**Disjunctive and Conjunctive Normal Forms**

Every truth table (Boolean function) can be written as either a conjunctive normal form (CNF) or disjunctive normal form (DNF)

A Boolean expression is in disjunctive normal form (DNF) if:

1. the variables within each term are **AND**ed together,
2. the terms are **OR**ed together, and
3. every variable or its complement is represented in every term (i.e. either A or ~A is in each term, B or ~B is in each term, etc.).
4. No parentheses or other Boolean operations appear in the expression.

A propositional formula is in disjunctive normal form (DNF) if it is the disjunction of conjunctions of literals.

* A literal is a Boolean variable, or the negation of a Boolean variable (e.g., P,¬P).
* A conjunction is a logical formula that is the AND (∧) of (smaller) formulas (e.g., P∧¬Q∧R).
* A disjunction is a logical formula that is the OR (∨) of (smaller) formulas (e.g., P∨¬Q∨R).

With this in mind, the meaning of DNF should be clearer: a DNF formula is an OR of AND’s. For example,

X = (A∧¬B∧¬D)∨(B∧C)∨(C∧¬D∧E)

is a DNF formula: the literals of X are A,¬B,¬D,B,C,¬D, and E; the conjunctions are A∧¬B∧ ¬D,B∧C, and C∧¬D∧E; and X is the disjunction (OR) of these three conjunctions. On the other hand,

Y = (A∨¬B)∧(B∨C∨¬D)

is not a DNF formula, because the structure of the operators is inverted. But this formula is in the other “standard” form (CNF) for propositional formulas.

The rules for conjunctive normal form (CNF) are similar to the rules for DNF except for the precedence of the ANDs and ORs. A Boolean expression is in CNF if:

1. the variables within each term are **OR**ed together,
2. the terms are **AND**ed together, and
3. every variable or its complement is represented in every term (i.e. either A or ~A is in each term, B or ~B is in each term, etc.).
4. No parenthesis - other than those separating the terms - or other Boolean operations appear in the expression.

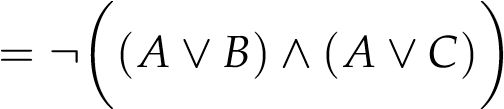
A propositional formula is in conjunctive normal form (CNF) if it is the conjunction of disjunctions of literals.

As noted above Y is a CNF formula because it is an AND of OR’s. The literals of Y are A,¬B, B, C, and ¬D; the disjunctions are (A∨¬B) and (B∨C∨¬D); and Y is the conjunction (AND) of the two disjunctions.

Any propositional formula is tautologically equivalent to some formula in disjunctive and conjunctive normal form.

**Converting to DNF**

We now describe a procedure that converts any propositional formula *X* into disjunctive normal form. Note that the resulting DNF is not necessarily the shortest expression equivalent to *X* (or even the shortest DNF equivalent to *X*), but the result is guaranteed to be a DNF formula. We illustrate the procedure on the following example:

*X* 

The first step is to write the truth table of *X*:

|  |  |  |  |
| --- | --- | --- | --- |
| *A* | *B* | *C* | *X* |
| **T** | **T** | **T** | **F** |
| **T** | **T** | **F** | **F** |
| **T** | **F** | **T** | **F** |
| **T** | **F** | **F** | **F** |
| **F** | **T** | **T** | **F** |
| **F** | **T** | **F** | **T** |
| **F** | **F** | **T** | **T** |
| **F** | **F** | **F** | **T** |

(Note that if *A* = **T**, then the ∧ “succeeds”, so *X* = **F**. Otherwise, the ∧ “succeeds” only if *B* = *C* = **T**, and so *X* = **F** in this case and *X* = **T** in all remaining rows.)

The second step is to identify the rows that contain *X* = **T**; in our case, these are rows 6, 7, and 8. Finally, we encode each row as a conjunction of the corresponding literals, and the final result is the disjunction of these conjunctions. Intuitively, we can think of the DNF we’re constructing as the following expression:

(row 6)∨(row 7)∨(row 8).

In our example, we have the following encoding:

**row 6 : (¬*A* ∧ *B* ∧¬*C*)**

**row 7 : (¬*A* ∧¬*B* ∧ *C*)**

**row 8 : (¬*A* ∧¬*B* ∧¬*C*).**

Taking the disjunction of these conjunctions yields the following DNF formula for *X*:

(¬*A* ∧ *B* ∧¬*C*)∨(¬*A* ∧¬*B* ∧ *C*)∨(¬*A* ∧¬*B* ∧¬*C*).

**Converting to CNF**

Similarly, we can convert any propositional formula into conjunctive normal form. We will illustrate the process on the same propositional formula X from the previous subsection. In this case, the intuition is the following: X = T if the truth assignment does not belong to any of the rows that evaluate to F.

**NOT row 1 : ¬(*A* ∧ *B* ∧ *C*) = ¬*A* ∨¬*B* ∨¬*C***

**NOT row 2 : ¬(*A* ∧ *B* ∧¬*C*) = ¬*A* ∨¬*B* ∨ *C***

**NOT row 3 : ¬(*A* ∧¬*B* ∧ *C*) = ¬*A* ∨ *B* ∨¬*C***

**NOT row 4 : ¬(*A* ∧¬*B* ∧¬*C*) = ¬*A* ∨ *B* ∨ *C***

**NOT row 5 : ¬(¬*A* ∧ *B* ∧ *C*) = *A* ∨¬*B* ∨¬*C*.**

Notice that we’ve applied De Morgan’s Law and the double negation rule to produce a disjunction.

The final CNF is the conjunction of these disjunctions:

(¬*A* ∨¬*B* ∨¬*C*)∧(¬*A* ∨¬*B* ∨ *C*)∧(¬*A* ∨ *B* ∨¬*C*)∧(¬*A* ∨ *B* ∨ *C*)∧(*A* ∨¬*B* ∨¬*C*).

**Proving Equivalence of Formulas**

A typical problem is the following: given two propositional formulas X and Y, are X and Y equivalent? That is, what is the truth value of the proposition X ↔ Y (denoted by X = Y)? Roughly speaking, there are three ways to prove that X and Y are equivalent:

1. Use a truth table to check if the columns corresponding to X and Y are identical.
2. Apply a sequence of rules to X until Y appears. (We can also “work backwards” from Y and meet X at some formula “in the middle.”)
3. Convert X and Y to DNF (or CNF) using the procedure described above, and check if the resulting formulas are identical. This is essentially a special case of Method 2.