Section 4.3. Boolean functions

Let us take another look at the simplest non-trivial Boolean algebra, $\mathcal{P}(\{0\})$, the power-set algebra based on a one-element set, chosen here as $\{0\}$. This has two elements, the empty set $\emptyset$, which is $\bot$ (bottom), and the set $\{0\}$, which is $\top$. Let us write $\mathbf{2}$ for this algebra; thus, $\mathbf{2} = \{\bot,\top\}$, with the total ordering given by $\bot < \top$. It is obvious, either because we are having a two-element total ordering, or because we are having an algebra of sets, that the Boolean operations are as follows:

$$\top \wedge \top = \top \qquad \top \wedge \bot = \bot$$

$$\bot \wedge \top = \bot \qquad \bot \wedge \bot = \bot$$

$$\top \vee \top = \top \qquad \top \vee \bot = \top$$

$$\bot \wedge \top = \top \qquad \bot \wedge \bot = \bot$$

$$-\top = \bot \qquad -\bot = \top .$$

We read $\top$ as **true**, $\bot$ as **false**; we read the operation $\wedge$ as **and**, $\vee$ as **or**, $-$ as **not**. In this way, the two-element Boolean algebra becomes an algebra of *truth-values*, and becomes the basis of *propositional logic*. In propositional logic, we analyze sentences into constituent parts out of which the sentence is built up using the *connectives*: $\wedge$ (conjunction; "and"), $\vee$ (disjunction; "or"), $\neg$ (negation; "not"; the difference to the "minus" sign, $-$, is inessential), and two more: $\longrightarrow$ (conditional; "if ..., then ...") and $\longleftrightarrow$ (biconditional; "if and only if").

We also call a sentence of the form $A \wedge B$ a *conjunction*, its terms A and B the *conjuncts* in the sentence. As indicated in the previous paragraph, the operation of conjunction is the one that forms $A \wedge B$ out of A, B. $A \vee B$ is a *disjunction*; A and B are its *disjuncts*. $A \longrightarrow B$ is a *conditional*; A is its *antecedent*, B its *succedent*. $A \longleftrightarrow B$ is a *biconditional*.

Consider the following sentences:

" n is divisible by 2, or n is divisible by 3."

" n is divisible by 2, and n is divisible by 3."

"If the greatest common divisor of n and 6 is not 1, then n is divisible by 2, or n is divisible 3."

" n is divisible by 6 if and only if n is divisible by 2 and n is divisible by 3."

By denoting the sentence " n is divisible by 2 " by A;

$$A \underset{\text{def}}{\equiv} \text{" } n \text{ is divisible by } 2 \text{ ";}$$

also

$$B \underset{\text{def}}{\equiv} \text{" } n \text{ is divisible by } 3 \text{ ",}$$

$$C \equiv \text{" } n \text{ is divisible by } 6 \text{ ",}$$

$$D \equiv \text{" The greatest common divisor of } n \text{ and } 6 \text{ is } 1 \text{ ",}$$

the above sentences may be analyzed, respectively, as

$$E \underset{\text{def}}{\equiv} A \wedge B,$$

$$F \underset{\text{def}}{\equiv} A \vee B,$$

$$G \underset{\text{def}}{\equiv} (\neg D) \longrightarrow (A \vee B),$$

$$H \underset{\text{def}}{\equiv} C \longleftrightarrow (A \wedge B).$$

The truth or falsity of the last four composite sentences depend on the truth-values of their constituents A, B, C and D. (Of course, the truth-value of each of A, B, C and D depend on the value of the n, which we assume to be a fixed, but unspecified, natural number.)

The dependence of the truth-value of E is exactly according to the truth-table given above describing the effect of the operation of $\wedge$ (conjunction) on the two truth-values. That is to say, if A and B are both **true**, so is $E \equiv A \wedge B$; in any other of the three cases concerning the values of A and B: $(\top, \bot)$, $(\bot, \top)$, and $(\top, \bot)$, the value of $A \wedge B$ is **false**. This corresponds to the ordinary use of the connective "and".

The truth-value of $F \equiv A \vee B$ is computed according to the truth-table for $\vee$ (disjunction) given above. E.g., if $n = 6$, or if $n = 2$, or if $n = 3$, $A \vee B$ is **true**; in fact, according to the first three lines of that table, in the given order. However, if $n = 1$, then $A \vee B$ is **false**; this corresponds to the last line of the table. Notice that disjunction as we are describing it here is *non-exclusive* "*or*" ; a disjunction is **true** if, in particular, both disjuncts are **true**. The sentence in question is, in more explicit form,

"Either n is divisible by 2, or n is divisible by 3, or both."

Let us note that in mathematics, "or" (disjunction) is *always* intended as non-exclusive "or". (With exclusive "or" , a disjunction would be true just in case *precisely one* disjunct is true.) This may be seen e.g. on the sentence G that is regarded as being true, no matter what n is. If $n = 6$, then the succedent of the conditional, $A \vee B$ is **true** under the non-exclusive interpretation, but not under the exclusive one.

The connective of negation as used in mathematical language, clearly corresponds to the table given for it above. If $n = 5$, then D (with D the sentence denoted by D above) is **true**, and $\neg D$ is **false**; if $n = 2$, then D is **false**, and $\neg D$ is **true**.

The connective of the conditional also corresponds to an operation in the two-element algebra $\mathbf{2}$ as follows:

$$\top \longrightarrow \top = \top, \qquad \top \longrightarrow \bot = \bot,$$

$$\bot \longrightarrow \top = \top, \qquad \bot \longrightarrow \bot = \top.$$

This table says that a conditional is **true** unless the antecedent is **true**, and the succedent is **false**. In particular, the conditional is **true** whenever the antecedent is **false**, independently of the truth-value of the succedent: "false implies everything".

We may verify that this corresponds to the usual mathematical use by considering that the sentence

$$I \stackrel{\text{def}}{=} C \longrightarrow A\,,$$

that is,

"If n is divisible by 6, then n is divisible by 2",

should be true no matter what the value of n is. If $n = 6$, $n = 2$, $n = 1$, we obtain the sentences

"If 6 is divisible by 6, then 6 is divisible by 2",

"If 2 is divisible by 6, then 2 is divisible by 2",

"If 1 is divisible by 6, then 1 is divisible by 2".

These are of the respective forms

$$\top \longrightarrow \top\,, \qquad \bot \longrightarrow \top\,, \qquad \bot \longrightarrow \bot\,.$$

As said, ordinary mathematical usage attributes the value **true** to these forms, in agreement with the table for the conditional above.

The fact that a conditional is **true** once the antecedent is **false** is also reflected in the general approach to the proof of a conditional, which is that we start by *assuming* that the antecedent is **true**. In fact, we may just as well do so, since if the antecedent is **false**, the whole conditional is automatically **true**, and we can rest in our task of proving the conditional to be **true**.

The conditional can be expressed in terms of negation and disjunction:

$$x \rightarrow y = (-x) \vee y \qquad (1)$$

is an identity true for any values of x and y in $\mathbf{2}$ (verify!). Thus, in principle, the conditional could be dispensed with; sentence G may be paraphrased as

"Either the greatest common divisor of n and 6 is 1, or n is divisible by 2, or n is divisible by 3."

The biconditional has the following truth-table:

$$\top \leftrightarrow \top = \top \qquad \top \leftrightarrow \bot = \bot$$

$$\bot \leftrightarrow \top = \bot \qquad \bot \leftrightarrow \bot = \bot .$$

In other words, the biconditional is **true** just in case its terms have equal truth-values. The biconditional can also be expressed in terms of previous connectives:

$$x \leftrightarrow y = (x \rightarrow y) \wedge (y \rightarrow x) \qquad (2)$$

(verify!). In fact, this corresponds to our general attitude towards the proof of a biconditional, which is that it involves the proof of two conditionals.

The equalities (1), (2) may be considered as definitions of the conditional $\rightarrow$ and the biconditional $\leftrightarrow$ as operations in an arbitrary Boolean algebra. In case that algebra is $\mathcal{P}(B)$, the power-set algebra, then, for sets X and $Y \subseteq B$, we have that

$$X \rightarrow Y \underset{\text{def}}{=} (-X) \cup Y$$

and

$$X \leftrightarrow Y \underset{\text{def}}{=} ((-X) \cup Y) \cap ((-Y) \cup X) .$$

Next, we introduce a general construction on Boolean algebras.

Let $\mathfrak{A} = (A, \leq)$ be any Boolean algebra, I any set. We consider all functions from I into A as the elements of a new Boolean algebra denoted $\mathfrak{A}^I$; read " $\mathfrak{A}$-to-the-power-I ", or more simply, " $\mathfrak{A}$-to-I ". The underlying set of $\mathfrak{A}^I$ is, as we said, A^I , the set of all functions $\xi : I \to A$. The ordering in $\mathfrak{A}^I$, $\leq^*$, is defined *componentwise* from $\leq$: for $\xi, \zeta \in A^I$,

$$\xi \leq^* \zeta \iff \xi(i) \leq \zeta(i) \text{ for all } i \in I .$$

There are several things to check: firstly, that $\leq^*$ is indeed an order on A^I ; further, that this order has all the requisite properties to make $\mathfrak{A}^I = (A^I, \leq^*)$ a Boolean algebra. In fact, what happens is that the Boolean operations $\top^*$, $\bot^*$, $\wedge^*$, $\vee^*$, and $-^*$ in $\mathfrak{A}^I$ are all computed *componentwise*: for all $\xi, \zeta \in A^I$ and $i \in I$, we have:

$$\begin{aligned} \top^*(i) &= \top , \\ \bot^*(i) &= \bot , \\ (\xi \wedge \zeta)(i) &= \xi(i) \wedge \zeta(i) \end{aligned}$$

(we should have written $\xi \wedge^* \zeta$, but it is not necessary to be that pedantic ...).

$$\begin{aligned} (\xi \vee \zeta)(i) &= \xi(i) \vee \zeta(i) \\ (-\xi)(i) &= -(\xi(i)) \end{aligned}$$

The proof of all these assertions is easy. For instance, the assertion for $\wedge$ is that the function $\eta \in A^I$ for which $\eta(i) = \xi(i) \wedge \zeta(i)$ for all $i \in I$ is, in fact, the meet of ξ and ζ in $\mathfrak{A}^I$. According to a display on page 80 in Section 3.2, the best way to prove this is showing that

for all $\chi \in A^I$,

$$\chi \leq^* \eta \iff \chi \leq \xi \text{ and } \chi \leq \zeta .$$

When we put in the definition of η and that of $\leq^*$, we get

for all $\chi \in A^I$,
$\chi(i) \leq \xi(i) \wedge \zeta(i) \iff \chi(i) \leq \xi(i)$ and $\chi(i) \leq \zeta(i)$,

which, for each $i \in I$, is an instance of the same relation on page 80 in Section 3.2 for the original Boolean algebra $\mathfrak{A}$.

Let us apply the power-construction to the algebra $\mathfrak{A}=\mathbf{2}$. The elements of the Boolean algebra $\mathbf{2}^I$ are the functions $I \longrightarrow \{\top, \bot\}$; for $\xi, \eta \varepsilon \{\top, \bot\}^I$, $\xi \leq \eta$ iff $\xi(i) \leq \eta(i)$ for all $i \varepsilon I$. Also note that since $\mathbf{2}$ has just two elements $\bot$ and $\top$, and $\bot < \top$, $\xi(i) \leq \eta(i)$ is equivalent to saying that if $\xi(i) = \top$, then $\eta(i) = \top$.

The power-algebra $\mathbf{2}^I$ is in fact a very familiar one: it is isomorphic to the power-set algebra $\mathcal{P}(I)$:

$$\mathcal{P}(I) \cong \mathbf{2}^I .$$

Let us specify the isomorphism, in fact, in both directions:

$$\mathcal{P}(I) \underset{g}{\overset{f}{\rightleftarrows}} \mathbf{2}^I \ ::$$

for $X \varepsilon \mathcal{P}(I)$, $f(X)$ is the function $I \longrightarrow \{\top, \bot\}$ for which

$$f(X)(u) = \begin{cases} \top & \text{if } u \varepsilon X \\ \bot & \text{if } u \notin X \end{cases}$$

and

for any function $\xi \varepsilon \{\top, \bot\}^I$, $g(\xi)$ is the subset of I given as

$$g(\xi) = \{u \varepsilon I \mid \xi(u) = \top\}.$$

These mappings f and g respect the orders, and they are inverses of each other; these facts are easily checked (*exercises*). In other words, f is an isomorphism $f:\mathcal{P}(I)\xrightarrow{\cong}\mathbf{2}^I$. Note that, for $X\in\mathcal{P}(I)$, $f(X)$ is what we call the *characteristic function of* X.

This representation of power-set algebras provides a direct proof that

any identity that holds in the 2*-element algebra* $\mathbf{2}$ *holds in any power-set algebra, and hence, in any Boolean algebra whatsoever.*

The reason is that, as it is seen by inspection, an identity that holds in an algebra $\mathfrak{A}$ holds also in a power $\mathfrak{A}^I$ of it; also remember that we said in the previous section that all identities in set-algebras hold in all Boolean algebras.

Seen in the light of the last statement, the definition of "Boolean algebra" is just a summary of what identities hold in the algebra of the two truth-values! Note carefully that, if we take this "definition" of "Boolean algebra" as basic, it is not obvious -- although *now* known by us -- that the Boolean laws are all *consequences* of the few that we earlier explicitly specified as the Boolean laws.

When people talk about "Boolean functions", they mean functions of possibly several variables, all of which range over the set $\{\top, \bot\}$, and whose values are also in $\{\top, \bot\}$. (Very often (specially in computer science), we write 1 for $\top$, and 0 for $\bot$; but we will stick to the $\top$,$\bot$-notation.) If the function f in question has n variables $P_1, \ldots, P_n$ [we write P for the variables, since they are seen as "propositions"; in fact, they simply take the values $\top$ and $\top$], then f is a function

$$f : \{\top, \bot\}^n \longrightarrow \{\top, \bot\};$$

for any $P_1, \ldots, P_n\in\{\top, \bot\}$, $f(P_1, \ldots, P_n)$ is again an element of $\{\top, \bot\}$.

With the fixed n, note that the number of distinct n-variable Boolean functions is $2^{(2^n)}$. A Boolean function f can be represented by a *truth-table* listing all possible systems of argument-values, and the corresponding function-values. For instance, when $n=3$, the

truth-table might be like this (we write P, Q, R *for* P_1, P_2, P_3):

P	Q	R	$f(P, Q, R)$
$\top$	$\top$	$\top$	$\bot$
$\top$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\bot$
$\bot$	$\top$	$\bot$	$\top$
$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\top$

(3)

Now, regard the set $\{\top, \bot\}$ as the underlying set of $\mathbf{2}$. Then the n-variable Boolean functions form the underlying set of the power-algebra $\mathbf{2}^I$, where $I=\{\top, \bot\}^n$. In other words, for a fixed n , the n-variable Boolean functions themselves form a Boolean algebra. Let us write $\vec{P}$ to abbreviate $P_1, \ldots, P_n$. The Boolean operations on the power-algebra $\mathbf{2}^{(\{\top, \bot\}^n)}$, *the Boolean algebra of* n*-variable Boolean functions,* is defined componentwise:

$$(f \wedge g)(\vec{P}) = f(\vec{P}) \wedge g(\vec{P}) ,$$

and similarly for the other operations.

We write, more simply, $\mathrm{BF}[n]$ for $\mathbf{2}^{(\{\top, \bot\}^n)}$, the Boolean algebra of n-variable Boolean functions.

The most natural examples for Boolean functions are the *Boolean polynomials*: these are the functions that can be written down by repeated use of the basic Boolean operations. For instance, when $n=3$, the following are Boolean polynomials :

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

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

The fact that the last does not contain the variable P does not make it illegitimate as a three-variable polynomial: this one simply does not depend on P.

Boolean polynomials should be seen as analogs to ordinary (algebraic) polynomials. The differences are that, Boolean polynomials are functions on the truth-values, instead of numbers; and the basic Boolean operations figure in them, instead of the ordinary arithmetical operations + , · , etc.

A Boolean polynomial is (or, denotes) a Boolean function: substituting definite truth-values for the variables, and using the basic Boolean operations on truth-values, we get a definite value for the polynomial. For instance, here are all the values of the first of the two listed polynomials:

P	Q	R	$(\neg(((P \vee \neg R) \wedge (Q \vee \neg R)) \wedge \top)) \vee R$
$\top$	$\top$	$\top$	$\top$
$\top$	$\top$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\bot$
$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$

Calculating the value, for instance in the third line, looks like this:

$$\begin{array}{cccccccccccccccc} (\neg(((P & \vee & \neg & R) & \wedge & (Q & \vee & \neg & R)) & \wedge & \top)) & \vee & R & . \\ \bot & & \top & \top & \top & \bot & \top & & \top & \top & \top & \bot & & \top & \top & & \bot & \bot \\ 7 & & & 2 & 1 & & 5 & & & 4 & 3 & & & 6 & & & 8 & \end{array}$$

In this, first, we wrote the value of every variable under each occurrence of the variable, including the value $\top$ under the constant $\top$ in the polynomial; next, we proceeded to calculate the values of the part-expressions from the inside out; there are as many as there are *connectives*, occurrences of $\wedge$, $\vee$, and $-$. The numbers indicate the order in which we go through all constituent expressions until we reach the total expression in stage 8 ; the final result is that above 8 , $\bot$.

There is a slight ambiguity in the meaning of the expression "Boolean polynomial". We

sometimes mean the formal expression itself, rather than the function denoted by it. However, the official meaning should remain the function itself; when one wants to refer to the formal notion, one should say "formal polynomial". This remark is relevant in the light of the fact that two formally different Boolean polynomials may be *equal* to each other. In the first of the last two examples, the values in the value-column coincide with the values of R; the polynomial coincides (denotes the same function as) the simple polynomial ("monomial") R.

Of course, this phenomenon is familiar in the case of ordinary (algebraic) polynomials. E.g, the two formal polynomial expressions $(x-y)(x+y)$ and x^2-y^2 denote the *same* polynomial. We can see this by using the basic algebraic laws. The situation with Boolean polynomials is similar. Instead of going through the tables of values (which tend to be very large even with a moderate number of variables), we may use the Boolean identities to establish that two formal Boolean polynomials are the same polynomial. For instance, in the example at hand:

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

$$\begin{aligned}
&= (\neg(((P \wedge Q) \vee \neg R) \wedge \top)) \vee R && \text{(distributive law)}\\
&= (\neg((P \wedge Q) \vee \neg R)\ ) \vee R && \text{(unit law)}\\
&= (\neg(P \wedge Q) \wedge \neg\neg R)\ ) \vee R && \text{(De Morgan)}\\
&= (\neg(P \wedge Q) \wedge R)\ ) \vee R && \text{(double negation)}\\
&= R && \text{(commutative law, absorption)}
\end{aligned}$$

In fact, what we said about all identities of Boolean algebras being true in $\mathbf{2}$ means that every time two formal Boolean polynomials are the same function on the truthc-values, this fact can be deduced by using the Boolean identities alone.

Note that the Boolean operations on the Boolean polynomials as Boolean functions are performed by formally applying the operation in question. For instance, if the three variable polynomials mentioned above are briefly called f and g, then $f \wedge g$ is

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

What this means is that

the Boolean polynomials form a subalgebra of $BF[n]$.

Consider now the variables P_1, P_2, ... P_n themselves as such Boolean functions, in fact, Boolean polynomials. P_i is the function that satisfies

$$P_i(\varepsilon_1, \ldots, \varepsilon_i, \ldots, \varepsilon_n) = \varepsilon_i ;$$

here, each ε_1, , ε_n is a truth-value, $\top$ or $\bot$. (This is similar as when the single variable say y is regarded as one of the ordinary polynomials in variables x, y, z.) We clearly have that

the particular elements P_i of $BF[n]$ generate the Boolean subalgebra of Boolean polynomials.

Now, I claim that

the Boolean functions P_1, P_2, ... P_n are independent in the Boolean algebra of all n-element Boolean functions.

What we have to see is that, for any distribution of the values ε_1, , ε_n in $\{\top, \bot\}$, the meet-expression

$$\varepsilon_1 P_1 \wedge \varepsilon_2 P_2 \wedge \ldots \wedge \varepsilon_n P_n$$

is *different from* $\bot$ in the Boolean algebra of Boolean functions; here, εP means P if $\varepsilon = \top$, and $-P$ if $\varepsilon = \bot$. But if we give the value ε_i to P_i, we get that $\varepsilon_i P_i$ takes the value $\top$: $(P)(P{=}\top) = \top$; $(-P)(P{=}\bot) = \top$; thus,

$$(\varepsilon_1 P_1 \wedge \varepsilon_2 P_2 \wedge \ldots \wedge \varepsilon_n P_n)(P_1{=}\varepsilon_1, \ldots, P_n{=}\varepsilon_n) = \top \wedge \top \wedge \ldots \wedge \top = \top .$$

Since the function takes the value $\top$ at at least one system of arguments, the function is not the $\bot$-function, which is constant $\bot$.

Remember that an independent family of n elements generate a Boolean subalgebra of size $2^{(2^n)}$. It follows that there are exactly $2^{(2^n)}$ distinct Boolean polynomials. But the

whole algebra $\mathtt{BF}[n]$ is of the same size, $2^{(2^n)}$. It follows that

all Boolean functions are (represented by) Boolean polynomials. In fact, all Boolean functions can be written as joins of complete meet expressions in terms of the variables.

The expressions P_i and $\neg P_i$ (or, $\bar{P}_i$) are also called *literals*. The expression of a Boolean function as a join of distinct complete meets of literals is called the *disjunctive normal form* (dnf) of the function. We have that every function has a unique dnf: the complete meets of literals appearing in the dnf are determined as those atoms of $\mathtt{BF}[n]$ that are below the given function.

Applying duality, one also gets a *conjunctive normal form.*

The last-stated fact has concerning the existence of the dnf also has a direct proof, together with a simple method of producing the dnf of a function, based on the truth-table of the function. The result is this. Consider the truth-table of the n-variable Boolean function f. Select those lines in the table in which the value of f is $\top$; say the lines in which this is the case are

$$\ell_1, \ell_2, \ldots, \ell_k .$$

Each line ℓ_j (j=1, . . ., k) has a certain system of the values of the variables. Let us denote the value of P_i in line ℓ_j by ε_{ji} (and not ε_{ij}, because j denotes the "row-number", i the "column-number"). The dnf of f is $\bigvee_{j=1}^{k} \bigwedge_{i=1}^{n} \varepsilon_{ji} P_i$; or in more detail,

$$\begin{array}{ll} & \varepsilon_{11}P_1 \wedge \varepsilon_{12}P_2 \wedge \ldots \wedge \varepsilon_{1n}P_n \\ \vee & \varepsilon_{21}P_1 \wedge \varepsilon_{22}P_2 \wedge \ldots \wedge \varepsilon_{2n}P_n \\ \vee & \ldots \\ \vee & \varepsilon_{k1}P_1 \wedge \varepsilon_{k2}P_2 \wedge \ldots \wedge \varepsilon_{kn}P_n . \end{array}$$

Here we used the convention applied before: $\top P$ is P, $\bot P$ is $\neg P$.

To give an example, consider the function whose truth-table is (3). There are five lines where

the value is $\top$. The dnf is

$$P Q \bar{R} \vee P \bar{Q} R \vee \bar{P} Q R \vee \bar{P} \bar{Q} R \vee \bar{P} \bar{Q} \bar{R} \ .$$

The proof that this is a correct procedure has to show that the dnf constructed assumes the same values at each system of values for the variables. Now, the dnf is $\top$ if and only if one of its disjuncts is $\top$. But each disjunct corresponds to a line, say ℓ_j, where the value of f is $\top$. The corresponding disjunct is

$$\varepsilon_{j1} P_1 \wedge \varepsilon_{j2} P_2 \wedge \ldots \wedge \varepsilon_{jn} P_n \ , \qquad (4)$$

and this will take the value $\top$ iff each conjunct $\varepsilon_{ji} P_i$ takes the value $\top$, which is the case if and only if P_i takes the value ε_{ji}. This means that the unique system of truth-values where the value of (4) is $\top$ is precisely the one in line ℓ_j! We have concluded that the dnf takes the value $\top$ exactly in the lines ℓ_j, for $j=1, \ldots, k$, which are also exactly the lines where f is $\top$. This proves that the dnf and f are identical functions.

The dnf of a Boolean function may be extremely large already in case of a moderate number of variables. The problem of *Boolean realization* is to find a possibly small formal Boolean polynomial representing a given Boolean function.