

First, we can confine attention to formulae in which the negation sign is always immediately followed by a sentence-letter, so that no other truth-functor ever occurs in the scope of a negation sign. This is because every formula in our language is logically equivalent to a formula obeying this condition. To establish this point, we need only consider the laws

$$\begin{array}{lll}
 \neg \top & \models & \perp \\
 \neg \perp & \models & \top \\
 \neg \neg \phi & \models & \phi \\
 \neg(\phi \wedge \psi) & \models & \neg \phi \vee \neg \psi \\
 \neg(\phi \vee \psi) & \models & \neg \phi \wedge \neg \psi.
 \end{array}$$

(The first two have no special name; the third is of course the law of double negation; the last two are called De Morgan's laws, after the logician Augustus de Morgan. You should check with a truth-table test that these laws are indeed correct.) Now consider any occurrence of the negation sign that does have some truth-functor in its scope. This means that it must be followed either by $\top$ or by $\perp$ or by another negation sign or by a bracket, introducing a formula with either $\wedge$ or $\vee$ as its main functor. In the first three cases that occurrence of the negation sign can simply be deleted, at the same time interchanging $\top$ and $\perp$ in the first two cases, and deleting the other negation sign in the third case. In the remaining two cases we apply De Morgan's laws, exchanging the one negation sign outside the bracket for two that are inside it, at the same time changing the $\wedge$ to $\vee$, or the $\vee$ to $\wedge$, whichever is appropriate. Then we look again at the two negation signs that result. Either they are now followed immediately by sentence-letters, as desired, or if they are not, then we apply the whole procedure once more. And we continue to do this as often as is needed to bring *every* negation sign into the required position.

It is clear that we must eventually reach the result desired. For in the first three cases the negation sign disappears altogether, and in the other two the negation sign is exchanged for two others, each of which has *fewer* truth-functors in its scope (for there must be fewer truth-functors in ϕ than there are in $\phi \wedge \psi$, and similarly with ψ). Finally, note that the formula we end with, after making all these transformations, must still be logically equivalent to the formula we began with. This, of course, is because each transformation exchanges one subformula for another that is logically equivalent to it, and we noted at the end of the last section that such an interchange of logically equivalent subformulae must preserve the equivalence of the whole.

Having got the negation signs into the right position, we can now carry out some further transformations. In particular we can rearrange the

occurrences of $\wedge$ and $\vee$ so that no conjunction sign has any disjunction in its scope, or alternatively so that no disjunction sign has any conjunction in its scope. The first is called disjunctive normal form (DNF), and the second is conjunctive normal form (CNF). These transformations rely on the laws of *distribution*, which are

$$\begin{aligned}\phi \wedge (\psi \vee \chi) & \models (\phi \wedge \psi) \vee (\phi \wedge \chi). \\ (\phi \vee \psi) \wedge \chi & \models (\phi \wedge \chi) \vee (\psi \wedge \chi). \\ \phi \vee (\psi \wedge \chi) & \models (\phi \vee \psi) \wedge (\phi \vee \chi). \\ (\phi \wedge \psi) \vee \chi & \models (\phi \vee \chi) \wedge (\psi \vee \chi).\end{aligned}$$

One uses the first pair to obtain DNF, and the second pair to obtain CNF. Let us concentrate just on the first, where our object is to ensure that no conjunction sign has any disjunction in its scope. If any does, then there must be somewhere a conjunction sign that *immediately* governs a disjunction, i.e. that conjoins two formulae at least one of which is a disjunction, as in the formulae on the left-hand side of the first pair of distribution laws. So we replace this by the right-hand side. The result is that the conjunction sign is split into two, but each of these has *fewer* disjunction signs in its scope than the one we began with. So if we continue the procedure we must eventually reach the desired position, where there are *no* disjunction signs in the scope of any conjunction signs. And, as before, these transformations simply interchange logically equivalent subformulae, so the formula that we end with must be logically equivalent to the one that we began from. (The argument in the case of CNF is exactly similar.)

At this point we may delete the superfluous brackets from a continued conjunction, and from a continued disjunction (p. 22), and our whole formula in DNF looks like this. It is as a whole a disjunction of one or more disjuncts, where each of the disjuncts is a conjunction of one or more conjuncts, and where each of the conjuncts is either an atomic formula—i.e. a sentence-letter or $\top$ or $\perp$ —or the negation of an atomic formula. Let us write $\pm P_i$ to indicate that there may or may not be a negation sign in front of the atomic formula P_i . Then the whole formula takes this form

$$(\pm P_i \wedge \pm P_j \wedge \dots) \vee (\pm P_k \wedge \pm P_l \wedge \dots) \vee \dots$$

A formula in CNF has the corresponding structure with $\wedge$ and $\vee$ everywhere interchanged, i.e.

$$(\pm P_i \vee \pm P_j \vee \dots) \wedge (\pm P_k \vee \pm P_l \vee \dots) \wedge \dots$$

But it should be noted that in this description of the structure of formulae in DNF and CNF we have made use of the degenerate case of a ‘disjunction’

which has only one 'disjunct', and similarly of a 'conjunction' which has only one 'conjunct'.

It follows that some formulae have both structures simultaneously, i.e. they are both in DNF and in CNF. The simplest example is a formula which is a single sentence-letter

P .

This formula is a disjunction of only one disjunct, where the disjunct is itself a conjunction of only one conjunct, namely P . So it is in DNF. Of course, we can say the same thing the other way round too. This formula is a conjunction of only one conjunct, where the conjunct is itself a disjunction of only one disjunct, namely P . So it is equally in CNF. Or one can make this point in the terminology with which I began: in this formula, no occurrence of $\neg$ has any truth-functor in its scope; no occurrence of $\wedge$ has any $\vee$ in its scope; and no occurrence of $\vee$ has any $\wedge$ in its scope. You should verify in a similar way that both of the following formulae are also both in DNF and in CNF:

$$\begin{aligned} P \wedge \neg Q \wedge R \\ P \vee \neg Q \vee R. \end{aligned}$$

By contrast, of the following formulae the first is in DNF but not CNF, and the second is in CNF but not DNF:

$$\begin{aligned} P \vee (Q \wedge \neg R) \\ (P \vee Q) \wedge (P \vee \neg R). \end{aligned}$$

Finally, the following formulae are neither in DNF nor in CNF:

$$\begin{aligned} \neg(P \wedge Q) \\ (P \wedge (Q \vee \neg R)) \vee \neg Q. \end{aligned}$$

The normal forms DNF and CNF, as characterized so far, can still be very untidy. To introduce a greater tidiness we may start by rearranging each conjunction, in a case of DNF, or each disjunction, in a case of CNF, so that in each one the sentence-letters occur in alphabetical order. In view of the laws of *commutativity* and *associativity*, namely

$$\begin{array}{llll} \phi \wedge \psi & \models & \psi \wedge \phi & \phi \vee \psi & \models & \psi \vee \phi \\ \phi \wedge (\psi \wedge \chi) & \models & (\phi \wedge \psi) \wedge \chi & \phi \vee (\psi \vee \chi) & \models & (\phi \vee \psi) \vee \chi, \end{array}$$

the order in which the conjuncts or disjuncts are written makes no difference. This will bring to light any repetitions, where the same sentence-letter occurs twice, and these may then be deleted in view of the laws of *idempotence*, namely

$$\phi \wedge \phi \models \phi \quad \phi \vee \phi \models \phi.$$

It will also bring to light any case in which we have both a sentence-letter and its own negation. These can then be simplified by the following laws of *contradiction* and *excluded middle*:

$$\phi \wedge \neg \phi \models \perp \quad \phi \vee \neg \phi \models \top.$$

The occurrences of $\top$ and $\perp$ thus introduced, and any other occurrences of $\top$ and $\perp$ that there may be in our formula can then be eliminated (except in one special case) by applying the laws

$$\begin{array}{ll} \perp \wedge \phi \models \perp & \top \vee \phi \models \top \\ \perp \vee \phi \models \phi & \top \wedge \phi \models \phi. \end{array}$$

For example, suppose that we have a formula in DNF, and one of the disjuncts contains the conjunction $P_i \wedge \neg P_i$. We begin by replacing this by $\perp$, using the law of contradiction. Then, if this is part of a longer conjunction, we replace the whole conjunction by $\perp$, using the law next cited in the left column. Thus $\perp$ becomes one of the disjuncts in our overall disjunction. If there are any other disjuncts, then this disjunct can simply be eliminated, using the final law cited in the left column. If there are no other disjuncts, our whole formula has already been reduced just to $\perp$, and we make no further simplifications. Generalizing upon this, it is easy to see that any occurrence of $\perp$ in a formula in DNF either can be eliminated or can replace the whole formula. Using the laws cited in the right column, the same holds for any occurrence of $\top$. And, using both columns together, the same result holds for formulae in CNF just as well as for formulae in DNF. We can always assume, then, that our formulae do not contain either $\top$ or $\perp$, except in the special case when $\top$ or $\perp$ is the whole formula.

The final ‘tidying up’ operation is to transform the formula so that each disjunct, in a case of DNF, or each conjunct, in a case of CNF, contains exactly the same sentence-letters. This is done by applying the laws of *elaboration*:

$$\phi \models (\phi \wedge \psi) \vee (\phi \wedge \neg \psi) \quad \phi \models (\phi \vee \psi) \wedge (\phi \vee \neg \psi)$$

For example, suppose we have a formula in DNF. We begin by listing all the sentence-letters that occur anywhere in that formula, and we check to see whether each disjunct contains one occurrence of each of them. If we find a disjunct ϕ which lacks some letter, say P_j , then we use the law of elaboration in the left column to replace ϕ by $(\phi \wedge P_j) \vee (\phi \wedge \neg P_j)$. (Each of the new disjuncts $\phi \wedge P_j$ and $\phi \wedge \neg P_j$ is then rearranged alphabetically.) By repeating this

step as often as necessary, we can evidently expand each conjunction to the desired length. Then we make one final step of simplification: if it now turns out that any disjunct is repeated, we delete the repetition as superfluous (by the law of idempotence), and we have now reached our goal. Our formula is now in *perfect DNF*, which is to say: it is in DNF; no disjunct is superfluous; each disjunct contains the same sentence-letters; no disjunct contains the same letter twice; the truth-functors $\top$ and $\perp$ do not occur, except in the special case where one of them is the whole formula. (*Perfect CNF* is defined similarly, with 'conjunct' for 'disjunct' throughout.)

The method just explained is a method for transforming any formula, step by step, into a logically equivalent formula in perfect DNF, or in perfect CNF, whichever is desired. The method relies throughout on the principle that logically equivalent subformulae may be substituted for one another and the results will also be logically equivalent. It has some quite interesting applications, as will be shown later (Exercises 6.2.2 and 6.5.1). But in practice it can be very tedious indeed, so it is relevant that there is an alternative method that is often quicker. This involves (a) constructing the full truth-table for the formula in question, and then (b) simply reading the desired normal form off the truth-table. I give first an example to show how this is done, and afterwards a general description of the technique.

Suppose we have a formula $\phi(P,Q,R)$, containing just the sentence-letters P, Q, R , and we wish to find for it a logically equivalent formula in perfect DNF. We first calculate its truth-table, which is, say, this

P	Q	R	$\phi(P,Q,R)$
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T

We look down the truth-table to see in which rows the formula is true, i.e. in this case rows 1, 4, 5, 8, and we ignore the others. For each such row we then form a conjunction of the sentence-letters or their negations which corresponds to the truth-values assigned to the letters in that row in this way: if the letter P_i is assigned T in that row, we write P_i , and if it is assigned F, then we write $\neg P_i$. So in the present case we write down four such conjunctions, namely

$$\begin{aligned}
&P \wedge Q \wedge R \\
&P \wedge \neg Q \wedge \neg R \\
&\neg P \wedge Q \wedge R \\
&\neg P \wedge \neg Q \wedge \neg R.
\end{aligned}$$

The point of this is that each formula we have written down is true if and only if the letters are assigned the values in the corresponding rows of the truth-table. Now our whole formula $\phi(P, Q, R)$ is true if and only if the letters are assigned values as in one or other of the four rows being considered, so it is true if and only if one or other of the four conjunctions that we have written down is true, i.e. if and only if the disjunction of those four conjunctions is true. So finally we write down the disjunction of those four conjunctions, namely

$$(P \wedge Q \wedge R) \vee (P \wedge \neg Q \wedge \neg R) \vee (\neg P \wedge Q \wedge R) \vee (\neg P \wedge \neg Q \wedge \neg R).$$

This is the formula desired. As we have argued, it is logically equivalent to the formula we began with, since it must have the same truth-table, and it is in perfect DNF.

Here is a general statement of the method. Consider any formula ϕ with sentence-letters $P_1, \dots, P_n$. List the interpretations of the sentence-letters under which the whole formula is true. If there are none, then simply write $\perp$, for this is a formula in perfect DNF that is logically equivalent to ϕ . If there are some, then form for each the corresponding conjunction $\pm P_1 \wedge \pm P_2 \wedge \dots \wedge \pm P_n$, where $\pm P_i$ is P_i if P_i is true in that interpretation, and $\neg P_i$ otherwise. The disjunction of all these conjunctions is then the formula required.

The method is easily adapted to the case of CNF. Here what we need to do is to consider the interpretations in which our formula is false. If there are none, then we simply write $\top$, and the task is done. If there are some, then again we form the corresponding conjunctions for each. For example, in the case of the formula $\phi(P, Q, R)$ considered earlier there are four such conjunctions, namely

$$\begin{aligned}
&P \wedge Q \wedge \neg R \\
&P \wedge \neg Q \wedge R \\
&\neg P \wedge Q \wedge \neg R \\
&\neg P \wedge \neg Q \wedge R.
\end{aligned}$$

Now our formula $\phi(P, Q, R)$ is false iff one of these conjunctions is true, and hence it is true iff all of these conjunctions are false. We therefore form a formula which says that they are false, namely by conjoining the negations of them:

$$\neg(P \wedge Q \wedge \neg R) \wedge \neg(P \wedge \neg Q \wedge R) \wedge \neg(\neg P \wedge Q \wedge \neg R) \wedge \neg(\neg P \wedge \neg Q \wedge R).$$

This is logically equivalent to our original formula. Moreover, it is easily transformed into a formula in perfect CNF, simply by applying De Morgan's laws (and double negation) to obtain

$$(\neg P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee \neg R) \wedge (P \vee \neg Q \vee R) \wedge (P \vee Q \vee \neg R).$$

Thus for every formula there is a logically equivalent formula in perfect DNF, and there is also a logically equivalent formula in perfect CNF, and we have now proved this result in two ways, first by showing how to transform any formula that is not already in the required form into one that is, by systematically exchanging its subformulae for others that are logically equivalent; and second by showing how to read the required normal form off the truth-table. The first method yielded only a limited result, for we began by restricting attention to a language that contained only the truth-functors $\top, \perp, \neg, \wedge, \vee$. To remove the limitation, we should have to show that any truth-functor whatever is equivalent to one that can be expressed in this language. But that is just what the second method does. For any truth-functor will have a truth-table, and the second method shows us how to find an expression in our preferred language that has that same truth-table. It is the second method, therefore, that yields the more significant result. The next section will develop this further.

EXERCISES

2.6.1. Find perfect disjunctive normal forms for the following formulae, first by the method of successive transformations, and then by constructing a truth-table and reading the result off that

- (a) $\neg((P \wedge (\neg P \vee \perp)) \vee P)$
- (b) $(P \vee \neg Q) \wedge (Q \vee \neg P)$
- (c) $(P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (\neg P \vee \neg Q).$

Note that to deal with the last example you will need to restore the missing brackets; restore them in whatever way seems likely to abbreviate labour. (You may wish to experiment with this.) Note also that when applying the method of successive transformations there is no need to follow the order used in the previous section. We know that if that order is followed then the desired result must be reached in the end, but a different order may well get there more quickly. (You may wish to experiment on this point with example (a).)