DNF) Mean?

Disjunctive normal form (DNF) is the normalization of a logical formula in Boolean mathematics. In other words, a logical formula is said to be in disjunctive normal form if it is a disjunction of conjunctions with every variable and its negation is present once in each conjunction. All disjunctive normal forms are non-unique, as all disjunctive normal forms for the same proposition are mutually equivalent.

Explains Disjunctive Normal Form (DNF)

A logical formula is in disjunctive normal form if and only if there is an existence of alternation of one or more conjunctions of one or more literals. A formula is considered as in full disjunctive normal form if all the variables involved are represented only once in every clause. Similar to conjunctive normal form, the propositional operators in disjunctive normal form are same: AND, OR and NOT.

All logical formulas can be converted into an equivalent disjunctive normal form. However, in some cases, exponential explosion of the logical function is possible due to conversion to disjunctive normal form. Another salient point is that any unique Boolean function can be represented by only one and a unique full disjunctive normal form. With the help of techniques such as the truth table method, truth trees or a table of logical equivalences, disjunctive normal form for logical formulas can be generated. K-DNF, a variation of disjunctive normal form, is widely used and popular in the study of computational complexity.

Conjunctive Normal Form (CNF)

What Does Conjunctive Normal Form (CNF) Mean?

Conjunctive normal form (CNF) is an approach to Boolean logic that expresses formulas as conjunctions of clauses with an AND or OR. Each clause connected by a conjunction, or AND, must be either a literal or contain a disjunction, or OR operator. CNF is useful for automated theorem proving.

Explains Conjunctive Normal Form (CNF)

In conjunctive normal form, statements in Boolean logic are conjunctions of clauses with clauses of disjunctions. In other words, a statement is a series of ORs connected by ANDs.

For example:

(A OR B) AND (C OR D)

(A OR B) AND (NOT C OR B)

The clauses may also be literals:

A OR B

A AND B

Literals are seen in CNF as conjunctions of literal clauses and conjunctions that happen to have a single clause. It is possible to convert statements into CNF that are written in another form, such as disjunctive normal form.

PDNF:

It stands for Principal Disjunctive Normal Form. It refers to the Sum of Products, i.e., SOP. For eg. : If P, Q, R are the variables then (P . Q' . R) + (P' . Q . R) + (P . Q . R') is an example of an expression in PDNF. Here '+' i.e. sum is the main operator. The Key difference between PDNF and DNF is that in case of DNF, it is not necessary that the length of all the variables in the expression is same.

For eg.: 1. (P . Q' . R) + (P' . Q . R) + (P . Q) is an example of an expression in DNF but not in PDNF.

2. (P . Q' . R) + (P' . Q . R) + (P . Q . R') is an example of an expression which is both in PDNF and DNF.

PCNF: It stands for Principal Conjunctive Normal Form. It refers to the Product of Sums, i.e., POS.

For eg. : If P, Q, R are the variables then (P + Q’+ R).(P’+ Q + R).(P + Q + R’) is an example of an expression in PCNF. Here ‘.’ i.e. product is the main operator Here also, the Key difference between PCNF and CNF is that in case of CNF, it is not necessary that the length of all the variables in the expression is same .

For eg.: 1. (P + Q’+ R).(P’+ Q + R).(P + Q) is an example of an expression in CNF but not in PCNF.

2. (P + Q’+ R).(P’+ Q + R).(P + Q + R’) is an example of an expression which is both in PCNF and CNF. Properties of PCNF and PDNF:

1. Every PDNF or PCNF corresponds to a unique Boolean Expression and vice versa.

2. If X and Y are two Boolean expressions then, X is equivalent to Y if and only if PDNF(X) = PDNF(Y) or PCNF(X) = PCNF(Y).

3. For a Boolean Expression, if PCNF has m terms and PDNF has n terms, then the number of variables in such a Boolean expression= $\log_2(m+n)$.