Warwick Mathematics Exchange

MA3K6

Boolean Functions

2025, May 7th

Desync, aka The Big Ree

Contents

Introduction

ree

Disclaimer: I make *absolutely no guarantee* that this document is complete nor without error. In particular, any content covered exclusively in lectures (if any) will not be recorded here. This document was written during the 2023 academic year, so any changes in the course since then may not be accurately reflected.

Notes on formatting

New terminology will be introduced in *italics* when used for the first time. Named theorems will also be introduced in *italics*. Important points will be **bold**. Common mistakes will be underlined. The latter two classifications are under my interpretation. YMMV.

Content not taught in the course will be outlined in the margins like this. Anything outlined like this is not examinable, but has been included as it may be helpful to know alternative methods to solve problems.

The table of contents above, and any inline references are all hyperlinked for your convenience.

History

First Edition: 2025-05-04*
Current Edition: 2025-05-07

Authors

This document was written by R.J. Kit L., a maths student. I am not otherwise affiliated with the university, and cannot help you with related matters.

Please send me a PM on Discord @Desync#6290, a message in the WMX server, or an email to Warwick.Mathematics.Exchange@gmail.com for any corrections. (If this document somehow manages to persist for more than a few years, these contact details might be out of date, depending on the maintainers. Please check the most recently updated version you can find.)

If you found this guide helpful and want to support me, you can buy me a coffee!

(Direct link for if hyperlinks are not supported on your device/reader: ko-fi.com/desync.)

*Storing dates in big-endian format is clearly the superior option, as sorting dates lexicographically will also sort dates chronologically, which is a property that little and middle-endian date formats do not share. See ISO-8601 for more details. This footnote was made by the computer science gang.

1 Introduction

Let $\mathcal{B} \coloneqq \{0,1\}$, and let n be a positive integer. The points of the set $\mathcal{B}^n$ are called *binary vectors* or *Boolean points*. To simplify notation, we will write the elements of $\mathcal{B}^n$ without commas or parentheses, e.g.

$$\mathcal{B}^2 = \{00,01,10,11\}$$

A *Boolean function of n variables* is a function $f : \mathcal{B}^n \to \mathcal{B}$. A point $X = (x_1, \ldots, n_n) \in \mathcal{B}^n$ is a *true point* of f if $f(X) = 1$, and is a *false point* if $f(X) = 0$. The set of true points of f is denoted by $T(f)$, and the set of false points by $F(f)$.

The most elementary way to define a Boolean function is via its *truth table*, i.e. a list of all of the points of $\mathcal{B}^n$, along with the value of the function at each point.

Example. A Boolean function f of three variables, x, y, and z, defined by its truth table:

x,y,z	$f(x,y,z)$
000	1
001	1
010	1
011	1
100	0
101	0
110	0
111	1

This function has five true points

$$T(f) = \{000,001,010,011,111\}$$

and three false points

$$F(f) = \{100,101,110\}$$

△

In a truth table, the Boolean points are normally listed in lexicographic order, in which case, we only need the output values, which can be represented as a *vector of values*. For instance, the function above as vector of values 11110001.

Theorem 1.1. *The number of Boolean functions of n variables is 2^{2^n}.*

Proof. There are 2^n Boolean points, and for each one, a Boolean function can take one of two possible values. ∎

1.1 Boolean Functions of One or Two Variables

There are four Boolean functions of one variable.

x	$g_1(x) \equiv \mathbf{0}_1$	$g_2(x) \equiv \mathbf{1}_1$	$g_3(x) \equiv x$	$g_4(x) \equiv \overline{x}$
0	0	1	0	1
1	0	1	1	0

- $\mathbf{0}_n$ is the *constant zero* function of n variables that takes the value 0 on all points of $\mathcal{B}_n$;
- $\mathbf{1}_n$ is the *constant one* function of n variables that takes the value 1 on all points of $\mathcal{B}_n$;
- $\overline{x}$ is the *negation*, *complementation*, or *Boolean NOT* of x.

There are sixteen Boolean functions of two variables:

x,y	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Many of these have special names and notations:

- $f_2(x,y) = x \wedge y = x\&y = xy$ is *conjunction* or *Boolean AND*;
- $f_8(x,y) = x \vee y = x\&y = x + y$ is *disjunction* or *Boolean OR*;
- $f_7(x,y) = x \oplus y$ is addition modulo 2 or *Boolean XOR*;
- $f_9(x,y) = x \downarrow y$ is the *Peirce arrow* or *Boolean NOR*;
- $f_{10}(x,y) = x \sim y$ is *equivalence*;
- $f_{14}(x,y) = x \to y$ is *implication*;
- $f_{15}(x,y) = x \uparrow y$ is the *Sheffer stroke* or *Boolean NAND*;

1.2 An Aside on Set Systems, Hypergraphs, and Graphs

A *set system* is a pair $(V,\mathcal{E})$ consisting of a finite set V called the *ground set* or *universe*, and a collection of subsets $\mathcal{E} \subseteq \mathcal{P}(V)$.

If $V = \{v_1, \ldots, v_n\}$, then any subset $A \subseteq V$ can be described by its characteristic vector e_A. That is, a binary vector $(a_1, \ldots, a_n) \in \mathcal{B}^n$ such that $a_i = 1$ if and only if $v_i \in A$.

Every set system over a totally ordered universe of n elements uniquely corresponds to a Boolean function f of n variables by mapping a Boolean point to true under f if and only if it is the characteristic vector of an subset in $\mathcal{E}$. This correspondence also establishes a relationship between set operations and Boolean functions. For instance, the union of two sets corresponds to disjunction in that $C = A \cup B$ if and only if $e_C = e_A \vee e_B$, where the disjunction is taken pointwise over the vector. Similarly, intersection correspond to conjunction, symmetric difference to addition modulo 2, relative difference to the function f_3 in the Table 1, and complementation to negation.

Set systems can also be interpreted as *hypergraphs*, with the ground set V containing *vertices* and $\mathcal{E}$ containing *hyperedges*. In particular, a hypergraph in which every hyperedge consists of two vertices is a *graph*, in which case the hyperedges are called *edges*. A *directed graph* is a graph in which every edge is an ordered pair of vertices, in which case the edges are called *arcs*.

1.3 Basic Identities

Theorem 1.2. *For all $x,y,z \in \mathcal{B}$,*

(i) $x \vee 1 = 1$ *and* $x \wedge 0 = 0$*;*

(ii) $x \vee 0 = x$ *and* $x \wedge 1 = x$*;*

(iii) $x \vee y = y \vee x$ *and* $x \wedge y = y \wedge x$ *(commutativity);*

(iv) $(x \vee y) \vee z = x \vee (y \vee z)$ *and* $(x \wedge y) \wedge z = x \wedge (y \wedge z)$ *(associativity);*

(v) $x \vee x = x$ *and* $x \wedge x = x$ *(idempotency);*

(vi) $x \vee (x \wedge y) = x$ *and* $x \wedge (x \vee y) = x$ *(absorption);*

(vii) $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ *and* $x(y \vee z) = (x \wedge y) \vee (x \wedge z)$ *(distribution);*

(viii) $x \vee \overline{x} of 1$ *and* $x \wedge \overline{x} = 0$*;*

(ix) $\overline{\overline{x}} = x$ *(involution);*

(x) $\overline{x \vee y} = \overline{x} \wedge \overline{y}$ *and* $\overline{x \wedge y} = \overline{x} \vee \overline{x}$ *(De Morgan's laws)*

(xi) $x \vee (\overline{x} \wedge y) = x \vee y$ *and* $x \wedge (\overline{x} \vee y) = xy$ *(Boolean absorption);*

Proof. All easily verified through truth tables. ■

1.4 Boolean Expressions

Given a finite collection of Boolean variables $x_1, \dots, x_n$, a *Boolean expression* or *Boolean formula* in the variables $x_1, \dots, x_n$ is defined inductively as follows:

- The constants 0 and 1, and the variables $x_1, \dots, x_n$ are Boolean expressions in $x_1, \dots, x_n$;
- If ϕ and ψ are Boolean expressions in $x_1, \dots, x_n$, then $(\phi \vee \psi)$, $(\phi \wedge \psi)$, and $\overline{\phi}$ are Boolean expressions in $x_1, \dots, x_n$.

Given a Boolean expression ϕ, we call $\vee$, $\wedge$, and the negation operator $\overline{\bullet}$ the *operators* of the expression. A *literal* is either a variable or its complement.

We will adopt the convention that the operators have precedence, in decreasing order: complementation, conjunction, then disjunction. Along with the associativity properties, these precedence assumptions allow us to simplify Boolean expressions by removing extraneous parentheses. For instance, $(((\overline{x} \vee y)(y \vee \overline{z})) \vee ((xy)z))$ simplifies to $(\overline{x} \vee y)(y \vee \overline{z}) \vee xyz$.

Every Boolean expression ψ represents a unique boolean function f_ψ in the obvious way. Two Boolean expresions ϕ and ψ are *equivalent* if they represent the same Boolean function, and we write $\phi = \psi$ to denote this relation.

Note the basic identities in the previous theorem preserve equivalence of Boolean expressions.

1.5 Duality

Given a Boolean function f, we define the *dual* f^d of f to be the Boolean function

$$f^d(x_1, \dots, x_n) \coloneqq \overline{f(\overline{x}_1, \dots, \overline{x}_n)}$$

If $X = (x_1, \dots, x_n)$, we abbreviate the vector of complemented variables by $\overline{X} = (\overline{x}_1, \dots, \overline{x}_n)$, so $f^d(X) = \overline{f(\overline{X})}$.

The vector of values of the dual f^d is given by vertically reflecting the vector of values of f, then complementing pointwise.

Example.

x,y	$f(x,y)$	$f^d(x,y)$
00	0	0
01	1	0
10	1	0
11	1	1

This shows that the dual of disjunction is conjunction, and vice versa. △

Theorem 1.3. *If f and g are Boolean functions, then,*

(i) $(f^d)^d = f$*;*

(ii) $\overline{f}^d = \overline{f^d}$*;*

(iii) $(f \vee g)^d = f^d \wedge g^d$*;*

(iv) $(f \wedge g)^d = f^d \vee g^d$.

Proof.

(i) Since complementation is involutive,

$$\begin{aligned}(f^d)^d(X) &= \overline{f^d(\overline{X})} \\ &= \overline{\overline{f(\overline{\overline{X}})}} \\ &= f(X)\end{aligned}$$

(ii) Similarly,

$$\begin{aligned}\overline{f}^d(X) &= \overline{\overline{f(\overline{X})}} \\ &= \overline{f^d(X)}\end{aligned}$$

(iii) By De Morgan's laws,

$$\begin{aligned}(f \vee g)^d &= \overline{(f \vee g)(\overline{X})} \\ &= \overline{f(\overline{X}) \vee g(\overline{X})} \\ &= \overline{f(\overline{X})} \wedge \overline{g(\overline{X})} \\ &= f^d(X) \wedge g^d(X)\end{aligned}$$

(iv) Dually,

$$\begin{aligned}(f \wedge g)^d &= \overline{(f \wedge g)(\overline{X})} \\ &= \overline{f(\overline{X}) \wedge g(\overline{X})} \\ &= \overline{f(\overline{X})} \vee \overline{g(\overline{X})} \\ &= f^d(X) \vee g^d(X)\end{aligned}$$

■

Given a Boolean expression ϕ, we define the *dual* ϕ^d of ϕ to be the Boolean expression obtained from ϕ by interchanging the operators $\vee$ and $\wedge$, and the constants 0 and 1.

Theorem 1.4. *If the Boolean expression ϕ represents the Boolean function f, then ϕ^d represents f^d.*

Proof. We proceed by structural induction on ϕ. If ϕ is a constant or literal, then this is clear by involution. Otherwise, suppose $\phi = \phi_1 \vee \phi_2$ for some expressions ϕ_1 and ϕ_2. By the induction hypothesis, the expressions ϕ_1^d and ϕ_2^d represent the duals $f_{\phi_1}^d$ and $f_{\phi_2}^d$ respectively, so $\phi^d = \phi_1^d \wedge \phi_2^d$ represents the dual $f_{\phi_1}^d \wedge f_{\phi_2}^d$ of $f_{\phi_1} \vee f_{\phi_2} = f_\phi$. The proof for $\phi = \phi_1 \wedge \phi_2$ and $\phi = \overline{\psi}$ are similar. ■

1.6 Normal Forms

An *elementary conjunction* or a *term* is a conjunction of literals, and an *elementary disjunction* or a *clause* is a disjunction of literals. That is, an elementary conjunction is an expression of the form

$$C = \bigwedge_{i \in A} x_i \wedge \bigwedge_{j \in B} \overline{x}_j, \qquad A \cap B = \varnothing$$

and an elementary disjunction is an expression of the form

$$D = \bigvee_{i \in A} x_i \vee \bigvee_{j \in B} \overline{x}_j, \qquad A \cap B = \varnothing$$

That is, a variable can appear only once, and only either complemented or not.

A *disjunctive normal form* (*DNF*) is a disjunction of terms, and a *conjunctive normal form* (*CNF*) is a conjunction of clauses. That is, a DNF is a expression of the form

$$\bigvee_{k=1}^{m} C_k = \bigvee_{k=1}^{m} \left(\bigwedge_{i \in A_k} x_i \wedge \bigwedge_{j \in B_k} \overline{x}_j \right)$$

where each C_k is a term, and a CNF is an expression of the form

$$\bigwedge_{k=1}^{m} D_k = \bigwedge_{k=1}^{m} \left(\bigvee_{i \in A_k} x_i \vee \bigvee_{j \in B_k} \overline{x}_j \right)$$

where each D_k is a clause.

Theorem 1.5. *Every Boolean function admits a DNF and a CNF representation.*

Proof. Let f be a Boolean function and let $T = T(f)$ be the set its of true points. Consider the DNF

$$\phi_f(x_1, \ldots, x_n) = \bigvee_{Y \in T} \left(\bigwedge_{i : y_i = 1} x_i \wedge \bigwedge_{j : y_j = 0} \overline{x}_j \right) \tag{1}$$

Then, a point X^* is a true point of the function F represented by ϕ_f if and only if there exists a true point $Y = (y_1, \ldots, y_n) \in T$ of f such that

$$\bigwedge_{i : y_i = 1} x_i^* \wedge \bigwedge_{j : y_j = 0} \overline{x}_j^* = 1$$

But this just means that $x_i^* = 1$ whenever $y_i = 1$ and $x_j^* = 0$ whenever $y_i = 0$. That is, $X^* = Y$. Hence X^* is a true point of F if and only if it is a true point of f, and hence $f = F$ is represented by ϕ_f.

Similar reasoning establishes that f is also represented by the CNF

$$\phi_f(x_1, \ldots, x_n) = \bigwedge_{Y \in F} \left(\bigvee_{i : y_i = 0} x_i \vee \bigvee_{j : y_j = 1} \overline{x}_j \right) \tag{2}$$

where $F = F(f)$ is the set of false points of f. Alternatively, this expression can be obtained as the dual of (1). ∎

The expressions in (1) and (2) are of a special form. In particular, every term in (1) contains n literals. Such a term is called a *minterm*, and the whole expression (1) representing f is called the *minterm expression* or *canonical DNF* of f. Similarly, every clause in (2) contains n literal and is called a *maxterm*, and the whole expression (2) is called the *maxterm expression* or *canonical CNF* of f.

The proof above gives an easy way to construct the canonical DNF/CNF of a function from its truth table. For the canonical DNF:

- Identify all rows where the function output is 1;

- For each row, write a minterm by including each variable with polarity corresponding to its appearance in the row;
- Take the disjunction of all the minterms.

That is, each minterm corresponds to a unique input combination, so we take the ones where the function is true, then allow any of the combinations by taking a disjunction.

Similarly, for the canonical CNF:

- Identify all rows where the function output is 0;
- For each row, write a maxterm by including each variable with polarity opposite to its appearance in the row;
- Take the conjunction of all the maxterms.

Dually, each maxterm corresponds to the complement of a unique input combination, i.e. evaluates to false for only one input, so we take the ones where the function is false, then mask out the false rows by taking a conjunction.

Example. Consider the following function:

x,y,z	$f(x,y,z)$
000	1
001	0
010	1
011	1
100	0
101	0
110	1
111	1

This function is represented by the DNF

$$\overline{x}\,\overline{y}\,\overline{z} \vee \overline{x}y\overline{z} \vee \overline{x}yz \vee xy\overline{z} \vee xyz$$

and the CNF

$$(x \vee y \vee \overline{z})(\overline{x} \vee y \vee z)(\overline{x} \vee y \vee \overline{z})$$

△

A canonical DNF and CNF is unique up to permutation of its terms and literals. However, a function generally admits many other non-canonical DNF and CNF representations.

Example. The function from the previous example can also be represented by the DNF

$$\overline{x}\,\overline{z} \vee y$$

△

The *degree* of a term $C = \bigwedge_{i\in A} x_i \bigwedge_{j\in B} \overline{x}_j$ is the number of literals appearing in C. That is, $|A| + |B|$.

More generally, the *degree* of a DNF $\phi = \bigvee_{k=1}^{m} C_k$ is the maximum degree of the C_k. A DNF is called *linear* if its degree is at most 1, *quadratic* if at most 2, *cubic* if at most 3, etc. We write $|\phi|$ for the number of literals in ϕ, also called its *length*; and $\|\phi\|$ for the number of terms in ϕ.

Example. The DNF $\phi = \overline{x}\,\overline{z} \vee y$ has $|\phi| = 3$ literals and $\|\phi\| = 2$ terms, and has term degrees 2 and 1, and is thus of degree 2, or is quadratic. △

The following example shows that a shortest DNF may have a length exponential in the number of variables.

Example. The function represented by the CNF

$$\psi(x_1,\dots,x_{2n}) = (x_1 \vee x_2)(x_3 \vee x_4)\cdots(x_{2n-1} \vee x_{2n})$$

has a unique shortest DNF consisting of 2^n terms, each containing exactly one literal from each clause of ψ. △

1.7 Orthogonal DNFs

A DNF

$$\phi = \bigvee_{k=1}^{m} C_k = \bigvee_{k=1}^{m} \left(\bigwedge_{i\in A_k} x_i \wedge \bigwedge_{j\in B_k} \overline{x}_j \right), \qquad A_k \cap B_k = \varnothing$$

is *orthogonal* or is a *sum of disjoint products* if $(A_k \cap B_\ell) \cup (A_\ell \cap B_k) \neq \varnothing$ for all $k \neq \ell$.

That is, a DNF is orthogonal if every pair of terms is "conflicting" in at least one variable; there must exist a variable that appears complemented in one term and uncomplemented in the other. Alternatively, a DNF is orthogonal if and only if the product (conjunction) of every pair of its terms is 0 at every Boolean point (since conflicting variables would yield 0 in the product).

Example. The DNF $\phi = \overline{x}_1\overline{x}_2x_4 \vee \overline{x}_1x_3x_4$ is not orthogonal since x_1 is negative in both terms, x_2 and x_3 do not appear in both terms, and x_4 is positive in both terms, so none of the variables are in conflict. Alternatively, both terms (and hence their product) are equal to 1 at 0011. △

Note that orthogonality is not preserved under equivalence.

Example. The DNF ϕ is equivalent to the DNF $\psi = \overline{x}_1\overline{x}_2x_4 \vee \overline{x}_hx_2x_3x_4$, which is orthogonal since its only pair of terms are conflicting at the variable x_2. △

Note that the minterm expression constructed in the proof of Theorem 1.5 is orthogonal, so we can specialise the result further to:

Theorem 1.6. *Every Boolean function can be represented by an orthogonal DNF.*

One of the main motivations for our interest in orthogonal DNFs is that the number of true points $\omega(f) := \left|T(f)\right|$ of a function f expressed in this form can be efficiently computed.

Theorem 1.7. *If a Boolean function f on $\mathcal{B}^n$ is represented by an orthogonal DNF, then the number of its true points is*

$$\omega(f) = \sum_{k=1}^{m} 2^{n-|A_k|-|B_k|}$$

Proof. The DNF takes the value 1 precisely when one of its terms takes the value 1. ■

The *Chow parameters* of a Boolean function f on $\mathcal{B}^n$ are the $n+1$ integers $(\omega_1,\dots,\omega_n,\omega)$ where $\omega = \omega(f)$ is the number of true points of f and ω_i is the number of true points $X^* = (x_1^*,\dots,x_n^*)$ of f with $x_i^* = 1$.

Example. The function f represented by the orthogonal DNF $\psi = \overline{x}_1\overline{x}_2\overline{x}_4 \vee \overline{x}_1x_2x_3x_4$ is true at 0001, 0011, and 0111. None of these have $x_1 = 1$; 1 has $x_2 = 1$; 2 have $x_3 = 1$; 3 have $x_4 = 1$; and there are 3 total true points; so its Chow parameters are $(\omega_1,\omega_2)(0,1,2,3,3)$. △

The same reasoning as in the proof of the previous theorem also shows that the Chow parameters of a function represented in orthogonal form can be efficiently computed: for ω, this is precisely the statement of the theorem; for ω_i, this follows from the fact that the DNF obtained by fixing $x_i = 1$ in an orthogonal DNF is still orthogonal.

1.8 Implicants

Given two Boolean functions f,$g : \mathcal{B}^n \to \mathcal{B}$, we say that f *implies* g, that f is a *minorant* of g, or that g is a *majorant* of f, and write $f \leq g$, if $f(X) = 1$ implies $g(X) = 1$ for all $X \in \mathcal{B}^n$. That is, f implies g pointwise. Equivalently, if we identify each function f with its set of true points, then this relation is precisely the subset containment relation.

This definition extends to Boolean expressions in the obvious way. We will often identify Boolean functions with representing Boolean expressions, and write, for instance, $\psi \leq f$ for $\psi \leq \phi_f$.

Theorem 1.8. *For all Boolean functions f,$g : \mathcal{B}^n \to \mathcal{B}$, the following are equivalent:*

(*i*) $f \leq g$;

(*ii*) $f \vee g = g$;

(*iii*) $\overline{f} \vee g = \mathbf{1}_n$;

(*iv*) $f \wedge g = f$;

(*v*) $f \wedge \overline{g} = \mathbf{0}_n$.

Proof. It suffices to note that each of these statements fail precisely when there exists $X \in \mathcal{B}^n$ such that $f(X) = 1$ but $g(X) = 0$. ■

Theorem 1.9. *For all Boolean functions f,g,$h : \mathcal{B}^n \to \mathcal{B}$,*

(*i*) $\mathbf{0}_n \leq f \leq \mathbf{1}_n$;

(*ii*) $f \wedge g \leq f \leq f \vee g$;

(*iii*) $f = g$ *if and only if* $f \leq g$ *and* $g \leq f$;

(*iv*) ($f \leq h$ *and* $g \leq h$) *if and only if* $f \vee g \leq h$;

(*v*) ($f \leq g$ *and* $f \leq h$) *if and only if* $f \leq g \wedge h$;

(*vi*) *if* $f \leq g$, *then* $f \wedge h \leq g \wedge h$;

(*vii*) *if* $f \leq g$, *then* $f \vee h \leq g \vee h$;

For two Boolean function represented by arbitrary Boolean expressions, it can be non-trivial to verify whether or not f implies g. However, for elementary conjunctions, implication is easy to verify. An elementary conjunction implies another elementary conjunction if and only if the latter results from the former by deleting literals, i.e. by removing constraints.

Theorem 1.10. *The elementary conjunction $C_{AB} = \bigwedge_{i\in A} x_i \wedge \bigwedge_{j\in B} \overline{x}_j$ implies the elementary conjunction $C_{XY} = \bigwedge_{i\in F} x_i \wedge \bigwedge_{j\in G} \overline{x}_j$ if and only if $F \subseteq A$ and $G \subseteq B$.*

Proof. Suppose that $F \subseteq A$ and $G \subseteq B$, and let $X \in \mathcal{B}^n$. If $C_{AB}(X) = 1$, then $x_i = 1$ for all $i \in A$ and $x_j = 0$ for all $j \in B$. So in particular, $x_i = 1$ for all $i \in F$ and $x_j = 0$ for all $j \in G$, so $C_{FG} = 1$ and hence $C_{AB} \leq C_{FG}$.

Conversely, suppose $C_{AB} \leq C_{FG}$, and suppose for a contradiction that $F \not\subseteq A$, so there exists $k \in F \setminus A$. Fix $x_i = 1$ for all $i \in A$ and $x_j = 0$ for $j \notin A$, and let $X = (x_1, \ldots, x_n)$. Then, $C_{AB}(X) = 1$, but $C_{FG}(X) = 0$ since $x_k = 0$ and $k \in F$. ■

Let f be a Boolean function and C be an elementary conjunction. Then, C is an *implicant* of f if $C \leq f$.

Theorem 1.11. *If ϕ is a DNF representation of f, then every term of ϕ is an implicant of f. Moreover, if an elementary conjunction C is an implicant of f, then the DNF $\phi \vee C$ also represents f.*

Proof. Firstly, note that whenever any term of ϕ takes the value 1, then the entire disjunction ϕ, and hence f, takes the value 1. Then, $\phi \vee C \leq f$ since ϕ and C both imply f, and $f \leq \phi \leq \phi \vee C$, so f is represented by $\phi \vee C$. ■

Example. Let $f = xy \vee x\overline{y}z$. Then, the terms xy and $x\overline{y}z$ are implicants of f. The term xz is also an implicant of f, so $xy \vee x\overline{y}z \vee xz$ also represents f. △

Let f be a Boolean function and C_1 and C_2 be implicants of f. Then, C_1 *absorbs* C_2 if $C_1 \vee C_2 = C_1$, or equivalently, if $C_2 \leq C_1$.

Let f be a Boolean function and C be an implicant of f. Then, C is a *prime implicant* of f if C is not absorbed by any other implicant of f. That is, a prime implicant is a maximal conjunction implying f.

Theorem 1.12. *Every Boolean function can be represented by the disjunction of all its prime implicants.*

Proof. Let f be a Boolean function on $\mathcal{B}^n$ with prime implicants $P_1, \ldots, P_m$. Consider any DNF representation of f, say $\phi = \bigvee_{k=1}^{r} C_k$. Then, the DNF

$$\psi = \bigvee_{k=1}^{r} C_k \vee \bigvee_{k=1}^{m} P_j$$

Now, every term C_k is an implicant of f and is hence absorbed by some prime implicant P_j. So $C_k \vee P_j = P_j$, and $\psi = \bigvee_{j=1}^{m} P_j$ represents f. ■

The DNF of all prime implicants of a Boolean function is called the *complete DNF* or *Blake canonical form* of the function.

Example. Consider again the function $f = xy \vee x\overline{y}z$. Its prime implicants are xy and xz, so $f = xy \vee xz$ is its complete DNF. △

An interesting corollary of this theorem is that every Boolean function is uniquely identified by the list of its prime implicants. Equivalently, two Boolean functions are equal if and only if they have the same complete DNF.

Let $\phi = \bigvee_{k\in\Omega} C_k$ be a DNF representation of a Boolean function f on $\mathcal{B}^n$. We say that ϕ is a *prime DNF* of f if each term C_k is a prime implicant of f. We say that ϕ is an *irredundant DNF* of f if there is no $j \in \Omega$ such that $\psi = \bigvee_{k\in\Omega\setminus\{j\}} C_k$ represents f.

The notion of (prime) implicants naturally have a dual notion for disjunctions. Let f be a Boolean function and D be an elementary disjunction. Then, D is an *implicate* of f if $f \leq D$, and is furthermore a *prime implicate* if it is not implied by any other implicate of f. That is, a prime implicate is a minimal disjunction implied by f.

Theorem 1.13. *Every Boolean function can be represented by the conjunction of all its prime implicates.*

Example. The function $g = x\overline{y} \vee \overline{x}y \vee x\overline{z}$ has four implicates, namely $(x \vee y)$, $(x \vee y \vee z)$, $(x \vee y \vee \overline{z})$, and $(\overline{x} \vee \overline{y} \vee \overline{z})$. Only the first and last implicate in this list are prime, so $g = (x \vee y)(\overline{x} \vee \overline{y} \vee \overline{z})$. △

1.9 Generation of All Prime Implicates from a DNF Representation

If xC and $\overline{x}D$ are two elementary conjunctions such that CD is not identically 0, then we say that CD is the *consensus* of xC and $\overline{x}D$, and that CD is *derived from* xC *and* $\overline{x}D$ *by consensus on* x.

Given an arbitrary DNF ϕ, the *consensus procedure* generates the complete DNF equivalent to ϕ by repeatedly applying the operations of absorption ($x \vee xy = x$) and consensus:

- If there exist two terms C and D of ϕ such that C absorbs D, remove D from ϕ.

- If there exist two terms x_iC and $\overline{x}_iD$ of ϕ such that x_iC and $\overline{x}_iD$ have a consensus CD that is not absorbed by another term of ϕ, then add CD to ϕ.

The procedure halts when:

- the absorption operation cannot be applied, and;
- either the consensus operation cannot be applied, or all the terms that can be produces by consensus are absorbed by other terms of ϕ.

A DNF is *closed under absorption* if it satisfies the first stopping condition, and is *closed under consensus* if it satisfies the second.

The consensus procedure always terminates and produces a DNF closed under consensus and absorption in a finite number of steps. Indeed, the number of terms in the given variables is finite, and once a term is removed by absorption, it will never be added by consensus.

Example. Consider the DNF

$$\phi(x_1,x_2,x_3,x_4) = x_1\overline{x}_2x_3 \vee \overline{x}_1\overline{x}_2x_4 \vee x_2x_3x_4$$

Absorption is not possible. We can apply consensus on x_1 on the first two terms to obtain the term $\overline{x}_2x_3\overline{x}_2x_4 = \overline{x}_2x_3x_4$ not absorbed by any term of ϕ, to obtain the DNF

$$\phi'(x_1,x_2,x_3,x_4) = x_1\overline{x}_2x_3 \vee \overline{x}_1\overline{x}_2x_4 \vee x_2x_3x_4 \vee \overline{x}_2x_3x_4$$

Again, absorption is not possible. We can apply consensus on x_2 on the last two terms to obtain the term x_3x_4 not absorbed by any existing terms:

$$\phi''(x_1,x_2,x_3,x_4) = x_1\overline{x}_2x_3 \vee \overline{x}_1\overline{x}_2x_4 \vee x_2x_3x_4 \vee \overline{x}_2x_3x_4 \vee x_3x_4$$

The new term absorbs the previous two:

$$\phi''(x_1,x_2,x_3,x_4) = x_1\overline{x}_2x_3 \vee \overline{x}_1\overline{x}_2x_4 \vee x_3x_4$$

Now, neither consensus and absorption can be applied, so the procedure stops, and these three terms are the prime implicants of ϕ. △

We have already observed that the operations of absorption and consensus transform DNFs,but do not change the Boolean functions that they represent. This is implied by the two lemmata below, which easily follow from the basic Boolean identities.

Lemma 1.14. *For any two elementary conjunctions C and CD,*

$$C \vee CD = C$$

Lemma 1.15. *For any two elementary conjunctions xC and $\overline{x}D$,*

$$xC \vee \overline{x}D = xC \vee \overline{x}D \vee CD$$

The importance of the consensus procedure is due to the fact that it produces the complete DNF. To prove this, we need one more lemma.

Lemma 1.16. *Given any DNF ϕ of a Boolean function f, if C is an implicant of f that involves all variables present in ϕ, then C is absorbed by a term of ϕ.*

Proof. If C contains all the variables of ϕ, then the valuation of that makes $C = 1$ assigns values to all the variables in ϕ. Since C is an implicant of f, this assignment also makes $\phi = 1$, and hence at least one term of ϕ is 1. This term absorbs C. ■

Theorem 1.17. *Given any DNF ϕ of a Boolean function f, the consensus procedure applied to ϕ yields the complete DNF of f.*

Proof. Suppose otherwise that there exists a Boolean function f and a DNF ϕ of f such that the consensus procedure produces a DNF ψ that does not contain a prime implicant C_0 of f. Then, by Theorem 1.20, C_0 only involves variables present in ψ. Consider the set $\mathcal{S}$ of elementary conjunctions C satisfying the following conditions:

- C only contains variables present in ψ;
- $C \leq C_0$ (and therefore C is an implicant of f);
- C is not absorbed by any term in ψ.

The set $\mathcal{S}$ is non-empty since C_0 satisfies all three conditions, so let C^m be a term of maximum degree in $\mathcal{S}$. Since C^m is not absorbed by any term in ψ, C^m cannot involve all the variables present in ψ by the previous lemma. Let x be a variable present in ψ not present in C^m. The degree of the elementary conjuctions xC^m and $\overline{x}C^x$ exceeds that of C^m, and since the degree of C^tm is maximum, xC^m and $\overline{x}C^m$ do not belong to $\mathcal{S}$ and therefore cannot satisfy all three conditions. Since they clearly must satisfy the first two conditions, they must violate the third, so there must exist terms C' and C'' in ψ such that $xC^m \leq C'$ and $\overline{x}C^m \leq C''$. Since C^m is not absorbed by either C' nor C'', it follows that $C' = xD'$ and $C'' = \overline{x}D''$, where D' and D'' are elementary conjunctions that absorb C^m. This implies that D' and D'' do not conflict in any variable. Therefore, the consensus of C' and C'' exists, namely $D'D''$, and this term absorbs C^m. Since the consensus procedure stops on the DNF ψ, there must exist a term C''' in ψ that absorbs $D'D''$. Then, C''' must also absorb C^m, contradicting the assumption that $C^m \in \mathcal{S}$. ■

1.10 Restrictions of Functions, Essential Variables

Let f be a Boolean function on $\mathcal{B}^n$ and let $k \in [n]$. We define the *restrictions* $f|_{x_k=1}$ and $f|_{x_k=0}$ to be the Boolean functions on $\mathcal{B}^{n-1}$ defined by:

$$f|_{x_k=1}(x_1, \dots, \widehat{x_k}, \dots, x_n) = f(x_1, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_n)$$
$$f|_{x_k=0}(x_1, \dots, \widehat{x_k}, \dots, x_n) = f(x_1, \dots, x_{k-1}, 0, x_{k+1}, \dots, x_n)$$

That is, the functions obtained from f by fixing its kth argument to 1 and 0 respectively.

Example. Consider the function $f(x,y,z) = (xz \vee y)(x \vee \overline{z}) \vee \overline{x}\,\overline{y}$. Then,

$$\begin{aligned} f|_{x=1}(y,z) &= (1z \vee y)(1 \vee \overline{z}) \vee \overline{1}\overline{y} \\ &= z \vee y \end{aligned}$$

$$\begin{aligned} f|_{x=0}(y,z) &= (0z \vee y)(0 \vee \overline{z}) \vee \overline{0}\overline{y} \\ &= y\overline{z} \vee \overline{y} \\ &= \overline{y} \vee \overline{z} \end{aligned}$$

△

Theorem 1.18. *Let f be a Boolean function on $\mathcal{B}^n$, let ψ be a representation of f and let $k \in [n]$. Then, the expression obtained by substituting the constant 1 (respectively, 0) for every occurrence of x_k in ψ represents $f|_{x_k=1}$ (respectively, $f|_{x_k=0}$).*

Theorem 1.19. *Let f be a Boolean function on $\mathcal{B}^n$ and let $k \in [n]$. Then,*

$$f(x_1, \dots, x_n) = x_k f|_{x_k=1} \vee \overline{x}_k f|_{x_k=0}$$

Proof. Clear by case analysis on the value of x_k. ■

The right side of this identity is called the *Shannon expansion* of f with respect to x_k. By applying the Shannon expansion to a function and its successive restrictions until these restrictions become constants or literals, we obtain an orthogonal DNF of the function, which can easily be proved by induction. However, not every orthogonal DNF can be obtained in this way, since the Shannon expansion necessarily produces a DNF in which one of the variables appears in all of the terms.

Example. Consider the function $f = (xz \vee y)(x \vee \overline{z}) \vee \overline{x}\,\overline{y}$ from the previous example. The Shannon expansion of $f|_{x=1}$ with respect to y is

$$\begin{aligned} f|_{x=1} &= yf|_{x=1,y=1} \vee \overline{y}f|_{x=1,y=0} \\ &= y(z \vee 1) \vee \overline{y}(z \vee 0) \end{aligned}$$

Note that $z \vee 1 = 1$ is a constant and $z \vee 0 = z$ is a literal, so we stop here for the expansion of $f|_{y=1}$. The Shannon expansion of $f|_{x=0}$ with respect to y is

$$\begin{aligned} f|_{x=0} &= yf|_{x=0,y=1} \vee \overline{y}f|_{x=0,y=0} \\ &= y(\overline{1} \vee \overline{z}) \vee \overline{y}(\overline{0} \vee \overline{z}) \end{aligned}$$

Here, $\overline{1} \vee \overline{z} = \overline{z}$ is a literal and $\overline{0} \vee \overline{z} = 1$ is a constant, so we stop here for the expansion of $f|_{x=0}$. So, an orthogonal DNF of f is given by:

$$\begin{aligned} f(x,y,z) &= xf|_{x=1} \vee \overline{x}f|_{x=0} \\ &= x(yf|_{x=1,y=1} \vee \overline{y}f|_{x=1,y=0}) \vee \overline{x}(yf|_{x=0,y=1} \vee \overline{y}f|_{x=0,y=0}) \\ &= x(y1 \vee \overline{y}z) \vee \overline{x}(y\overline{z} \vee \overline{y}1) \\ &= xy \vee x\overline{y}z \vee \overline{x}y\overline{z} \vee \overline{x}\,\overline{y} \end{aligned}$$

Another orthogonal DNF of f is

$$xy \vee \overline{x}\,\overline{z} \vee \overline{y}z$$

However, this DNF cannot be obtained from successive Shannon expansions since there is no variable common to all three terms. △

Let f be a Boolean function on $\mathcal{B}^n$ and let $k \in [n]$. The variable x_k is *inessential* for f, or that f does not *depend* on x_k if $f|_{x_k=1}(X) = f|_{x_k=0}(X)$ for all $X \in \mathcal{B}^{n-1}$. That is, the value of f is the same, regardless of the value of x_k.

Theorem 1.20. *Let f be a Boolean function on $\mathcal{B}^n$ and let $k \in [n]$. Then, the following are equivalent:*

(*i*) *The variable x_k is inessential for f;*

(*ii*) *The variable x_k does not appear in any prime implicant of f;*

(*iii*) *f has a DNF representation in which the variable x_k does not appear.*

Proof. (*ii*) → (*iii*) since any function can be represented by a DNF of its prime implicants (Theorem 1.12), and (*iii*) → (*i*) by Theorem 1.18.

Now, suppose the variable x_k is inessential for f, and consider an implicant $C_{AB} = \bigwedge_{i\in A} x_i \bigwedge_{j\in B} \overline{x}_j$ of f. Suppose that $k \in A$ (the argument being symmetric for $k \in B$), and consider the conjunction C obtained by deleting x_k from C_{AB}:

$$C = \bigwedge_{i\in A\setminus\{k\}} x_i \bigwedge_{j\in B} \overline{x}_j$$

We claim that C is an implicant of f, and therefore any prime implicant need not involve x_k. Let $X = (x_1, \ldots, x_n) \in \mathcal{B}^n$ such that $C(X) = 1$. Since neither C nor f depend on x_k, we may suppose that $x_k = 1$ in X. Then, $C(X) = X_{AB}(X) = 1$, and hence $f(X) = 1$. ■

It should be clear that any particular representation of a Boolean function may involve a variable that the function does not depend on.

Example. The DNF $\phi(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_1\overline{x}_2 \vee \overline{x}_1x_2 \vee \overline{x}_1\overline{x}_2$ represent the constant function $\mathbf{1}_4$. In particular, ϕ does not depend on any of its variables. △

Finally, let us mention an interesting connection between essential variables and Chow parameters.

Theorem 1.21. *Let f be a Boolean function on $\mathcal{B}^n$, let $(\omega_1,\ldots,\omega_n,\omega)$ be its vector of Chow parameters, and let $k \in [n]$. Then, if the variable x_k is inessential for f, then $\omega = 2\omega_k$.*

Proof. The sets $A = \{X \in T(f) : x_k = 1\}$ and $B = \{X \in T(f) : x_k = 0\}$ of true points with $x_k = 1$ and $x_k = 0$, respectively, partition $T(f)$, with $|A| = \omega_k$ and $|B| = \omega - \omega_k$. If x_k is inessential, then $|A| = |B|$, so $\omega = 2\omega_k$. ■

The converse fails in general, however. For instance, the function $f(x_1,x_2) = x_1\overline{x}_2 \vee \overline{x}_1x_2$ has Chow parameters (1,1,2), and both variables x_1,x_2 are essential.

1.11 Monotone Boolean Functions

Let f be a Boolean function on $\mathcal{B}^n$ and let $k \in [n]$. We say that f is *positive* in the variable x_k if $f|_{x_k=0} \leq f|_{x_k=1}$. Dually, f is *negative* in x_k if $f|_{x_k=1} \leq f|_{x_k=0}$. More generally, f is *monotone* in x_k if it is positive or negative in x_k.

To check that a variable is positive or negative, we can just check that flipping that variable from 0 to 1 does not decrease the value of the function.

Example. Consider the function:

x,y,z	$f(x,y,z)$
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

Notation: $\text{input} \xrightarrow{\text{output}\to\text{output}} \text{input}$.

- x is positive, since we have $000 \xrightarrow{0\to0} 100$, $001 \xrightarrow{0\to1} 101$, $010 \xrightarrow{0\to1} 110$, and $011 \xrightarrow{0\to1} 111$ all non-decreasing;
- y is positive, as we have $000 \xrightarrow{0\to0} 010$, $001 \xrightarrow{0\to1} 011$, $100 \xrightarrow{0\to1} 110$, and $101 \xrightarrow{1\to1} 111$ all non-decreasing;
- z is positive, as we have $000 \xrightarrow{0\to0} 001$, $010 \xrightarrow{0\to0} 011$, $100 \xrightarrow{0\to1} 101$, and $110 \xrightarrow{1\to1} 111$ all non-decreasing.

△

The function f is furthermore *positive* (respectively, *negative*) if it is positive (respectively, negative) in every variable, and *monotone* if it is positive or negative in each variable (not necessarily all positive or all negative).

Example. The function above is positive, since it is positive in all of its variables. △

Theorem 1.22. *Let f be the Boolean function on $\mathcal{B}^n$, and let g be the function defined by*

$$g(x_1,x_2,\ldots,x_n) = f(\overline{x}_1,x_2,\ldots,x_n)$$

Then, g is negative in x_1 if and only if f is positive in x_1.

Proof. If f is positive in x_1, then changing x_1 from 0 to 1 does not decrease the value of f. But changing x_1 from 0 to 1 in f is same as changing x_1 from 1 to 0 in g, and g shares the same output values as f when x_1 is reversed, so changing x_1 from 1 to 0 does not decrease the value of g, i.e. g is negative in x_1. ∎

Theorem 1.23. *A Boolean function f on $\mathcal{B}^n$ is positive if and only if $f(X) \leq f(Y)$ for all $X,Y \in \mathcal{B}^n$ such that $X \leq Y$.*

Proof. Suppose that f is positive, so $f|_{x_k=0} \leq f|_{x_k=1}$ for all $k \in [n]$, and let $X,Y \in \mathcal{B}^n$ with $X \leq Y$. Consider the sequence of Boolean points

$$X = Z^1 \leq Z^2 \leq \cdots \leq Z^k = Y$$

where Z^{i+1} is obtained from Z^i by flipping the first bit of Z^i that disagrees with Y, say in the k_ith position. Note that since $X \leq Y$, all such flip change a 0 to a 1. Then, since f is positive in each variable and in particular in x_{k_i}, we have

$$f(Z^i) = f|_{x_{k_i}=0}(Z^{i+1}) \leq f|_{x_{k_i}=1}(Z^{i+1}) = f(Z^{i+1})$$

so by induction,

$$f(X) = f(Z^1) \leq f(Z^2) \leq \cdots \leq f(Z^k) = f(Y)$$

Conversely, suppose $f(X) \leq f(Y)$ whenever $X \leq Y$. Let $k \in [n]$, and let $\tilde{X} = (x_1, \ldots ,\widehat{x_k}, \ldots ,x_n) \in \mathcal{B}^{n-1}$ be any assignment of values to all variables apart from x_k. Then, the Boolean points $X,Y \in \mathcal{B}^n$ obtained from $\tilde{X}$ by assigning the value $x_k = 0$ and $x_k = 1$, respectively,satisfy $X \leq Y$, and thus,

$$f|_{x_k=0}(\tilde{X}) = f(X) \leq f(Y) = f|_{x_k=1}(\tilde{X})$$

so $f|_{x_k=0} \leq f|_{x_k=1}$. ∎

Let $\psi(x_1, \ldots ,x_n)$ be a DNF and let $k \in [n]$. Then,

- ψ is *positive* (respectively, *negative*) in the variable x_k if the complemented literal $\overline{x}_k$ (respectively, uncomplemented literal x_k) does not appear in ψ;
- ψ is *monotone* in x_k if ψ is either positive or negative in x_k;
- ψ is *positive* (respectively, *negative*) if it is positive (respectively, negative) in all of its variables;
- ψ is *monotone* if it is positive or negative in each of its variables.

Example.

- Every elementary conjunction is monotone since a variable appears in it at most once.
- The DNF $\phi(x,y,z) = xy \vee x\overline{y}\,\overline{z} \vee xz$ is positive in x, and is neither positive nor negative (i.e. is not monotonic) in x and y.
- The DNF $\psi(x,y,z) = xy \vee x\overline{z} \vee y\overline{z}$ is positive in x, positive in y, and negative in z and is thus monotone, but neither negative nor positive. △

Every positive DNF represents a positive function: since all literals are positive, increasing any input variable from 0 to 1 cannot cause a term to decrease.

However, the converse is not true in general: a non-positive, or even non-monotone DNF may represent a positive function. For instance, $\phi(x,y,z) = xy \vee x\overline{y}\,\overline{z} \vee xz$ represents the positive function $f(x,y,z) = x$.

The next theorem characterising positive variables closely mirrors a previous result characterising inessential variables, Theorem 1.20, in the sense that f being positive in x_k means that the negative literal $\overline{x}_k$ is "inessential" in f.

Theorem 1.24. *Let f be a Boolean function on $\mathcal{B}^n$ and let $k \in [n]$. Then, the following are equivalent:*

(i) *f is positive in x_k;*

(ii) *The literal $\overline{x}_k$ does not appear in any prime implicant of f;*

(iii) *f has a DNF representation in which the literal $\overline{x}_k$ does not appear.*

Proof. As for Theorem 1.20, $(ii) \to (iii)$ since any function can be represented by a DNF of its prime implicants (Theorem 1.12), and $(iii) \to (i)$ by Theorem 1.18, since if $\phi = \bigvee_{j=1}^{m} C_j$ is any DNF representing f where $\overline{x}_k$ does not appear, then the substitution $x_k = 1$ has no effect on any terms involving x_k, while the substitution $x_k = 0$ deletes any such terms, and hence $f|_{x_k=0} \leq f|_{x_k=1}$.

Now, suppose f is positive in x_k and consider an prime implicant $C_{AB} = \bigwedge_{i\in A} x_i \bigwedge_{j\in B} \overline{x}_j$ of f. If $k \notin B$, then we are done, so otherwise suppose $k \in B$ and consider the conjunction C obtained by deleting $\overline{x}_k$ from C_{AB}:

$$C = \bigwedge_{i\in A} x_i \bigwedge_{j\in B\setminus\{k\}} \overline{x}_j$$

Since C_{AB} is prime, C is not an implicant of f, so there exists a point $X = (x_1, \ldots ,x_n) \in \mathcal{B}^n$ such that $C(X) = 1$ but $f(X) = 0$. Since C_{AB} is an implicant of f, we must have $C_{AB}(X) = 0$ and hence $x_k = 1$. Now consider the point $Y \in \mathcal{B}^n$ equal to X in every component apart from the kth, where $y_k = 0$. Then, $C_{AB}(Y) = 1$, and hence $f(Y) = 1$, contradicting that f is positive in x_k. ■

Corollary 1.24.1. *A Boolean function is positive if and only if it can be represented by an expression with no complemented variables.*

Theorem 1.25. *Let ϕ and ψ be DNFs and suppose that ψ is positive. Then, ϕ implies ψ if and only if each term of ϕ is absorbed by some term of ψ.*

Proof. Suppose without loss of generality that ϕ and ψ are expressions in the same n variables. The forward direction follows immediately from Theorem 1.10. Conversely, suppose ϕ implies ψ and consider some term of ϕ, say,

$$C_k = \bigwedge_{i\in I} x_i \bigwedge_{j\in B} \overline{x}_j$$

Let e_A be the characteristic vector of A, so $C_k(e_A) = \phi(e_A) = 1$. Since ϕ implies ψ, we have $\phi(e_A) = 1$, and thus some term $C_j = \bigwedge_{i\in F} x_i$ of ψ satisfies $C_j(e_A)$, and thus $F \subseteq A$, so C_j absorbs C_k as required. ■

Theorem 1.26. *The complete DNF of a positive Boolean function f is positive and irredundant, and is furthermore the unique prime DNF of f.*

Proof. Let f be a positive Boolean function with prime implicants $P_1, \ldots ,P_n$, and let $\psi = \bigvee_{i=1}^{m} P_i$ be the complete DNF of f. By Theorem 1.24, ψ is positive. Now, let $\phi = \bigvee_{k=1}^{r} P_k$ be any prime expression of f, where $r \in [m]$. Since $f = \phi = \psi$, we have in particular that ψ implies ϕ, so by the previous theorem, each term of ψ is absorbed by some term of ϕ. In particular, if $m > r$, then P_m must be absorbed by some other prime implicant P_k for $k \leq r$, contradicting the primality of P_m. So, $r = m$, and hence $\psi = \phi$ is irredundant and unique. ■

This theorem shows that the complete DNF provides a "canonical" shortest DNF representation of a positive Boolean function. Since a shortest DNF representation is necessarily prime and irredundant, no other DNF representation of a positive Boolean function can be as short as its complete DNF.

Theorem 1.27. *Let $\phi = \bigvee_{k=1}^{m} \left(\bigwedge_{i \in A_k} x_i \bigwedge_{j \in B_k} \overline{x}_j \right)$ be a DNF representation of a positive Boolean function f. Then, $\psi = \bigvee_{k=1}^{m} \bigwedge_{i \in A_k} x_i$ is a positive DNF representation of f, and the prime implicants of f are the terms of ψ which are not absorbed by other terms of ψ.*

Proof. Since every term of ϕ is absorbed by some term in ψ, i.e. $\bigwedge_{i \in A_k} x_i \bigwedge_{j \in B_k} \overline{x}_j$ by $\bigwedge_{i \in A_k} x_i$ and ψ is positive, $\phi = f$ implies ψ. For the reverse inequality, consider any point $X = (x_1, \ldots, x_n) \in \mathcal{B}^n$ such that $\psi(X) = 1$. Then, there is a term of ψ that takest the value 1 at X, or equivalently, one of the terms defined by A_k has $x_i = 1$ for all $i \in A_k$. Let e_{A_k} be the characteristic vector of this A_k, so $\phi(e_{A_k}) = f(e_{A_k}) = 1$. Moreover, $e_{A_k} \leq X$, and therefore, by the positivity of f, $f(X) = 1$, and hence $\psi \leq f$. So $\psi = f$.

For the second part of the statement, consider the complete DNF ψ^* of f. Since ϕ is positive and $\psi*$ implies ψ, every term of ψ^* is absorbed by some term of ψ. However, the terms of ψ are implicants of f, while the terms of ϕ^* are prime implicants of f, so all prime implicants of f must appear amongst the terms of ψ. ■

Example. As mentioned previously, the DNF $\phi(x,y,z) = xy \vee x\overline{y}\,\overline{z} \vee xz$ represents the positive function $f(x,y,z) = x$. By deleting all the negative literals from ϕ, we obtain the DNF $\psi = xy \vee x \vee xz$. The terms xy and xz are absorbed by x, so they are not prime. The remaining term x is thus the only prime implicant of f. △

Let f be a Boolean function on $\mathcal{B}^n$ and let $X \in T(f)$ be a true point of f. Then, X is a *minimial true point* of f if there is no distinct true point $Y \neq X$ such that $Y \leq X$ (pointwise). Dually, $X \in F(f)$ is a *maximal false point* of f if there is no distinct false point $Y \neq X$ such that $X \leq Y$.

We denote by $\min T(f)$ the set of minimal true points of f, and $\max F(f)$ the set of maximal false points of f.

Theorem 1.28. *Let f be a positive Boolean function on $\mathcal{B}^n$ and let $Y \in \mathcal{B}^n$. Then,*

(*i*) *Y is a true point of f if and only if there exists a minimal true point X of f such that $X \leq Y$;*

(*ii*) *Y is a false point of f if and only if there exists a maximal false point X of f such that $Y \leq X$.*

Proof. The forward implications are trivial in both cases and are independent of the positivity assumption. The reverse implications are straightforward corollaries of the characterisation of positivity as $f(X) \leq f(Y)$ whenever $X \leq Y$. ■

Theorem 1.29. *Let f be a positive Boolean function on $\mathcal{B}^n$, let $C_A = \bigwedge_{i \in A} x_i$ be an elementary conjunction, and let e_A be the characteristic vector of A. Then,*

(*i*) *C_A is an implicant of f if and only if e_A is a true point of f;*

(*ii*) *C_A is a prime implicant of f if and only if e_A is a minimal true point of f.*

Proof.

(*i*) The forward implication is again trivial and independent of the positivity assumption. Conversely, if e_A is a true point of f, then $\bigwedge_{i \in a} x_i \bigwedge_{j \notin A} \overline{x}_j$ is an implicant of f. Then, by the positivity of f, C_A is also an implicant of f by the same reasoning as in the proof of Theorem 1.27.

(*ii*) Let C_B be an elementary conjunction, and let e_B be the characteristic vector of B. Note that $C_A \leq C_B$ if and only if $B \subseteq A$, or equivalently, $e_B \leq e_A$. Together with (*i*), this implies that C_A is a prime implicant of f if and only if e_A is a minimal true point of f.

■

Example. Consider the positive function $f(x,y,z,w) = xy \vee xzw \vee yz$. Each term is a implicant, so the indicator vectors 1100, 1011, and 0110 are true points of f.

Note that the positivity requirement in this theorem is essential:

(*i*) The function $g(x,y) = x\overline{y}$ has true point 10, but x is not an implicant of g;

(*ii*) The function $h(x,y,z) = xy \vee \overline{x}\,\overline{z}$ has prime implicants xy and $\overline{x}\,\overline{z}$ with the corresponding true points 110 and 101, but 000 is the unique minimal true point of g.

△

A dual correspondence also holds between maximal false points and prime implicates of a positive function:

Theorem 1.30. *Let f be a positive Boolean function on $\mathcal{B}^n$, let $D_A = \bigvee_{i \in A} x_i$ be an elementary disjunction, and let $e_{[n]\setminus A}$ be the characteristic vector of $[n] \setminus A$. Then,*

(*i*) *D_A is an implicate of f if and only if $e_{N\setminus A}$ is a false point of f;*

(*ii*) *D_A is a prime implicate of f if and only if $e_{N\setminus A}$ is a maximal false point of f.*

Proof. The structure of the previous proof also suffices for this result with minor modifications. Alternatively, De Morgan's laws and simple duality arguments can be applied to the previous theorem statement. ■

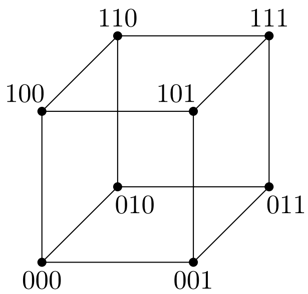
Example. Again, consider the positive function $f(x,y,z,w) = xy \vee xzw \vee yz$. Its prime implicates are $x \vee y$, $x \vee z$, $y \vee z$, and $y \vee w$, so the complementary indicator vectors 0011, 0101, 1001, and 1010 are maximal false points of f. △

1.12 Other Representations of Boolean Functions

Boolean functions can be represented in many other ways than just truth tables and Boolean expressions. In this section, we briefly outline some other representations.

1.12.1 Geometric Interpretation

The *hypercube* Q_n is the graph with vertex set $\mathcal{B}^n$, where two vertices are adjacent if and only if they differ in one coordinate.



The hypercube Q_3.

A subset $S \subseteq \mathcal{B}^n$ is a *subcube* of Q_n if $|S| = 2^k$ for some $k \leq n$, and there are $n - k$ coordinates in which all the vectors of S coincide. That is, a subcube (of dimension k) is obtained from Q_n by fixing $n - k$ coordinates to 0 or 1.

Lemma 1.31. *Let $C = \bigwedge_{i \in A} x_i \bigwedge_{j \in B} \overline{x}_j$ be an elementary conjunction of length $k = |A \cup B| \leq n$. Then, the set of true points of C consists of 2^{n-k} points and defines a subcube of Q_n of codimension k.*

Proof. Let $F = A \cup B$ be the set of (indices of) variables present in C. Then, changing any of the $n-k$ variables not in F does not affect the value of C, so there are 2^{n-k} true points of C. Moreover, these true points all agree in the $n-k$ coordinates fixed by F and hence describe a subcube of codimension k. ■

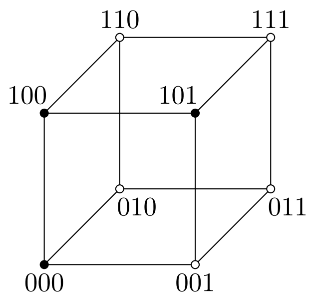
Example. The elementary conjunction $\overline{x}_1 x_3$ on $\mathcal{B}^3$ has true points 001 and 011, which has first and last coordinates fixed and hence describes a codimension-2 subcube. △

Corollary 1.31.1. *There is a bijection between elementary conjunctions and subcubes of Q_n.*

Proof. The correspondence described in the previous theorem is a bijection. ■

We can represent a Boolean function f on $\mathcal{B}^n$ by colouring the vertices corresponding to true points white, and vertices corresponding to false points black.

Example. The function f on $\mathcal{B}^3$ with true points $T(f) = \{001,010,011,110,111\}$ is represented by:

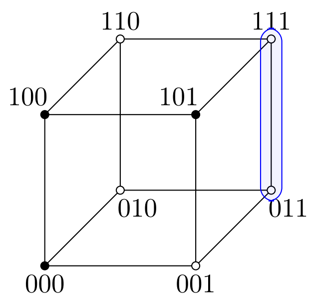


aaaaaaaaaaaaaaaaaa

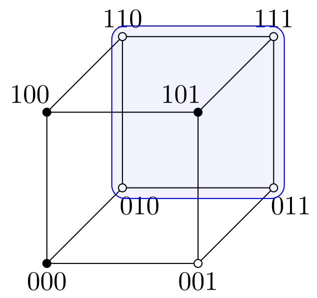
△

In view of the previous discussion, an implicant of f then corresponds to a subcube of Q_n that does not contain any false points, and is furthermore prime if it is maximal with this property, i.e. is not contained inside a larger subcube without false points.

Example. The expression x_2x_3 (i.e. the vertices of the form _11) is an implicant of the function f above, since it defines the subcube:



However, it is not prime, since it is contained in the subcube



△

Let $\phi = \bigvee_{k=1}^{m} C_k$ be a DNF representing f. The set of true points of f then coincides with the union of the sets of true points of the terms C_k. So, a DNF representing f can be viewed as a collection of subcubes of Q_n that cover precisely the true points of f. In particular, an orthogonal DNF is one for which the subcubes in the collection are all disjoint.

1.12.2 Representations of Boolean Functions over GF(2)

The *exclusive-or function*, *Boolean XOR*, or *parity function* is the Boolean function $\oplus : \mathcal{B}^n \to \mathcal{B}$ defined by

$$\oplus(x_1,x_2) \coloneqq x_1\overline{x}_2 \vee \overline{x}_1 x_2$$

We write this function in infix notation as $x_1 \oplus x_2$. When viewed as a binary operation, $\oplus$ is commutative and associative, and the iteration

$$f(x_1,\dots,x_n) = x_1 \oplus \cdots \oplus x_n$$

takes the value 1 precisely when the number of 1s in $(x_1,\dots,x_n)$ is odd. Moreover, $\oplus$ defines addition modulo 2 in the Galois field $\mathrm{GL}(2) = \big(\{0,1\}, \oplus, \wedge\big) \cong \mathbb{Z}/2$.

Theorem 1.32. *For every Boolean function f on $\mathcal{B}^n$, there exists a unique mapping $c : \mathcal{P}\big([n]\big) \to \{0,1\}$ such that*

$$f(x_1,\dots,x_n) = \bigoplus_{A \in \mathcal{P}\left([n]\right)} c(A) \prod_{i \in A} x_i$$

Proof. We provide a constructive proof from first principles. To establish the existence of this representation, we induct on the dimension n. For $n = 1$, such a representation exists since $x = x$ and $\overline{x} = x \oplus 1$. Then, for $n > 1$, existence of the representation follows from the trivial identity,

$$f = f|_{x_n=0} \oplus x_n f|_{x_n=0} \oplus x_n f|_{x_n=1}$$

Indeed, by the inductive hypothesis, $f|_{x_n=1}$ and $f|_{x_n=1}$ have representations of the required form, and hence, after removing any duplicated terms with the identity $x \oplus x = 0$, f also has a representation of this form.

For uniqueness, it suffices to observe that there are exactly 2^{2^n} expressions of this form, and this is also the number of Boolean functions on $\mathcal{B}^n$. ■

These representations of Boolean functions over GL(2) are sometimes called *Zhegalkin polynomials*, *Reed-Muller expansions*, or *algebraic normal forms* (ANF).

To compute the Zhegalkin polynomials of a function, we proceed essentially by comparing coefficients.

Example. We represent the function $f = (x_1 \vee x_2)(x_2 \vee \overline{x}_3)$ as a Zhegalkin polynomial.

Let us write $c_{123} = c(\{x_1,x_2,x_3\})$, etc. for the coefficient mappings, so the general form of a Zhegalkin polynomial on three variables is:

$$(x_1 \vee x_2)(x_2 \vee \bar{x}_3) = c_{123}x_1x_2x_3 \oplus c_{12}x_1x_2 \oplus c_{13}x_1x_3 \oplus c_{23}x_2x_3 \oplus c_1x_1 \oplus c_2x_2 \oplus c_3x_3 \oplus c_0$$

Now, compare coefficients by evaluating each side at each Boolean point in $\mathcal{B}^3$:

(*i*) $(x_1,x_2,x_3) = (0,0,0)$
$$0 = c_0$$

(*ii*) $(x_1,x_2,x_3) = (0,0,1)$
$$0 = c_3$$

(*iii*) $(x_1,x_2,x_3) = (0,1,0)$
$$1 = c_2$$

(*iv*) $(x_1,x_2,x_3) = (0,1,1)$
$$\begin{aligned} 1 &= c_{23} \oplus c_2 \oplus c_3 \\ 1 &= c_{23} \oplus 1 \oplus 0 \\ 0 &= c_{23} \end{aligned}$$

(*v*) $(x_1,x_2,x_3) = (1,0,0)$
$$1 = c_1$$

(*vi*) $(x_1,x_2,x_3) = (1,0,1)$
$$\begin{aligned} 0 &= c_{13} \oplus c_1 \oplus c_3 \\ 0 &= c_{13} \oplus 1 \oplus 0 \\ 1 &= c_{13} \end{aligned}$$

(*vii*) $(x_1,x_2,x_3) = (1,1,0)$
$$\begin{aligned} 1 &= c_{12} \oplus c_1 \oplus c_2 \\ 1 &= c_{12} \oplus 1 \oplus 1 \\ 1 &= c_{12} \end{aligned}$$

(*viii*) $(x_1,x_2,x_3) = (1,1,1)$
$$\begin{aligned} 1 &= c_{123} \oplus c_{12} \oplus c_{13} \oplus c_{23} \oplus c_1 \oplus c_2 \oplus c_3 \\ 1 &= c_{123} \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \\ 1 &= c_{123} \end{aligned}$$

$$(x_1 \vee x_2)(x_2 \vee \bar{x}_3) = x_1x_2x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1 \oplus x_2$$

△

A Boolean function f on $\mathcal{B}^n$ is *linear* if there are coefficients $c_1,\dots,c_n \in \{0,1\}$ such that

$$f(x_1,\dots,x_n) = c_0 \oplus \bigoplus_{i=1}^{n} c_ix_i$$

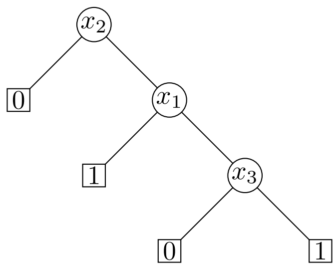
That is, each term is at most linear (i.e. it is an affine combination of variables).

1.12.3 Decision Trees

A *decision tree* is a rooted directed binary tree in which every non-leaf vertex v is labelled by a variable $x_{j(v)}$ for some labelling function $j : V \to [n]$, and every leaf vertex is labelled by the constants 0 or 1.

Every decision tree D corresponds to a Boolean function $\phi_D : \mathcal{B}^n \to \mathcal{B}$ as follows. Let $x = (x_1, \ldots, x_n)$ be a binary vector. Starting from the root, we move between vertices on the tree, following the left arc out of v if $x_{j(v)} = 0$, and the right arc otherwise, and stop when we arrive at a leaf, in which case we say that x is *classified* into this leaf. The label of the leaf then defines the value of $\phi_D(x)$.

Example. The decision tree



represents the function:

x_1,x_2,x_3	$f(x_1,x_2,x_3)$
000	0
001	0
010	1
011	1
100	0
101	0
110	0
111	1

△

A decision tree can be converted into a DNF as follows:

- For each leaf vertex v with label 1, construct the term corresponding to the path from the root to v, where a left arc corresponds to a complemented variable and a right arc to an uncomplemented variable;
- Take the disjunction of all such terms.

Example. For the tree above, the left leaf node with label 1 has a left arc from x_1, and a right arc from x_2, so the corresponding term is $\overline{x}_1 x_2$. The rightmost leaf node similarly has term $x_1 x_2 x_3$. So, the DNF is given by

$$\overline{x}_1 x_2 \vee x_1 x_2 x_3$$

△

We can also easily find the DNF for $\overline{f}$ by performing this algorithm on the leaves with label 0 instead.

Similar to DNF representations, decision tree representations are not unique.

2 Duality Theory

Recall that the dual f^d of a Boolean function $f(x_1, \ldots, x_n)$ is the function

$$f^d(x_1, \ldots, x_n) = \overline{f(\overline{x}_1, \ldots, \overline{x}_n)}$$

Lemma 2.1. *$g = f^d$ if and only if $f(X) \vee g(\overline{X}) = 1$ and $f(X) \wedge g(\overline{X}) = 0$ for all $X \in \mathcal{B}^n$.*

Lemma 2.2. *$f \leq g$ if and only if $g^d \leq f^d$.*

Theorem 2.3. *Let $\phi = \bigvee_{k=1}^{m} \left(\bigwedge_{i \in P_k} x_i \bigwedge_{j \in N_i} \overline{x}_j \right)$ be a DNF of a Boolean function f, and let $C_{PN} = \bigwedge_{i \in P} x_i \bigwedge_{j \in N} \overline{x}_j$ be an elementary conjunction. Then,*

(i) C_{PN} is an implicant of f^d if and only if

$$(P \cap P_k) \cup (N \cap N_i) \neq \varnothing$$

for all $k \in [m]$;

(ii) C_{PN} is a prime implicant of f^d if and only if (i) holds , for every $P' \subseteq P$ and $N' \subseteq N$ with $P' \cup N' \neq P \cup N$, there exists an index $k \in [m]$ such that $(P' \cap P_k) \cup (N' \cap N_k) = \varnothing$.

Proof.

(*i*) By the definition of a dual function, $C_{PN} = \bigwedge_{i \in P} x_i \bigwedge_{j \in N} \overline{x}_j$ is an implicant of f^d if and only if $C_{NP} = \bigwedge_{i \in P} \overline{x}_i \bigwedge_{j \in N} x_j$ is an impcilant of $\overline{f}$. Since $f \wedge \overline{f} = 0$, the identity $C_{P_i N_j} \wedge C_{NP} = 0$ must hold, and hence $(P \cap P_i) \cup (N \cap P_i) \neq \varnothing$ for all $i \in [m]$.

Conversely, if this intersection is non-empty, then $f \wedge C_{NP} = 0$ holds identicallyh, so C_{NP} is an implicant of $\overline{f}$.

(*ii*) This follows from the defition of prime implicants.

■

Note that in the previous theorem, the conjunctions $C_{P_i N_i}$ could have been taken to be prime implicants, rather than arbitrary implicants of f.

2.1 Dual-comparable Functions

A Boolean function f is *dual-minor* if $f \leq f^d$, *dual-major* if $f \geq f^d$, and *self-dual* if $f = f^d$.

Example. The function $f = x_1 x_2 x_3$ is dual-minor, with dual $f^d = x_1 \vee x_2 \vee x_3$:

x_1,x_2,x_3	f	f^d
000	0	0
001	0	1
010	0	1
011	0	1
100	0	1
101	0	1
110	0	1
111	1	1

Equivalently, $f^d \geq (f^d)^d = f$ is dual-major.

The function $g = x_1 x_2 \overline{x}_3 \vee x_1 \overline{x}_2 x_3 \vee \overline{x}_1 x_2 x_3 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3$ has dual

$$\begin{aligned} g^d &= (x_1 \vee x_2 \vee \overline{x}_3)(x_1 \vee \overline{x}_2 \vee x_3)(\overline{x} \vee x_2 \vee x_3)(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3) \\ &= x_1 x_2 \overline{x}_3 \vee x_1 \overline{x}_2 x_3 \vee \overline{x}_1 x_2 x_3 \vee \overline{x}_1 \overline{x}_2 \overline{x}_3 \\ &= g \end{aligned}$$

so g is self-dual. △

Theorem 2.4. *Suppose that a Boolean function f has a prime implicant of degree 1. Then, f is dual-major. Moreover, f is dual-minor (and hence self-dual) if and only if it has no other prime implicants.*

Proof. Without loss of generality, suppose that x_1 is a prime implicant of f, so $f(x_1, \dots, x_n) = x_1 \vee g(x_2, \dots, x_n)$. Then, $f^d = x_1 g^d$. Since $f = 0$ requires $x_1 = 0$, we have $f^d = 0$ whenever $f = 0$, so f is dual-major.

Now, suppose that f has no other prime implicant. Since x_1 is an implicant, $f = 1$ when $x_1 = 1$; and conversely, no point with $x_1 = 0$ is covered by any implicant, so $f = 0$ if $x_1 = 0$. Hence, $f = x_1$ is a projection, and is in particular self-dual.

Conversely, if f has another prime implicant, then there exists a point $X = (x_1^*, \dots, x_n^*) \in \mathcal{B}^n$ such that $x_1^* = 0$ and $f(x_1^*, \dots, x_n^*) = 1$. But $x_1^* = 0$ implies $f^d(x_1^*, \dots, x_n^*) = 0$, so f is not dual-minor. ■

The following result can be viewed as a restatement of the definition of dual comparisons:

Theorem 2.5. *Let f be a Boolean function on $\mathcal{B}^n$. Then,*

(i) *f is dual-minor if and only if the complement of every true point of f is a false point of f. That is, for all $X \in \mathcal{B}^n$, $f(X) = 1$ implies $f(\overline{X}) = 0$, or equivalently, $f(X)f(\overline{X}) = 0$.*

(ii) *f is dual-major if and only if the complement of every false point of f is a true point of f. That is, for all $X \in \mathcal{B}^n$, $f(X) = 0$ implies $f(\overline{X}) = 1$, or equivalently, $f(X) \vee f(\overline{X}) = 0$.*

(iii) *f is self-dual if and only if every pair of complementary points contains exactly one true point and one false point of f. That is, for every $X \in \mathcal{B}^n$, $f(X) = 1$ if and only if $f(\overline{X}) = 0$.*

A dual-minor function f is *maximally dual-minor* if there does not exist a distinct dual-minor function $g \neq f$ such that $f \leq g$.

Theorem 2.6. *A Boolean function is self-dual if and only if it is maximally dual-minor.*

Proof. If f is self-dual and g is a dual-minor function such that $f \leq g$, then,

$$g^d \leq f^d = f \leq g \leq g^d$$

so $g^d = f = f^d$ and hence $f = g$ is maximally dual-minor.

Conversely, suppose that f is not self-dual. If f is not dual-minor, we are done. Otherwise, suppose that f is dual-minor, so there exists a point $X^* = (x_1^*, \dots, x_n^*)$ with $f(X^*) = 0$ and $f^d(X^*) = 1$. Without loss of generality, suppose that $x_1^* = 1$, and consider the function $g = f \vee f^d x_1$ (if $X^* = 0$, take $g = f \vee f^d \overline{x}_1$ instead). Clearly, $f \leq g$, and $f \neq g$ since $g(X^*) = 1$. Moreover, g is dual-minor (actually, self-dual):

$$g^d = f^d(f \vee x_1) = f^d f \vee f^d x_1 = g$$

so f is not maximally dual-minor. ■

The dual result followss similarly:

Theorem 2.7. *A Boolean fuction is self-dual if and only if it is minimally dual-major.*

The construction in the proof above can be generalised to yield a simple standand way of associating a self-dual function to any arbitrary Boolean function.

Given a Boolean function f on $\mathcal{B}^n$, the *self-dual extension* of f is the function f^{SD} on $\mathcal{B}^{n+1}$ defined by

$$f^{\mathrm{SD}}(x_1, \dots, x_{n+1}) \coloneqq f(x_1, \dots, x_n)\overline{x}_{n+1} \vee f^d(x_1, \dots, x_n)x_{n+1}$$

Lemma 2.8. *For every Boolean function f, the function f^{SD} is self-dual.*

Proof. The dual of the f^{SD} is:

$$\begin{aligned}(f^{\mathrm{SD}})^d &= \big(f^d(X) \vee \overline{x}_{n+1}\big)\big(f(X) \vee x_{n+1}\big) \\ &= f^d(X)f(X) \vee f(X)\overline{x}_{n+1} \vee f^d(X)x_{n+1} \vee x_{n+1}\overline{x}_{n+1} \\ &= f(X)\overline{x}_{n+1} \vee f^d(X)x_{n+1}\end{aligned}$$

and hence f^{SD} is self-dual. ■

Theorem 2.9. *The mapping $(-)^{\mathrm{SD}} : f \mapsto f^{\mathrm{SD}}$ is a bijection from the set of Boolean functions of n variables and the set of self-dual functions of $n+1$ variables.*

Proof. The mapping $(-)^{\mathrm{SD}}$ is injective, since the restriction of f^{SD} to $x_{n+1} = 0$ is precisely f. Moreover, $(-)^{\mathrm{SD}}$ has an inverse given by $g \mapsto g|_{x_{n+1}=0}$ for every self-dual function g on $\mathcal{B}^n$:

$$\begin{aligned}(g|_{x_{n+1}=0})^{\mathrm{SD}} &= g|_{x_{n+1}=0}\overline{x}_{n+1} \vee (g|_{x_{n+1}=0})^d x_{n+1} \\ &= g|_{x_{n+1}=0}\overline{x}_{n+1} \vee g^d|_{x_{n+1}=0}x_{n+1} \\ &= g|_{x_{n+1}=0}\overline{x}_{n+1} \vee g|_{x_{n+1}=1}x_{n+1}\end{aligned}$$

which is precisely the Shannon expansion of g. ■

When applied to dual-minor functions, the definition of a self-dual extension takes a simpler form:

Theorem 2.10. *If f is dual-minor, then $f^{\mathrm{SD}} = f \vee f^d x_{n+1}$*

Proof. This holds since for all $a,b,x \in \mathcal{B}$, $a \leq b$ implies $a\overline{x} \vee bx = a \vee bx$. ■

2.2 Duality Properties of Positive Functions

Recall that a Boolean function f is positive if and only if $X \leq Y$ implies $f(X) \leq f(Y)$ for all $X,Y \in \mathcal{B}^n$, and if and only if f can be represented by a positive expression, i.e. an expression without complemented variables.

We have also seen that the complete DNF of a positive Boolean function is positive and irredundant, and since the dual of a positive expression is positive, we have:

Theorem 2.11. *A function f is positive if and only if its dual f^d is positive.*

Recall that for a positive function f, we denote by $\min T(f)$ the set of minimal true points of f, and by $\max F(f)$ the set of maximal false points of f. We have seen that an elementary conjunction $C_A = \bigwedge_{i\in A} x_i$ is a prime implicant of f if and only if its characteristic vector e_A is a minimal true point of f. This can dualised for maximal false points as follows:

Theorem 2.12. *Let f be a positive Boolean function on $\mathcal{B}^n$, let $C_A = \bigwedge_{i\in A}$ be an elementary conjunction, and let $e_{[n]\setminus A}$ be the characteristic vector of $[n] \setminus A$. Then, C_A is a prime implicant of f^d if and only if $e_{[n]\setminus A}$ is a maximal false point of f.*

Example. Let $f = x_1 \vee x_2x_3$ be a positive function. Its dual is given by $f^d = x_1(x_2 \vee x_3) = x_1x_2 \vee x_1x_3$, with prime implicants x_1x_2 and x_1x_3. So, the maximal false point of f are given by the complementary indicator vectors 001 and 010. △

We can also characterise dual prime implicants of positive Boolean functions in terms of hypergraphs.

Let $\mathcal{H} = (V,\mathcal{E})$ be a hypergraph. Then, a set $S \subseteq V$ of vertices is:

- *stable* if it does not contain any edge of $\mathcal{H}$;
- a *transversal* of $\mathcal{H}$ if it intersects every edge of $\mathcal{H}$.

A transversal is furthermore *minimal* if it is minimal with respect to inclusion of transversals. A set $E \subseteq \mathcal{E}$ of pairwise disjoint edges is a *matching*.

A hypergraph $\mathcal{H} = (V,\mathcal{E})$ is a *clutter*, *Sperner family*, or a *simple hypergraph* if no edge of $\mathcal{H}$ is a subset of any other edge.

For a positive Boolean function f on $\mathcal{B}^n$, we associate the hypergraph $\mathcal{H}_f = \big([n],\mathcal{P}\big)$ where $\mathcal{P}$ is the collection of sets $P \subseteq [n]$ of indices such that $\bigwedge_{i\in P} x_i$ is a prime implicant of f. $\mathcal{H}_f$ is necessarily a clutter since the implicants are prime.

Theorem 2.13. *Let $f = \bigvee_{P\in\mathcal{P}} \bigwedge_{i\in P} x_i$ and $g = \bigvee_{T\in\mathcal{T}} \bigwedge_{i\in T} x_i$ be the complete DNFs of two positive functions on $\mathcal{B}^n$. Then, the following are equivalent:*

(i) $g = f^d$;

(ii) For every partition of $[n]$ into two disjoint sets A and $\overline{A}$, there is either a member of $\mathcal{P}$ contained in A, or a member of $\mathcal{T}$ containied in $\overline{A}$, but not both.

(iii) $\mathcal{T}$ is precisely the family of minimal transversals of $\mathcal{P}$.

Example. Again, consider the positive function $f = x_1 \vee x_2x_3$ and its dual $f^d = x_1x_2 \vee x_1x_3$. The hypergraph $\mathcal{H}_f$ has the edge set $\mathcal{E} = \big\{\{1\},\{2,3\}\big\}$, and $\{1,2\}$ and $\{1,3\}$ are exactly the minimal transversals of $\mathcal{H}_f$. △

3 Complexity Measures of Boolean Functions

In this section, let f be a Boolean function on $\mathcal{B}^n$ and $X = (x_1,\ldots,x_n) \in \mathcal{B}^n$ be a Boolean point.

3.1 Certificate Complexity

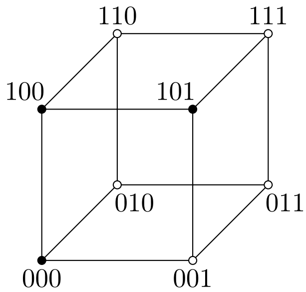
A *certificate* of f on X is a set $S \subseteq [n]$ of indices such that $f(Y) = f(X)$ for every Boolean point Y with $y_i = x_i$ for all $i \in S$. That is, a certificate is a collection of indices sufficient to determine the value of f.

The *size* of a certificate S is its cardinality $|S|$. The *certificate complexity* of f on X, denoted by $C(f,X)$, is the size of a smallest certificate of f on X. The *certificate complexity* $C(f)$ of f is the maximum certificate complexity of f over all Boolean points $X \in \mathcal{B}^n$. The 0-*certificate complexity* $C_0(f)$ of f is the maximum certificate complexity of f over all false points of f, and the 1-*certificate complexity* $C_1(f)$ of f is the maximum taken over all true points.

$$C(f) = \max_{X\in\mathcal{B}^n} C(f,X)$$
$$C_0(f) = \max_{X\in F(f)} C(f,X)$$
$$C_1(f) = \max_{X\in T(f)} C(f,X)$$

Informally, the certificate complexity of a Boolean function f on $\mathcal{B}^n$ at a point X is the codimension of the largest subcube containing X that defines a constant function; the 0-certificate complexity is the maximum codimension taken over all 0 points, and the 1-certificate complexity is the maximum codimension taken over all 1 points.

Example. Consider the function f represented by:



The certificate complexity at 111 is 1, since the largest constant subcube containing 111 is the entire back face, which is of codimension 1. The certificate complexity at 000 is 2 since the largest constant subcube is the edge 000-100 of codimension 2. We also have $C(f) = C_0(f) = C_1(f) = 2$ since every point is contained in a constant edge of codimension 2. △

Lemma 3.1. *The intersection of any 0-certificate and any 1-certificate is empty.*

Proof. Let S_0 be a 0-certificate on a false point X and let S_1 be a 1-certificate on a true point Y. Suppose $S_0 \cap S_1 = \varnothing$, and let Z be a point that coincides with X in all the S_0-positions and coincides with Y in all the S_1-positions. Then, Z is simultaneously true and false, which is impossible. So $S_0 \cap S_1 \neq \varnothing$. ■

Lemma 3.2. *A Boolean function f can be written as a k-DNF (a DNF where every term has at most k literals) if and only if $C_1(f) \leq k$. Similarly, a Boolean function f can be written as a k-CNF if and only if $C_0(f) \leq k$.*

Proof. Let ϕ_f be a k-DNF representing f. For every true point X of f, there is a term T of ϕ_f with $T(X) = 1$. Observe that the set of indices of literals in T is a certificate of X. Since ϕ_f is a k-DNF, then $C_1(f) \leq k$.

Conversely, let $C_1(f) \leq k$. For every true point X, a minimal certificate on X correspont to a maximal subcube containing X, all of whose points are true. This subcube corresponds to a term with at most k literals. The disjunction of all such terms taken over all true points then represents f. The proof for C_0 is similar. ■

3.2 Sensitivity and Block Sensitivity

Given a set $S \subseteq [n]$ of indices, we denote by X^S the Boolean point obtained from X by complementing all the components x_i with $i \in S$. In particular, we abbreviate $X^{\{i\}}$ to X^i.

The *sensitivity* $s(f,X)$ of f on X is the number of indices i such that $f(X) \neq f(X^i)$. The *sensitivity* $s(f)$ of f is the maximum sensitivity over all Boolean points $X \in \mathcal{B}^n$; the 0-*sensitivity* $s_0(f)$ is the maximum sensitivity over all false points of f; and the 1-*sensitivity* $s_1(f)$ is the maximum taken over all true points:

$$
\begin{aligned}
s(f) &= \max_{X \in \mathcal{B}^n} s(f,X) \\
s_0(f) &= \max_{X \in F(f)} s(f,X) \\
s_1(f) &= \max_{X \in T(f)} s(f,X)
\end{aligned}
$$

In terms of the hypercube, the sensitivity of f at a given vertex is the number of neighbouring vertices with a different colour.

Example. For the function f from the previous example, the sensitivity of f at 011 is 0 since it has zero neigbours of differing colours, i.e. complementing any of the bits does not change the value of f. On the other hand, the sensitivity of f at 000 is 2, since flipping the second or third bit changes f from 0 to 1, and flipping the first bit leaves f unchanged. △

The *block sensitivity* $bs(f,X)$ of f on X is the maximum number of disjoint non-empty sets of indices $B_1,\ldots,B_b \subseteq [n]$ called *sensitivity blocks* such that $f(X) \neq f(X^{B_i})$ for all i. The *block sensitivity* $bs(X)$ of f is the maximum block sensitivity over all Boolean points $X \in \mathcal{B}^n$; the 0-*block sensitivity* $s_0(f)$ is the maximum block sensitivity over all false points of f; and the 1-*block sensitivity* $s_1(f)$ is the maximum taken over all true points:

$$\begin{aligned} bs(f) &= \max_{X\in\mathcal{B}^n} bs(f,X) \\ bs_0(f) &= \max_{X\in F(f)} bs(f,X) \\ bs_1(f) &= \max_{X\in T(f)} bs(f,X) \end{aligned}$$

Example. For the same function f as in previous example, the sensitivity of f at 100 is 1 since f only changes if we flip the middle bit, but the block sensitivity of f at 100 is 2, since we have the blocks $B_1 = \{1,3\}$ and $B_2 = \{2\}$ (yielding 001 and 110, respectively). △

Lemma 3.3. *For any Boolean function f,*

$$s(f) \leq bs(f) \leq C(f)$$

Proof. For any point $X \in \mathcal{B}^n$, the sensitivity of f on X coincides with the block sensitivity of f on X if we do not allow blocks of size more than 1. Therefore, by allowing blocks of arbitrary size, we cannot decrease the sensitivity and hence $s(f) \leq bs(f)$.

For each Boolean point X, a certificate on X must contain at least one index from each sensitivity block, and hence $bs(f,X) \leq C(f,X)$, so $bs(f) \leq C(f)$. ■

Theorem 3.4. *For any Boolean function f,*

$$C(f) \leq s(f)bs(f)$$

Proof. Consider a point $X \in \mathcal{B}^n$. First, note that if B is a minimal sensitivity block for X, then $|B| \leq s(f)$, since of we flip one of the bits in X^B indexed by B, then the function value must flip from $f(X^B)$ to $f(X)$ since B is minimal, so every coordinate in B is sensitive on X^B. Therefore, $|B| \leq s(f,X^B) \leq s(f)$.

Now, let $B_1,\ldots,B_b$ be disjoint minimal blocks that achieve the block sensitivity $b = bs(f,X) \leq bs(f)$, and let $C = \bigcup_i B_i$. If C is not a certificate for f on X, then there is an index $i \notin C$ such that $f(X) \neq f(X^i)$. But then, $\{i\}$ is a sensitivity block for f on X disjoint from $B_1,\ldots,B_b$, contradicting that $b = bs(f,X)$. Thus, C is a certificate for f on X. By the same argument as above, $|B_i| \leq s(f)$ for all $i \in [b]$, and hence $|C| = |\bigcup_i B_i| \leq s(f)bs(f)$.

Since for each X there is a certificate of size at most $s(f)bs(f)$, we have $C(f) \leq s(f)bs(f)$. s(f)bs(f). ■

It was a long standing open problem, known as the *sensitivity conjecture*, whether there is a constant c such that $bs(f) \leq s(f)^c$ for any Boolean function f. The conjecture was eventually resolved positively in the following:

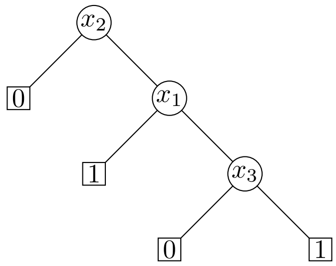
Theorem 3.5 (Huang)**.** *For every Boolean function f,*

$$bs(f) \leq s(f)^4$$

3.3 Decision Tree Complexity

Let t be a decision tree for a Boolean function f. For each point $X \in \mathcal{B}^n$, the number of bits of X examined by t before it is classified is called the *cost* of t on X, denoted $\text{cost}(t,X)$.

Example. Let t be the following decision tree:



The cost of t on 000 is 1, since only x_2 is examined before 000 is immediate classified with 0, while the cost of t on 101 is 3, since x_2 then x_1 must be examined before 101 is classified with 1. △

We denote by $\mathcal{T}$ the set of all decision trees that represent f. Then, the *decision tree complexity* of f is defined as:

$$D(f) = \min_{t \in \mathcal{T}} \max_{X \in \mathcal{B}^n} \text{cost}(t,X)$$

Equivalently, the decision tree complexity of f is the depth of an optimal decision tree that represents f.

Theorem 3.6. *For any Boolean function f,*

$$bs(f) \leq D(f)$$

Proof. Consider a point $X \in \mathcal{B}^n$ with maximally many sensitivity blocks $B_1, \ldots, B_{bs(f)}$. To evaluate f on X, a decision tree must examine at least one index from each block B_i, since other wise we could flip that block without the tree being able to detect this, i.e. the tree would be unable to distinguish $f(X) \neq f(X^{B_i})$. Thus, the tree must make at least $bs(f)$-many queries on X. ■

Theorem 3.7. *For any Boolean function f,*

$$D(f) \leq C_1(f)C_0(f) \leq C(f)^2$$

Proof. Let $k = C_1(f)$ and $\ell = C_0(f)$. Using Lemma 3.2, let D be a k-DNF representation of f and C be an ℓ-DNF representation of f. Take a term T in D, and examine the values of (the at most k) variables in T. Once the values of the variables in T are fixed, we are left with a function f' with fewer variables. Since every clause in C has a variable in common with T by Lemma 3.1, after fixing the values of the variables in T, C transforms into an $(\ell - 1)$-CNF C' representing f'. By induction and Lemma 3.2, $D(f') \leq C_1(f')C_0(f') \leq k(\ell - 1)$, and hence $D(f) \leq k + k(\ell - 1) = k\ell = C_1(f)C_0(f)$. ■

4 Functional Completeness

So far, we have considered the notion of a Boolean expression as compositions defined inductively over the set of three functions; namely conjunction, disjunction, and negation. We have also considered the notion of a Zhegalkin polynomials as expressions defined inductively over the set of functions $\{0,1,\oplus,\wedge\}$.

We now consider expressions defined inductively over arbitrary sets of function, and not necessarily of two variables.

A set of Boolean functions $\{f_1, f_2, \ldots\}$ is (*functionally*) *complete* if any Boolean function can be written as an expression over the functions in the set.

The set of all Boolean functions is trivially complete, but we have also seen that the sets $\{\overline{x}, x_1 \wedge x_2, x_1 \vee x_2\}$ and $\{0, 1, x_1 \oplus x_2, x_1 \wedge x_2\}$ are complete. Obviously, not every set of functions is complete, as, for instance, the set $\{0, 1\}$ is not complete. The following theorem allows us to reduce the question of completeness of some sets of Boolean functions to the same question for other sets of Boolean functions.

Theorem 4.1. *Suppose we have two sets of Boolean functions $\mathcal{F} = \{f_1, f_2, \ldots\}$ and $\mathcal{G} = \{g_1, g_2, \ldots\}$. If the set $\mathcal{F}$ is complete and every function in $\mathcal{F}$ can be represented as an expression over the functions in $\mathcal{G}$, then $\mathcal{G}$ is also complete.*

Proof. Let h be a Boolean function represented as an expression $C[f_1, f_2, \ldots]$ in $\mathcal{F}$. By assumption, every function f_i in $\mathcal{F}$ can be represented as an expression $C_i[g_1, g_2, \ldots]$ over the functions in $\mathcal{G}$. Then, $C[C_1, C_2, \ldots]$ expresses h over the functions in $\mathcal{G}$, so $\mathcal{G}$ is complete. ■

Example.

- The set $\{\overline{x}, x_1 \wedge x_2\}$ is complete, since $x_1 \vee x_2 = \overline{\overline{x_1} \wedge \overline{x_2}}$, and $\{\overline{x}, x_1 \wedge x_2, x_1 \vee x_2\}$ is complete. Dually, the set $\{\overline{x}, x_1 \vee x_2\}$ is complete.
- The set $\{x_1 \uparrow x_2\}$ is famously complete, enabling most modern computer hardware to be built from only NAND gates. To see this, observe that $x \uparrow x = \overline{x}$ and $(x_1 \uparrow x_2) \uparrow (x_1 \uparrow x_2) = x_1 \wedge x_2$. Dually, $\{x_1 \downarrow x_2\}$ is also complete.

△

Let $\mathcal{F}$ be a set of Boolean functions. The *closure* $[\mathcal{F}]$ of $\mathcal{F}$ is the set of all Boolean functions that can be represented as expressions over the functions in $\mathcal{F}$.

Theorem 4.2. *For any sets $\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2$ of functions,*

- $\mathcal{F} \subseteq [\mathcal{F}]$;
- $[[\mathcal{F}]] = [\mathcal{F}]$;
- *If $\mathcal{F}_1 \subseteq \mathcal{F}_2$, then $[\mathcal{F}_1] \subseteq [\mathcal{F}_2]$;*
- $[\mathcal{F}_1] \cup [\mathcal{F}_2] \subseteq [\mathcal{F}_1 \cup \mathcal{F}_2]$.

A set $\mathcal{F}$ of functions is (*functionally*) *closed* if $\mathcal{F} = [\mathcal{F}]$.

Example. The set of all Boolean functions is closed, while the set $\{1, x_1 \oplus x_2\}$ is not. Conversely, the set of all linear functions is closed, since a linear expression of linear expressions is linear. △

We can also characterise completeness in terms of closedness: a set $\mathcal{F}$ is complete if and only if $[\mathcal{F}]$ contains the set of all Boolean functions.

4.1 Important Closed Classes

4.1.1 Functions Preserving Constants

We denote by T_0 the class of Boolean functions that map the constant zero vector $000\ldots0$ to 0. That is, the functions $f(x_1, \ldots, x_n)$ such that

$$f(0, 0, \ldots, 0) = 0$$

Example. The functions 0, x, $x_1 \wedge x_2$, $x_1 \vee x_2$, and $x_1 \oplus x_2$ belong to T_0, while the functions 1 and $\overline{x}$ do not. △

Lemma 4.3. *The class T_0 contains $2^{2^n - 1}$ functions on $\mathcal{B}^n$.*

Proof. We fix the value of the function at 0, and there are $2^n - 1$ other Boolean points, each of which can take 2 values. ∎

Lemma 4.4. *The class T_0 is closed.*

Proof. For any functions $f, f_1, \ldots, f_n \in T_0$, the function $F = f(f_1, \ldots, f_n)$ belongs to T_0:

$$\begin{aligned} F(0,\ldots,0) &= f\big(f_1(0,\ldots,0),\ldots,f_n(0,\ldots,0)\big) \\ &= f(0,\ldots,0) \\ &= 0 \end{aligned}$$

so F preserves 0. ∎

Similarly, we denote by T_1 the class of Boolean functions that send the constant one vector $111\ldots1$ to 1. That is, the functions $f(x_1,\ldots,x_n)$ such that

$$f(1,1,\ldots,1) = 1$$

Example. The functions 1, x, $x_1 \wedge x_2$, and $x_1 \vee x_2$, belong to T_1, while the functions 0, $\overline{x}$ do not. △

Since T_1 consists of functions dual to the functions of T_0, all theorems immediately dualise to T_1:

Corollary 4.4.1. *The class T_1 contains 2^{2^n-1} functions on $\mathcal{B}^n$.*

Corollary 4.4.2. *The class T_1 is closed.*

4.1.2 Self-Dual Boolean Functions

We denote by S the class of all self-dual Boolean functions.

Example. x and $\overline{x}$ are self-dual functions. △

Lemma 4.5. *The class S contains $2^{2^{n-1}}$ Boolean functions on $\mathcal{B}^n$.*

Proof. Self-dual Boolan functions must take opposite values on complementary Boolean points, so a self-dual Boolean function need only be defined on half the Boolean points. ∎

Lemma 4.6. *The class S is closed.*

Proof. For any functions $f, f_1, \ldots, f_n \in S$, the function $F = f(f_1, \ldots, f_n)$ belongs to S:

$$\begin{aligned} F^d &= f^d(f_1^d,\ldots,f_n^d) \\ &= f(f_1,\ldots,f_n) \\ &= F \end{aligned}$$

so F is self-dual. ∎

Lemma 4.7. *If f does not belong to S, then by substituting functions x and $\overline{x}$, it can be transformed into a constant 0 or 1.*

Proof. Since f is not self-dual, there exists a point $X \in \mathcal{B}^n$ such that $f(X) = f(\overline{X})$. For each i, define the functions

$$\phi_i(x) = \begin{cases} x & x_i = 1 \\ \overline{x} & x_i = 0 \end{cases}$$

and consider the function

$$F(x) = f\big(\phi_1(x), \ldots, \phi_n(x)\big)$$

That is, we complement the components corresponding to the non-zero entries of X. Then,

$$\begin{aligned} F(0) &= f\big(\phi_1(0), \ldots, \phi_n(0)\big) \\ &= f(\overline{x}_1, \ldots, \overline{x}_n) \\ &= f(x_1, \ldots, x_n) \\ &= f\big(\phi_1(1), \ldots, \phi_n(1)\big) \\ &= F(1) \end{aligned}$$

so F is a constant. ■

4.1.3 Positive Functions

Recall that a Boolean function f is positive if any of the following equivalent conditions hold:

- The restrictions in every variable x_i satisfy $f|_{x_i=0} \leq f|_{x_i=1}$;
- Whenever $X \leq Y$, $f(X) \leq f(Y)$;
- f has a DNF representation without complemented variables.

We denote by M the class of positive Boolean functions.

Example. 0, 1, x, $x_1 \wedge x_2$, and $x_1 \vee x_2$ are positive functions. △

Lemma 4.8. *The class M is closed.*

Proof. For any functions $f, f_1, \ldots, f_n \in M$, let $F = f(f_1, \ldots, f_n)$. Denote by p_i the number of variables of the function f_i, and by m the number of variables of F, and suppose without loss of generality that F depends only on the variables that appear in the function $f_1, \ldots, f_n$.

For a Boolean point $X \in \mathcal{B}^n$, denote by X^i the projection of X into $\mathcal{B}^{p_i}$ along the variables corresponding to f_i. Note that if $X \leq Y$, then the projections also satisfy $X^i \leq Y^i$. Since the functions $f_1, \ldots, f_n$ are positive, $f_i(X^i) \leq f_i(Y^i)$, so

$$\big(f_1(X^1), \ldots, f_n(X^n)\big) \leq \big(f_1(Y^1), \ldots, f_n(Y^n)\big)$$

and since f is positive, we have

$$F(X) = f\big(f_1(X^1), \ldots, f_n(X^n)\big) \leq f\big(f_1(Y^1), \ldots, f_n(Y^n)\big) = F(Y)$$

so F is positive. ■

Let us call two Boolean points $X, Y \in \mathcal{B}^n$ *neighbouring* if they differ in precisely one coordinate.

Lemma 4.9. *If f does not belong to M, then by substituting the constants* 0 *and* 1 *and the function x, it can be transformed into the function $\overline{x}$.*

Proof. First, we claim that there exist two neighbouring points $X^*, Y^* \in \mathcal{B}^n$ such that $X^* \leq Y^*$, and $f(X^*) > f(Y^*)$. Indeed, since f is not positive, there exist two Boolean points X' and Y' such that $X' \leq Y'$ and $f(X') > f(Y')$, and if X' and Y' are not neighbours and differ in $t > 1$ coordinates, then there is a sequence of Boolean points

$$X' = Z^1 \leq Z^2 \leq \cdots \leq Z^t = Y'$$

where Z^{i+1} is obtained from Z^i by flipping the first bit of Z^i that disagrees with Y, i.e. Z^{i+1} and Z^i are neighbours. Since $f(X') > f(Y')$, there is a pair of consecutive points X^* and Y^* in the sequence above such that $X^* \leq Y^*$, and $f(X^*) > f(Y^*)$. Suppose that $X^* = X^i$, so X^* and Y^* differ in the ith coordinate, and consider the function

$$\phi(x) = f(x_1^*, \ldots, x_{i_1}^*, x, x_{i+1}^*, \ldots, x_n^*)$$

Then,

$$\begin{aligned}\phi(0) &= f(x_1^*, \ldots, x_{i-1}^*, 0, x_{i+1}^*, \ldots, x_n^*)\\ &= f(X^*)\\ &> f(Y^*)\\ &= f(x_1^*, \ldots, x_{i-1}^*, 1, x_{i+1}^*, \ldots, x_n^*)\\ &= \phi(1)\end{aligned}$$

so $\phi(0) = 1$ and $\phi(1) = 0$, i.e. $(x) = \overline{x}$. ■

4.1.4 Linear Functions

Recall that a Boolean function f is linear if it can be expressed in the form

$$f(x_1, \ldots, x_n) = c_0 \oplus \bigoplus_{i=1}^{n} c_n x_n$$

That is, it is a GL(2)-affine combination of variables.

We denote the class of linear Boolean functions by L.

Example. 0, 1, x, $\overline{x} = x \oplus 1$, and $x_1 \oplus x_2$ are linear functions, but $x_1 \wedge x_2$ and $x_1 \vee x_2$ are not. △

Lemma 4.10. *If f does not belong to L, then by substituting the constants 0 and 1 and the functions x and $\overline{x}$, and possibly by negating f, it can be transformed into the function $x_1 \wedge x_2$.*

Proof. Let

$$f(x_1, \ldots, x_n) = \bigoplus_{A \in \mathcal{P}([n])} c(A) \prod_{i \in A} x_i$$

be a Zhegalkin polynomial for f. Since f is not linear, there is a non-zero term in this polynomial of at least quadratic order, involving at least, say, x_1 and x_2. Then, the polynomial can be transformed as

$$\bigoplus_{A \in \mathcal{P}([n])} c(A) \prod_{i \in A} x_i = x_1 x_2 f_1(x_3, \ldots, x_n) \oplus x_1 f_2(x_3, \ldots, x_n) \oplus x_2 f_3(x_3, \ldots, x_n) \oplus f_4(x_3, \ldots, x_n)$$

where $f_1 \neq \mathbf{0}_{n-2}$, since the polynomial is unique. That is, there exist $(a_3, \ldots, a_n) \in \mathcal{B}^{n-2}$ such that $f_1(a_3, \ldots, a_n) = 1$. Now, consider the function

$$f(x_1, x_2) = f(x_1, x_2, a_3, \ldots, a_n) = x_1 x_2 \oplus \alpha x_1 \oplus \beta x_2 \oplus \gamma$$

for some coefficients $\alpha, \beta, \gamma \in \{0, 1\}$. Now, let $\psi(x_1, x_2)$ be the function defined as

$$\psi(x_1, x_2) = \phi(x_1 \oplus \beta, x_2 \oplus \alpha) \oplus \alpha\beta \oplus \gamma$$

Then,

$$\begin{aligned}\phi(x_1 \oplus \beta, x_2 \oplus \alpha) \oplus \alpha\beta \oplus \gamma &= (x_1 \oplus \beta)(x_2 \oplus \alpha) \oplus \alpha(x_1 \oplus \beta) \oplus \beta(x_2 \oplus \alpha) \oplus \gamma \oplus \alpha\beta \oplus \gamma\\ &= x_1 x_2 \oplus \alpha x_1 \oplus \beta x_2 \oplus \alpha\beta \oplus \alpha x_1 \oplus \alpha\beta \oplus \beta x_2 \oplus \alpha\beta \oplus \gamma \oplus \alpha\beta \oplus \gamma\\ &= x_1 x_2\end{aligned}$$

so $\alpha = \beta = \gamma = 0$. To complete the proof it suffices to observe that $x \oplus 0 = x$ and $x \oplus 1 = \overline{x}$. ■

4.2 Post's Theorem

Theorem 4.11 (Post). *A set $F = \{f_1,f_2,\dots\}$ of Boolean functions is complete if and only if it is not a subset of any of the following five closed classes: T_0, T_1, S, M, L.*

Proof. Suppose F is complete, so $[F]$ is the class of all Boolean functions. Now, suppose for a contradiction that $F \subseteq X$ for X being one of the forbidden classes T_0, T_1, S, M, L. But then, $[F] \subseteq [X] = X$, which is a contradiction, since none of the five classes contain all Boolean functions.

Conversely, suppose that F is not contained in any of the forbidden classes. Then, F contains a subset $F' = \{f_0,f_1,f_s,f_m,f_\ell\}$ of 5 (not necessarily distinct) functions witnessing this non-containment, i.e. $f_0 \notin T_0$, $f_1 \notin T_1$, $f_s \notin S$, $f_m \notin M$, $f_\ell \notin L$. Without loss of generality, suppose that these functions all depend on the same set of variables $x_1,\dots,x_n$. We claim that F' is complete.

First, the constants 0 and 1 can be obtained from f_1, f_0, and f_s. If $f_0(1,\dots,1) = 1$, then $\phi(x) = f_0(x,\dots,x)$ is the constant 1, and the function $f_1\big(\phi(x),\dots\phi(x)\big) = f_1(1,\dots,1) = 0$ is the constant 0 function. Otherwise, if $f_0(1,\dots,1) = 0$, then $\phi(x) = f_0(x,\dots,x) = \overline{x}$, and hence by Lemma 4.7 we can use ϕ and f_s to obtain a constant. The second constant can then be obtained from the first by using ϕ.

Now, we can apply Lemma 4.9 using the two constants 0 and 1 and the function f_m to obtain the function $\overline{x}$.

Finally, we can apply Lemma 4.10 using the two constants 0 and 1, and the functions $\overline{x}$ and f_ℓ to constuct the function $x_1 \wedge x_2$.

Since $\{\overline{x},x_1 \wedge x_2\}$ is complete, the set F', and hence F, is also complete. ■

Corollary 4.11.1. *Every closed set of Boolean functions, different from the set of all Boolean functios is contained in one of the classes T_0, T_1, S, M, L.*

A set F of Boolean functions is *precomplete* if F is not complete, but for any Boolean function $f \notin F$, the set $F \cup \{f\}$ is complete. It follows that any precomplete set is closed.

Corollary 4.11.2. *There exist precisely* 5 *precomplete sets of Boolean functions: T_0, T_1, S, M, L.*

Theorem 4.12. *Every complete set F of Boolean functions contains a complete subset of at most* 4 *functions.*

Proof. We have seen in the proof of Post's theorem that F contains a complete subset of at most 5 functions. Moreover, we have seen that the function $f_0 \notin T_0$ does not belong either to S (if $f_0(0,\dots,0) = f_0(1,\dots,1) = 1$) or to $T_1 \cup M$ (if $f_0(0,\dots,0) = 1$, $f_0(1,\dots,1) = 0$). Therefore, either the set $\{f_0,f_1,f_m,f_\ell\}$ or the set $\{f_0,f_s,f_\ell\}$ is complete. ■

This bound is sharp, since the set of functions

$$\{f_1 = x_1x_2, \quad f_2 = 0, \qquad f_3 = 1, \quad f_4 = x_1 \oplus x_2 \oplus x_3\}$$

satisfies $f_3 \notin T_0$, $f_2 \notin S$, $f_4 \notin M$, and $f_1 \notin L$, and is thus complete, but any proper subset is incomplete, since $\{f_2,f_3,f_4\} \subseteq L$, $\{f_1,f_3,f4\} \subseteq T_1$, $\{f_1,f_2,f_4\} \subseteq T_0$, and $\{f_1,f_2f_3\} \subseteq M$.

A subset F' of a closed set F is *complete in* F if $[F'] = F$. That is, every function in F can be represented as an expression over the functions in F'. A *basis* of F is a minimal subset F' complete in F.

Example. From the previous example,

$$\{f_1 = x_1x_2, \quad f_2 = 0, \qquad f_3 = 1, \quad f_4 = x_1 \oplus x_2 \oplus x_3\}$$

is a basis for the set of all Boolean functions.

It is also possible to show that the set $\{0,1,x_1x_2,x_1 \vee x_2\}$ is a basis of M. △

In addition to the main theorem characterising functionally complete sets, Post also proved the following results:

Theorem 4.13. *Every closed class of Boolean functions has a finite basis.*

Theorem 4.14. *The set of all closed classes of Boolean functions is countable.*

We observe that the first of these two theorems implies the second one. However, originally Post proved the second theorem before the first.

5 Quadratic Functions

A DNF

$$\phi(x_1, \ldots, x_n) = \bigvee_{k=1}^{m} \bigwedge_{i \in P_k} x_i \bigwedge_{j \in N_k} \overline{x}_j$$

is *quadratic* if althe its terms are quadratic. That is, if they are conjunctions of at most two literals. A term is called *linear* or *purely quadratic* if it has exactly one or exactly two literals, respectively. Similarly, a CNF is *quadratic* if all its clauses are disjunctions of at most two literals.

A Boolean function f is *quadratic* if it admits a quadratic DNF representation. The function f is *dually quadratic* if it admits a quadratic CNF representation. This is equivalent to f^d being quadratic.

A quadratic Boolean function f is *purely quadratic* if it is not constant and has no linear prime implicant. Equivalently, f is purely quadratic if no linear term appears in any DNF of f.

The next result follows immediately from the definition.

Lemma 5.1. *If f is purely quadratic, then in every quadratic DNF of f, every term is a prime implicant.*

Note that it is possible for a quadratic function to be represented by a DNF of higher degree.

Example. The function

$$f = x_1x_2\overline{x}_3\overline{x}_4 \vee x_1x_2\overline{x}_3x_4 \vee \overline{x}_1x_2\overline{x}_3x_4 \vee x_1x_2x_3 \vee \overline{x}_2\overline{x}_3x_4 \vee \overline{x}_1x_3 \vee \overline{x}_2\overline{x}_4$$

is quadratic, since it also admits the DNF

$$f = x_1x_2 \vee \overline{x}_1x_3 \vee \overline{x}_2\overline{x}_4 \vee \overline{x}_3x_4$$

△

5.1 Quadratic Boolean Functions and Graphs

There are many connections between certain classes of quadratic functions and graphs.

Given any undirected graph $G = (V,E)$, its *stability function* is the quadratic Boolean function given by

$$f_G = \bigvee_{ij \in E} x_ix_j$$

Note that the prime implicants of f are precisely the terms x_ix_j of this DNF, which is also the unique irredundant DNF of f. It follows that this mapping from undirected graphs to positive purely quadratic Boolean functions is a a bijection.

5.1.1 The Matched Graph

Another graph that can be conveniently associated with a quadratic DNF ϕ is the *matched graph* G_ϕ. This undirected graph has vertex set $\{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n\}$ with edges given by

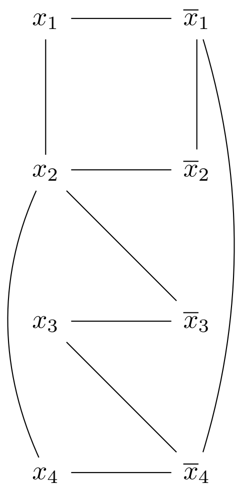
$$\big((x_i,\overline{x}_i) : i \in [n]\big) \cup \big((\alpha,\beta) : \alpha\beta \text{ is a term of } \phi\big)$$

That is, start with the bipartite graph with parts $\{x_1, \ldots, x_n\}$ and $\{\overline{x}_1, \ldots, \overline{x}_n\}$ with complementary vertices matched, then connect vertices that are paired up in terms. Note that if ϕ contains linear terms, a loop (α,α) is added for each such term α.

Example. The matched graph associated to the DNF

$$\phi = x_1x_2 \vee \overline{x}_1\overline{x}_2 \vee \overline{x}_1\overline{x}_4 \vee x_2\overline{x}_3 \vee x_2x_4 \vee x_3\overline{x}_4$$

is given by:



△

The edges of G_ϕ are classified as:

- *positive*, (x_i,x_j);
- *negative*, $(\overline{x}_i,\overline{x}_j)$;
- *mixed*, $(x_i,\overline{x}_j)$;
- *null*, $(x_i,\overline{x}_i)$.

Example. The positive edges are the edges within the left part; the negative edges are the edges within the right part; the mixed edges are the non-horizontal edges between the two parts; and the null edges are the horizontal edges. △

The consistency of the quadratic Boolean equation $\phi = 0$ has a nice graph-theoretic counterpart for G_ϕ as follows.

Let $\mu(G)$ be the maximum cardinality of a matching on G, and $\tau(G)$ be the minimum cardinality of a vertex cover in G. Note that

$$\tau(G) \leq \tau(G)$$

since we need at least one vertex in a minimum vertex cover for every edge in a maximum matching.

The graph G is said to have the *Kőnig–Egerváry* (KE) *property* if this the above expression is in fact an equality.

Theorem 5.2. *The quadratic Boolean equation $\phi = 0$ in n variables is consistent if and only if the matched graph G_ϕ has the Kőnig–Egerváry property.*

Proof. The null edges form a maximum matching in G_ϕ, so G_ϕ has the KE property if and only if there is a vertex cover C in G_ϕ with cardinality n.

Suppose first that G_ϕ has the KE property, and let C be a vertex cover with cardinality n. As every null edge has exactly one endpoint in C, we define the Boolean point $Z = (z_1, \dots, z_n) \in \mathcal{B}^n$ as $z_i = 0$ if and only if $x_i \in C$, and $z_i = 1$ otherwise. Since C is a vertex cover, Z is a solution of the equation $\phi = 0$.

Conversely, let Z be a solution of $\phi = 0$, and let C be the set of vertices x_i for which $z_i = 0$ and $\overline{x}_i$ for which $x_i = 1$. Then, C is a vertex cover of cardinality n, and so G_ϕ has the KE property. ■

That is, we can find a solution to $\phi = 0$ by finding a vertex cover with cardinality n, and taking the complementary indicator vector of the vertex cover.

Example. In the matched graph from the previous example, $\{\overline{x}_1, x_2, \overline{x}_3, x_4\}$ is a vertex cover of cardinality $4 = n$, so $\phi = 0$ is consistent, and in particular, the Boolean point 1010 is a solution. △

5.1.2 The Implication Graph

As an alternative to the matched graph G_ϕ, we can also associate with a quadratic DNF ϕ a directed graph D_ϕ called the *implication graph* of ϕ, and again characterise the consistency of $\phi = 0$ in terms of a simple property of this graph.

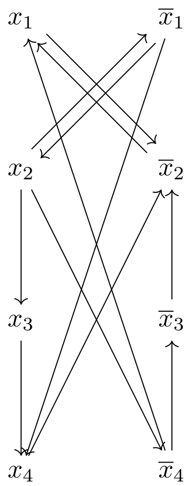
The definition of an implication graph arises from the observation that the relation $\alpha\beta = 0$ is equivalent to the implication $\alpha \Rightarrow \overline{\beta}$, as well as to the implication $\beta \Rightarrow \overline{\alpha}$.

As in the matched graph G_ϕ, the vertices of the implication graph D_ϕ has vertex set $\{x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n\}$. For each quadratic term $\alpha\beta$, we add the arcs $(\alpha, \overline{\beta})$ and $(\beta, \overline{\alpha})$. Either of these arcs is called the *mirror arc* of the other, and the simultaneous presence of these two arcs is called the *mirror property* (MP). Then, for each linear term α, we add the single arc $(\alpha, \overline{\alpha})$.

Example. The graph associated with the DNF

$$\phi = x_1x_2 \vee \overline{x}_1\overline{x}_2 \vee \overline{x}_1\overline{x}_4 \vee x_2\overline{x}_3 \vee x_2x_4 \vee x_3\overline{x}_4$$

is given by:



△

Recall that a *strongly connected component*, or just *strong component*, of a graph $G = (V,E)$ is a maximal subset $C \subseteq V$ of vertices such that every two vertices of C are connected by a path in C in both directions. The strong components of G form a partition of V.

By replacing each strong component of an implication graph D_ϕ by a single vertex, we obtain an acyclic digraph $\hat{D}_\phi$ called the *condensed implication graph* of ϕ. Notice that, because of the mirror propertly, the strong components of D_ϕ come in pairs: if C is a strong component of D_ϕ, then the set $\overline{C}$ of the negation of all literals in C is also a strong component.

Example. The strong components of the implication graph in the previous example are $\{\overline{x}_1, x_2\}$, $\{\overline{x}_2,x_1\}$, and the singletons of every other vertex. △

Lemma 5.3. *An assignment of binary values to the vertices of D_ϕ is a solution of $\phi = 0$ if and only if*

(i) x_i and $\overline{x}_i$ receive complementary values;

(ii) no arc (and hence no direct path) connects from a 1-vertex to a 0-vertex.

Theorem 5.4. *The quadratic Boolean equation $\phi = 0$ in n variables is consistent if and only if no strongly connected component of the implication graph D_ϕ simultaneously contains a literal and its negation.*

Proof. ■

Example. The strong components found in the previous example are either $\{\overline{x}_1, x_2\}$, $\{\overline{x}_2,x_1\}$, or singletons, none of which include a literal and its negation, so ϕ is consistent (agreeing with the previous result from the matched graph). △

The implication graph not only allows us to determine the consistency of the corresponding quadratic Boolean equation, but also, if it is consistent, to infer further properties of its solutions.

A literal α is *forced to value a* for $a \in \{0,1\}$ if either $\phi = 0$ is inconsistent, or if α takes the value a in all possible solutions.

Theorem 5.5. *Suppose that $\phi = 0$ is consistent. Then, the literal α is forced to 0 if and only if there is a directed path from α to $\overline{\alpha}$ in D_ϕ.*

Proof. ■

Example. In the previous example, x_1 is not forced to 0, since there is no directed path from x_1 to $\overline{x}_1$ in D_ϕ. Conversely, x_2 is forced to 0 since there is a directed path $x_2,\overline{x}_4,\overline{x}_3,\overline{x}_2$ from x_2 to $\overline{x}_2$. △

Theorem 5.6. *Let α be a literal not forced to 0 and β be a literal not forced to 1. Then, the relation $\alpha \leq \beta$ holds in all solutions of $\phi = 0$ if and only if there is a directed path from α to β in D_ϕ.*

Proof. ■

Two literals α and β are said to be *twins* if $\alpha = \beta$ in every solution to $\phi = 0$.

Corollary 5.6.1. *Suppose that two literals α and β are not forced. Then, they are twins if and only if they are in the same strong component of D_ϕ.*

5.1.3 More Relations Between Quadratic Equations and Graphs

Recall that an *independent set* in a graph $G = (V,E)$ is a set of vertices such that no two are adjacent, and a *clique* is a set of vertices such that every pair are adjacent, i.e. induces a complete subgraph. A graph is *bipartite* if its vertex set V can be partitioned into two independent sets $V = L \sqcup R$, and is *split* if V can be partitioned into an independent set and a clique $V = I \sqcup C$.

Given a graph $G = (V,E)$, introduce a variable x_i for each vertex $i \in V$. Then, G is bipartite if and only if the quadratic Boolean equation

$$\bigvee_{ij \in E} (x_i x_j \vee \overline{x}_i \overline{x}_j) = 0$$

is consistent. Also, G is split if and only if the quadratic Boolean equation

$$\bigvee_{ij \in E} \overline{x}_i \overline{x}_j \vee \bigvee_{ij \notin E} x_i x_j \vee \overline{x}_i \overline{x}_j = 0$$

is consistent.

6 Horn Functions

An elementary conjunction is a *Horn term* if it contains at most one negated variable. A Horn term is *pure Horn* if it contains precisely one negated variable, and is *positive* otherwise. A DNF is *Horn* if all of its terms are Horn, and a Boolean fnction is a *Horn function* if it can be represented by a Horn DNF.

Lemma 6.1. *The consensus of two Horn terms is Horn. Specifically, the consensus of two pure Horn terms is pure Horn, while the consensus of a positive and a pure Horn term is positive.*

Proof. Let xC and $\overline{x}D$ be two Horn terms that have a consensus. Then, D must only contain positive literals, and C can contain at most one negated variable, which cannot belong to A. Hence their consensus CD contains at most one negated variable, i.e. is Horn, and is positive (respectively, pure Horn) if xC is positive (respectively, pure Horn). ■

Lemma 6.2. *All prime implicants of a Horn function are Horn.*

Proof. Let f be a Horn function and ϕ be a (pure) Horn DNF representing f. Then, we may compute all prime implicants of f by applying the consensus procedure to ϕ. Thus, all prime implicants of f may be obtained by a sequence of consensus operations, starting with the (pure) Horn terms in ϕ, and by the previous lemma, consensus operations preserve (pure) Horn terms, so every prime implicant must also be (pure) Horn. ■

Theorem 6.3. *A Boolean function is Horn if and only if the set of its false points is closed under conjunction.*

Proof. ■

Corollary 6.3.1. *A Boolean function f on $\mathcal{B}^n$ is Horn if and only if $f(X \wedge Y) \leq f(X) \vee f(Y)$ for all $X,Y \in \mathcal{B}^n$.*

Proof. Suppose $f(X \wedge Y) \leq f(X) \vee f(Y)$ for all $X,Y \in \mathcal{B}^n$. Then, $f(X) \vee f(Y) = 0$ if and only if X and Y are false points of f, and $f(X \vee Y) \leq f(X) \vee f(Y) = 0$, so $X \vee Y$ is also a false point of f, so $F(f)$ is closed under conjunction, and hence f is Horn. The same argument in reverse proves the reverse implication. ■

6.1 Horn Boolean Functions and the Union-Closed Sets Conjecture

Let $(U,\mathcal{F})$ be a set system. The family $\mathcal{F}$ is *union-closed* if for any two sets $A,B \in \mathcal{F}$, we have $A \cup B \in \mathcal{F}$. The following conjecture is known as the *union-closed sets conjecture* or *Frankl's conjecture.*

Conjecture 6.1. *Any finite union-closed family $\mathcal{F} \neq \{\varnothing\}$ of finite sets contains an element that belongs to at least of half the sets in the family.*

The family $\mathcal{F}$ is *intersection-closed* if for any two sets $A,B \in \mathcal{F}$, we have $A \cap B \in \mathcal{F}$. Without loss of generality, suppose that every element of the universe appears in at least one set of $\mathcal{F}$. Then, $\mathcal{F}$ is intersection closed if and only if the family of relative complements $\{U \setminus A : A \in \mathcal{F}\}$ is union-closed. So, Frankl's conjecture can be equivalently stated as:

Conjecture 6.2. *Any finite intersection-closed family of at least two finites sets contains an element that belongs to at most half of the sets in the family.*

The conjecture admits many other equivalent formulations, in particular, in the language of lattice and graph theory.

In spite of its simple formulation, the conjecture remains open and has been verified only for special classes of sets, lattices, or graphs. Here, we develop a Boolean approach to the conjecture and verify it for submodular functions.

Let F be an intersection-closed family over the universe $U = x_1,\dots,x_n$ and let $A \in \mathcal{F}$. We represent A by its characteristic vector c_A, i.e. a binary vector with 1 in the ith coordinate if $x_i \in A$, and 0 otherwise. In doing so, we can interpret $\mathcal{F}$ as a Boolean function f over the variables $x_1,\dots,x_n$ whose false points are precisely the elements of $\mathcal{F}$. Then, $\mathcal{F}$ being intersection-closed is equivalent to the false points of f being closed under conjunction, i.e. f is Horn.

We say that a variable x_i *belongs* to a Boolean point X if the ith component of X is 1. Frankl's conjecture can then be restated as follows:

Conjecture 6.3. *Any Horn Boolean function f with at least two false points contains a variable that belongs to at most half of the false point of f.*

Given a Horn Boolean function f, we associate to its set of true points $T = T(f)$ a set system $\mathcal{T}$ over the same universe U such that $A \subseteq U$ is an element of $\mathcal{T}$ if and only if the characteristic vector c_A is a true point of f.

Note that a variable belongs to at most half the false points if and only if it belongs to at least half of the true points of the function, which suggests that the relation between $\mathcal{F}$ and $\mathcal{T}$ is similar to the relation between intersection-closed and union-closed families. However, in general, $\mathcal{T}$ is neither intersection-closed nor union-closed.

In the terminology of set systems, an element that appears in at least half the subsets is *abundant*, and an element that appears in at most half the subsets is *rare.* In the terminology of Boolean functions, every variable that is abundant for true points is rare for false points, and vice versa. In the following results, we will frequently switch between the two roles of the same variable. To avoid ambiguities, we will call a variable abundant in false points, or equivalently, rare in true points, *good.* In this terminology, Frankl's theorem can be restated as:

Conjecture 6.4. *Any Horn Boolean function f with at least two false points contains good variable.*

We will say that a Horn Boolean function *satisfies Frankl's conjecture* if it satisfies this last characterisation of the conjecture. We will now verify Frankl's conjecture for a certain subclass of Horn functions.

A Boolean function $f(X)$ on $\mathcal{B}^n$ is *co-Horn* if $g(X) := f(\overline{X})$ is Horn. In other words, a function is co-Horn if it admits a DNF representation in which every term contains at most one positive literal.

Previous theorems about Horn functions then transform to results about co-Horn functions as follows:

Theorem 6.4. *A Boolean function is co-Horn if and only if the set of its false points is closed under disjunction.*

Corollary 6.4.1. *A Boolean function f on $\mathcal{B}^n$ is co-Horn if and only if $f(X \vee Y) \leq f(X) \vee f(Y)$ for all $X,Y \in \mathcal{B}^n$.*

A Boolean function $f(X)$ is *submodular* if $f(X \vee Y) \vee f(X \wedge Y) \leq f(X) \vee f(Y)$.

Theorem 6.5. *A Boolean function is submodular if and only if it is both Horn and co-Horn. All prime implicants of a submodular function are either linear or quadratic pure Horn.*

Proof. Since $A,B \leq A \vee B$ holds for any Boolean points A,B, $f(X \vee Y) \vee f(X \wedge Y) \leq f(X) \vee f(Y)$ (f is submodular) if and only if $f(X \wedge Y) \leq f(X) \vee f(Y)$ (f is Horn) and $f(X \vee Y) \leq f(X) \vee f(Y)$ (f is co-Horn).

So, if f is submodular, all prime implicants of f are both Horn and co-Horn, so each contains at most one positive and one negative literal, and is thus is either a single variable x_i or its complement $\overline{x}_i$ (i.e. is linear), or is of the form $x_i\overline{x}_j$ (i.e. is quadratic pure Horn). ■

Lemma 6.6. *Let f be a Horn function represented by a Horn DNF D_f. If a variable x_i of f does not appear in D_f negatively, then x_i is a good variable for f.*

Proof. If $f|_{x_i=0}$ does not have true points, then the number of false points of x_i containing x_i is at most equal to the number of false points that do not contain x_i (i.e. if f is identically zero, and is less in any other case), so x_i is a good variable for f.

Conversely, let X be a true point of $f|_{x_i=0}$, i.e. a true point of f with $x_i = 0$, and let t be a term of D_f with $t(X) = 1$. Since x_i does not appear negatively in t and $x_i = 0$, x_i must not appear in t. So, changing the x_i to 1 in X yields a true point of f with $x_i = 1$. This injects the set of true points of $f|_{x_i=0}$ into the set of true points of $f|_{x_i=1}$, so x_i belongs to at least half of the true points of f and is hence good. ■

Theorem 6.7. *Submodular Boolean functions satisfy Frankl's conjecture.*

Proof. ■

7 Threshold Functions

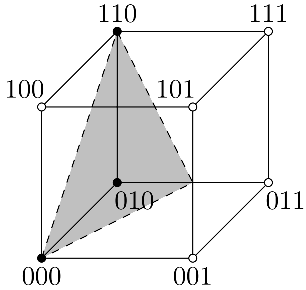
A Boolean function f on $\mathcal{B}^n$ is called a *threshold* or *linearly separable* function if there exist coefficients $w_1,\dots,w_n \in \mathbb{R}$ called *weights* and a *threshold* value $t \in \mathbb{R}$ such that

$$f(x_1,\dots,x_n) = 0 \qquad \iff \qquad \sum_{i=1}^{n} w_i x_i \leq t$$

The hyperplane $\{X \in \mathbb{R}^n : \sum_{i=1}^n w_i x_i \leq t\}$ is called a *separator* of f, and the tuple of weights and threshold value $(w_1,\dots,w_n,t)$ is called a *separating structure* of f. We say that the separator and the separating structure *represent* f.

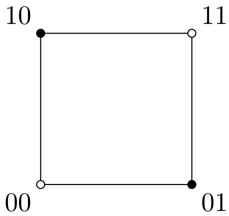
Geometrically, a function is threshold precisely if the set of its true points can be separated from the set of its false points by a hyperplane (where the hyperplane may contain false points).

Example. The function $f(x,y,z) = x\overline{y} \vee z$ is a threshold function with separator $\big\{(x,y,z) \in \mathbb{R}^3 : x - y + 2z = 0\big\}$ and structure $(1,-1,2,0)$.



Note that the separator of a threshold function is not unique, and in fact f in this case admits infinitely many separators.

The function $f(x,y) = xy \vee \overline{x}\,\overline{y}$ is not a threshold function:



The convex hulls of $T(f)$ and $F(f)$ intersect, so they cannot be separated by the hyperplane separation theorem. △

Theorem 7.1. *Every threshold function has an integral separating structure. That is, a separating structure $(w_1,\dots,w_n,t)$ with $w_1,\dots,w_n,t \in \mathbb{Z}$.*

7.1 Basic Properties of Threshold Functions

Theorem 7.2. *Elementary conjunctions and elementary disjunctions represent threshold functions.*

Proof. Given an elementary conjunction $C_{AB} = \bigwedge_{i\in A} x_i \bigwedge_{j\in B} \overline{x}_j$, the equation $\sum_{i\in A} x_i + \sum_{j\in B}(1 - x_j) = |A| - |B| - 1$ defines a separator for C_{AB}. Similarly, given an elementary disjunction $D_{AB} = \bigvee_{i\in A} x_i \bigvee_{j\in B} \overline{x}_j$, the equation $\sum_{i\in A} x_i + \sum_{j\in B}(1 - x_j) = 0$ defines a separator for D_{AB} ■

Geometrically, elementary conjunctions define a single true point of the hypercube, and elementary disjunctions define single false points, which can both clearly be separated from the opposite kind of points.

Another important property of thresfold functions is that they constitute a class of functions closed under restriction.

Theorem 7.3. *If f is a threshold function on $\mathcal{B}^n$ with separating structure $(w_1,\dots,w_n,t)$, then $f_{x_i=1}$ is a threshold function on $\mathcal{B}^{n-1}$ with separating structure $(w_1,\dots,\widehat{w_k}\dots,w_n,t-w_i)$, and $f_{x_i=0}$ is a threshold function on $\mathcal{B}^{n-1}$ with separating structure $(w_1,\dots,\widehat{w_i}\dots,w_n,t)$.*

Proof. Since f is threshold, $f|_{x_i=1}(x_1,\dots,\widehat{x_i},\dots,x_n) = f(x_1,\dots,1,\dots,x_n) = 0$ if and only if

$$\sum_{j\neq i} w_j x_j + w_i \leq t$$
$$\sum_{j\neq i} w_j x_j \leq t - w_i$$

so $f|_{x_i=1}$ is threshold with separating structure $(w_1,\dots,\widehat{w_i}\dots,w_n,t-w_i)$.

Similarly, $f|_{x_i=0}(x_1,\dots,\widehat{x_i},\dots,x_n) = f(x_1,\dots,0,\dots,x_n) = 0$ if and only if

$$\sum_{j\neq i} w_j x_j \leq t$$

so $f|_{x_i=0}$ is threshold with separating structure $(w_1,\dots,\widehat{w_i}\dots,w_n,t)$. ■

Our next observation is that every threshold function is monotone, and hence can be turned into a positive function by "switching" some of its variables. Moreover, the negativity and positivity of each variable is captured in the sign of the corresponding weight.

Theorem 7.4. *Every threshold function is monotone. More precisely, if f is a threshold function with separating structure $(w_1,\dots,w_h,t)$, then for each $i \in [n]$,*

(i) *If $w_i = 0$, then f does not depend on x_i;*

(ii) *If x_i does not depend on x_i, then $(x_1,\dots,w_{i-1},0,w_{i+1},\dots,x_n,t)$ is a separating structure of f;*

(iii) *If $w_i > 0$, then f is positive in x_i;*

(iv) *If f is positive in x_i and f depends on x_i, then $w_i > 0$;*

(v) *If $w_i < 0$, then f is negative in x_i;*

(vi) *If f is negative in x_i and f depends on x_i, then $w_i < 0$;*

(vii) *If $w_j \geq 0$ for $j = 1,\dots,k$, and $w_j < 0$ for $j = k+1,\dots,n$, then the function*

$$g(x_1,\dots,x_n) \coloneqq f(x_1,\dots,x_k,\overline{x}_{k+1},\dots,\overline{x}_n)$$

is a positive threshold function with separating structure

$$\left(w_1,\dots,w_k,-w_{k+1},\dots,-w_n,t-\sum_{j=k+1}^{n} w_j\right)$$

Example. As seen previously, the function $f(x,y,z) = x\overline{y} \vee z$ is a threshold function with separating structure $(1,-1,2,0)$. The associated function $g(x,y,z) = xy \vee z$ with all negations removed is then also a threshold function with separating structure $(1,2,3,2)$. △

We emphasise that a variably may have a non-zero weight in the separating structure of a threshold function even if the function dos not depend on the variable.

Example. The function $f(x,y,z,w) = xy \vee z$ is a threshold function with separating structure $(2,4,6,1,5)$. The variable w is inessential, but has positive weight in this separating structure. △

For three or fewer variables, monotonicity is equivalent to thresholdness. However, this fails in general for functions of more variables.

Example. The functions

$$\begin{aligned} f(x,y,z,w) &= xy \vee zw \\ g(x,y,z,w) &= xy \vee yz \vee zw \\ h(x,y,z,w) &= xy \vee yz \vee zw \vee xw \end{aligned}$$

are positive but not threshold. Up to permutation of their variables, these are the only positive non-threshold functions of four variables. △

Theorem 7.5. *If f is a threshold function on $\mathcal{B}^n$ and $(w_1,\dots,w_n,t)$ is in integral separating structure of f, then f^d is a threshold function with separating structure*

$$\left(w_1,\dots,w_n,\left(\sum_{i=1}^n w_i\right) - t - 1\right)$$

Furthermore,

(i) *If $t \le \frac{1}{2}\sum_{i=1}^n w_i - 1$, then f is dual-major;*

(ii) *If $t \ge \frac{1}{2}\sum_{i=1}^n w_i - 1$, then f is dual-minor;*

Proof. Let $t' = \sum_{i=1}^n w_i - t - 1$. Since the threshold t and weights $w_1,\dots,w_n$ are integra, the following equivalences hold for all $X \in \mathcal{B}^n$:

$$\begin{aligned} f^d(X) = 0 \qquad &\iff \qquad f(\overline{X}) = 1 \\ &\iff \qquad \sum_{i=1}^n w_i(1-x_i) > t \\ &\iff \qquad \sum_{i=1}^n w_i x_i \le t' \end{aligned}$$

So f^d is threshold with separating structure $(w_1,\dots,w_n,(\sum_{i=1}^n w_i) - t - 1)$.

The last two parts follow from the observation that $f^d \le f$ if $t \le t'$ and $f \le f^d$ if $t' \le t$. ■

Example. The function $f(x,y,z,w) = xy \vee xz \vee xw \vee yxw$ admits the separating structure (4,2,2,2,5), so the dual f^d is threshold with separating structure $(4,2,2,2,4+2+2+2-5-1) = (4,2,2,2,4)$. Since the new threshold is smaller, f^d is dual-minor.

However, another separating structure for f is (2,1,1,1,2), which yields the dual separating structure $(2,1,1,1,2+1+1+1-2-1) = (2,1,1,1,2)$, so f is in fact self-dual. △

Theorem 7.6. *A function $f(x_1,\dots,x_n)$ is a threshold function if and only if its self-dual extension $f^{\mathrm{SD}}(x_1,\dots,x_{n+1}) = f\overline{x}_{n+1} \vee f^d x_{n+1}$ is a threshold function.*

Proof. Suppose that f is a threshold function with integral separating structure $(w_1,\dots,w_n,t)$. Then, by the previous theorem,

$$\left(w_1,\dots,w_n,2t+1-\sum_{i=1}^n w_i,t\right)$$

is a separating structure for f^{SD}. Conversely, if f^{SD} is a threshold function with separating structure $(w_1,\dots,w_{n+1},t)$, then $(x_1,\dots,x_n,t)$ is a separating structure for f. ■

7.2 Characterisation of Threshold Functions

The first characterisation is a simple linear programming formulation which provides a useful computational tool for the recognition of threshold functions. For the sake of simplicity, we only state it for positive functions: since every threshold function is monotone, this restriction does not entail any essential loss of generality.

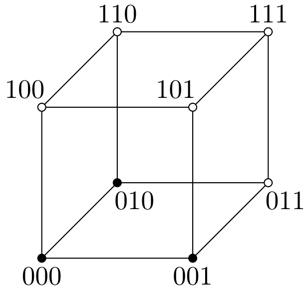
Theorem 7.7. *A positive Boolean function f with maximal false point $X^1,X^2,\dots,X^p$ and minimal true points $Y^1,Y^2,\dots,Y^m$ is a threshold function if and only if the system of inequalities*

$$\sum_{i=1}^n w_i x_i^j \le t \qquad\qquad j = 1,\dots,p$$

$$\sum_{i=1}^{n} w_i y_i^j \geq t+1 \qquad j = 1,\ldots,m$$
$$w_i \geq 0 \qquad j = 1,\ldots,m$$

has a solution $(w_1,\ldots,w_n,t)$. *When this is the case, every solution of the system is a separating structure for* f.

Example. Let $f(x,y,z) = x \vee yz$.



The maximal false points are 010 and 001, and the minimal true points are 100 and 011, so the system of inequalities is given by

$$\begin{aligned} w_2 &\leq t && \text{(010 false)} \\ w_3 &\leq t && \text{(001 false)} \\ w_1 &\leq t+1 && \text{(100 true)} \\ w_2 + w_3 &\geq t+1 && \text{(011 true)} \\ w_i &\geq 0 \end{aligned}$$

This system has solution $(w_1,w_2,w_3,t) = (2,1,1,1)$, so f is a threshold function with separating structure $(2,1,1,1)$. △

Let $k \geq 2$ be a natural number. A Boolean function f on $\mathcal{B}$ is *k-summable* if for some $r \in \{2,3,\ldots,k\}$, there exist r-many not-necessarily distinct false points of f, say $X^1,\ldots,X^r$ and r-many not-necessarily distinct true points $Y^1,Y^2,\ldots,Y^r$ such that

$$\sum_{i=1}^{r} X^i = \sum_{i=1}^{r} Y^i$$

A function is *k-asummable* if it is not k-summable, and is *asummable* if it is k-asummable for all $k \geq 2$.

Example. The function $f(x_1,x_2) = x_1x_2 \vee \overline{x}_1\overline{x}_2$ has true points 00 and 11, and false points 01 and 10. Then, f is 2-summable since

$$00 + 11 = 01 + 10$$

△

Theorem 7.8. *A Boolean function is a threshold function if and only if it is asummable.*

Proof. ■

7.3 Threshold Functions and Chow Parameters

Recall that the Chow parameters of a Boolean function f on $\mathcal{B}^n$ are the $n+1$ integers $(\omega_1,\ldots,\omega_n,\omega)$ where $\omega = \omega(f)$ is the number of true points of f and ω_i is the number of true points $X^* = (x_1^*,\ldots,x_n^*)$ of f with $x_i^* = 1$.

Note that

$$(\omega_1,\ldots,\omega_n) = \sum_{j=1}^{\omega} Y^j$$

where $Y^1,\ldots,Y^\omega$ are the true points of f.

A Boolean function f is a *Chow function* if no other function has the same Chow parameters as f.

Example. The function $f(x_1,x_2) = x_1x_2 \vee \overline{x}_1\overline{x}_2$ is not a Chow function since it has the same Chow parameters as $g(x_1,x_2) = x_1\overline{x}_2 \vee \overline{x}_1 x_2$, namely (1,1,2). △

Theorem 7.9. *Every threshold function is a Chow function.*

Proof. Let f be a threshold function on $\mathcal{B}^n$, and let g be a function on $\mathcal{B}^n$ with the same Chow parameters. Let $Y^1,\ldots,Y^\omega$ be the true points of f, and $X^1,\ldots,X^k,Y^{k+1},\ldots,Y^\omega$ be the true points of g, where the X^i are false points of f.

Since f and g have the same Chow parameters,

$$\sum_{j=1}^{\omega} Y^j = \sum_{j=1}^{k} X^j + \sum_{j=k+1}^{\omega} Y^j$$

or equivalently,

$$\sum_{j=1}^{k} Y^j = \sum_{j=1}^{k} X^j$$

Now, if $k \geq 1$, this contradicts the asummability of f. So $k = 0$, and f and g have the same set of true points, i.e. $f = g$. ■

Note that all the points occuring in this final sum are distinct. This motivates the following definition.

A Boolean function f is *weakly asummable* if for all $k \geq 1$, thexe do not exist k-many *distinct* false points $X^1,\ldots,X^k$ and k-many *distinct* true points $Y^1,\ldots,Y^k$ such that

$$\sum_{i=1}^{r} X^i = \sum_{i=1}^{r} Y^i$$

Clearly, every asummable (and hence threshold) function is weakly asummable. Moreover, the previous proof actually establishes that every weakly asummable function is a Chow function. In fact, the converse implication folds as well.

Theorem 7.10. *A Boolean function is weakly asummable if and only if it is a Chow function.*

Proof. The forward implication is shown above. For the reverse implication, let $X^1,\ldots,X^q$ denote the false points of a function f, and let $Y^1,\ldots,Y^p$ denote its true points.

If f is not weakly assumable, then without loss of generality by reordering the points, we have

$$\sum_{i=1}^{k} X^i = \sum_{j=1}^{k} Y^j$$

for some $k \geq 1$. Let g be the Boolean function whose true points are precisely $X^1,\ldots X^k,Y^{k+1},\ldots,Y^p$. Then, f and g have distinct true points and are hence distinct functions, but f and g share the same Chow parameters, and hence f is not a Chow function. ■

It is natural to expect some sort of relationship between the Chow parameters of a threshold function and the separating structure defining the function, since both types of coefficients provide a "measure" of the "influence" of each variable on the function. This relationship is most natural expressed in terms of the so-called *modified Chow parameters* of the function.

The *modified Chow parameters* of a Boolean function $f(x_1,\dots,x_n)$ are the $n+1$ numbers $(\pi_1,\dots,\pi_n,\pi)$ defined as $\pi = \omega - 2^{n-1}$ and $\pi_k = 2\omega_k - \omega$, where $(\omega_1,\dots,\omega_n,\omega)$ are the Chow parameters of f.

Since there is a bijection between Chow parameters and modified Chow parameters, every threshold function is uniquely determined by its modified Chow parameters, or by Chow parameters, or by any of its separating structures.

Theorem 7.11. *If f is a Boolean function with modified Chow parameters $(\pi_1,\dots,\pi_n,\pi)$, then for all $i \in [n]$,*

(i) If f is positive in x_i and f depends on x_i, then $\pi_i > 0$;

(ii) If f is negative in x_i and f depends on x_i, then $\pi_i < 0$;

(iii) If f does not depend on x_i, then $\pi_i = 0$;

(iv) The modified Chow parameters of f^d are $(\pi_1,\dots,\pi_n,-\pi)$;

(v) If f is dual-major ($f^d \le f$) then $\pi \ge 0$;

(vi) If f is dual-minor ($f \le f^d$) then $\pi \le 0$;

Proof. ■

Theorem 7.12. *If $f(x_1,\dots,x_n)$ is a threshold Boolea function give by the integral separating structure $(\omega_1,\dots,\omega_n,t)$, then the number of true points of f can be computed in $O(nt)$ arithmetic operations.*

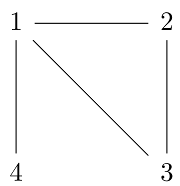
Proof. ■

7.4 Threshold Graphs

In this section, we specialise some of these previous results to the case of graphic (i.e. purely quadratic and positive) functions. Recall that such a function $f(x_1,\dots,x_n) = \bigvee_{ij\in E} x_i x_j$ can be identified with an undirected graph $G_f = \big([n],E\big)$. Conversely, if $G = (V,E)$ is an arbitrary undirected graph, we define its corresponding stability function f_G by the expression $\bigvee_{ij\in E} x_i x_j$.

A graph G is a *threshold graph* if its stability function f_G is threshold, and we say that $(\omega_1,\dots,\omega_n,t)$ is a *separating structure* of G if it is a separating structure of f_G.

Example. The function $f(x_1,x_2,x_3,x_4) = x_1x_2 \vee x_1x_3 \vee x_1x_4 \vee x_2x_3$ is graphic with associated graph:



This function is threshold with separating structure, say (3,2,2,1,3). So G_f is a threshold graph with separating structure (3,2,2,1,3). △

Is there an easy way to determine which graphic functions are threshold, or equivalently, which graphs are threshold?

Recall that an independent set in a graph is a set of vertices of which no two are adjacent.

Theorem 7.13. *A graph $G = (V,E)$ is a threshold graph if and only if there exists a structure $(\omega_1, \dots, \omega_n, t)$ such that for every subset S of vertices, S is an independent set if and only if*

$$\sum_{i \in S} w_i \leq t$$

We also recall that for a graph $G = (V,E)$ and a vertex $i \in V$, the *neighbourhood* $N(i)$ of i is the set of vertices adjacent to i. We say that i is *isolated* if $N(i) = \varnothing$, and that i is dominating if $N(i) = V \setminus \{i\}$. Note that isolated vertices of G correspond to inessential variables of f_G, since they don't appear in any term.

Theorem 7.14. *A graph G is threshold if and only if it is C_4, P_4, and $2K_2$-free.*

Proof. ■

Example. The graph above is threshold since it does not contain C_4, P_4, nor $2K_2$ as induced subgraphs. △

Theorem 7.15. *A graphic function $f(x_1, \dots, x_n)$ is threshold if and only if there is a permutation $\sigma : [n] \to [n]$ such that for every $i \in [n]$, σ_i is either isolated or dominating in the subgraph of G_f induced by $\{i, \dots, n\}$.*

8 Read-Once Functions

A Boolean function f is *read-once* if it can be represented by a Boolean expression over $\{\overline{x}, \wedge, \vee\}$ such that every variable appears exactly once. Such an expression is called a *read-once expression* for f.

Example. The function $f_0(a,b,c,w,x,y,z) = ay \vee cxy \vee bw \vee bz$ is a read-once function since it can be factored into the expression $f_0 = y(a \vee cx) \vee b(w \vee z)$, where every variable appears exactly once. △

Note that read-once functions are necessarily monotone since every variable appears either in its positive or negative form in the read-once expression, an thus cannot contribute in conflicting directions. However, we will make a stronger assumption that a read-once function is positive, simply by renaming any negative variables $\overline{x}_i$ as new positive variables x_i'.

Consider the two simple functions $f_1 = ab \vee bc \vee cd$ and $f_2 = ab \vee bc \vee ac$. Neither of these functions are read-once. As we will see, these illustrate the two types of forbidden functions that characterise read-once functions.

Let f be a positive Boolean function over the variables $x_1, \dots, x_n$. The *co-occurence graph* $G(f)$ of f is the undirected graph with vertex set $V = \{x_1, \dots, x_n\}$ and edge set defined by $(x_i, x_j) \in E$ if and only if x_i and x_y occur together at least once in some prime implicant of f.

Example. The co-occurence graphs of f_1 and f_2 are:

△

A Boolean function is *normal* if every clique of its co-occurence graph is contained in the set of variables of a prime implicant of f.

Example. f_2 is not normal, since $\{a,b,c\}$ is a clique, and the prime implicants of are ab, bc, and ac, which all contain only 2 variables. △

Theorem 8.1. *A positive Boolean function f is read-once if and only if it is normal and its co-occurence graph $G(f)$ is P_4-free.*

Before proving this theorem, we review a few properties of the dual of a Boolean function and prove an important result on positive Boolean functions.

8.1 Dual Implicants

Recall that the dual f^d of a Boolean function f is the function defined by

$$f^d(X) = \overline{f(\overline{X})}$$

An expression for f^d can be obtained from any expression for f by interchanging the operators $\wedge$ and $\vee$ as well as the constants 0 and 1. In particular, given a DNF expression for f, this exchange yields a CNF expression for f^d. This shows that the dual of a read-once function is also read-once.

Let $\mathcal{P}$ be the set of prime implicants of a Boolean function f over the variables $x_1,\dots,x_n$, and let $\mathcal{D}$ be the collection of prime implicants for the dual function f^d. We assume throughout that all of the variables for f (and hence for f^d) are essential.

We use the term "dual (prime) implicant" of f to mean a (prime) implicant of f^d. For positive functions, the prime implicants of f correspond precisely to the set of minimal true points $\min T(f)$, and the dual prime implicants of f correspond precisely to the set of maximal false points $\max F(f)$.

We have also seen that the implicants and dual implicants of a Boolean function f, viewed as sets of literals, have pairwise non-empty intersections. In particular, this holds for the prime implicants and the dual prime implicants, and moreover, the prime implicants and the dual prime implicants are minimal with this property. That is, for every proper subsets S of a dual prime implicant of f, there is a prime implicant P such that $P \cap S = \varnothing$.

In terms of hypergraph theory, the prime implicants $\mathcal{P}$ form a clutter (i.e. a collection of sets, or hyperedges, such that no set contains another set), as does the collection of dual prime implicants $\mathcal{D}$.

Theorem 8.2. *Let f and g be positive Boolean functions over the varibles $x_1,\dots,x_n$, and let $\mathcal{P}$ and $\mathcal{D}$ be the collections of prime implicants of f and g, respectively. Then, the following are equivalent:*

(i) $g = f^d$;

(ii) *For every partition of $\{x_1,\dots,x_n\}$ into two disjoint sets A and $\overline{A}$, there is either a member of $\mathcal{P}$ contained in A, or a member of $\mathcal{D}$ containied in $\overline{A}$, but not both;*

(iii) *$\mathcal{D}$ is precisely the family of minimal transversals of $\mathcal{P}$;*

(iv) *$\mathcal{P}$ is precisely the family of minimal transversals of $\mathcal{D}$;*

(v) *For all $P \in \mathcal{P}$ and $D \in \mathcal{D}$, we have $P \cap D \neq \varnothing$, and for every set $B \subset \{x_1,\dots,x_n\}$ of variables, there exists $D \in \mathcal{D}$ such that $D \subseteq B$ if and only if $P \cap B \neq \varnothing$ fot every $P \in \mathcal{P}$.*

Theorem 8.3. *A set of variables B is a dual implicant of the function f if and only if $P \cap B \neq \varnothing$ for all prime implicants P of f.*

A subset T of the variables is called a *dual sub-implicant* of f if T is a subset of a dual prime implicant of f. That is, there exists a prime implicant D of f^d such that $T \subseteq D$. A *proper* dual sub-implicant is a non-empty proper subset of a dual prime implicant.

Example. Let $f = x_1x_2 \vee x_2x_3x_4 \vee x_4x_5$. Its dual is $f^d = x_1x_3x_5 \vee x_1x_4 \vee x_2x_4 \vee x_2x_5$. The proper dual sub-implicants of f are the pairs $\{x_1,x_3\}$, $\{x_3,x_5\}$, $\{x_1,x_5\}$, and the singletons of each variable. △

Note that if T is a proper dual sub-implicant o f, then there exists a prime implicant $P \in \mathcal{P}$ such that $T \cap P = \varnothing$.

9 Characterising Read-Once Functions

A positive Boolean expression over the operation of conjunction and disjunction may be represented as a rooted parse tree whose leaves are labeled by the variables $\{x_1, \ldots, x_n\}$, and whose internal nodes are labeled by the Boolean operations $\vee$ and $\wedge$. The parse tree represents the computation of the associated Boolean function according to the given expression, and each internal node is the root of a subtree corresponding to a part of the expression. If the expression is read-once, then each variable appears on exactly one leaf of the tree, and there is a unique path from the root to the variable.

Lemma 9.1. *Let T be a parse tree for a read-once expression for a positive Boolean function f over the variables $x_1, \ldots, x_n$. Then (x_i, x_j) is an edge in the co-occurence graph $G(f)$ if and only if the lowest common ancestor of x_i and x_j in the tree T is labeled by a conjunction $\wedge$.*

Proof. ■

Theorem 9.2. *Let f be a positive Boolean function over the variables $x_1, \ldots, x_n$. Then, the following are equivalent:*

(i) *f is a read-once function;*

(ii) *The co-occurence graphs $G(f)$ and $G(f^d)$ are complementary, i.e. $\overline{G(f)} = G(f^d)$;*

(iii) *The co-occurence graphs $G(f)$ and $G(f^d)$ have no edges in common, i.e. $E\big(G(f)\big) \cap E\big(G(f^d)\big) = \varnothing$;*

(iv) *For all $p \in \mathcal{P}$ and $D \in \mathcal{D}$, $|P \cap D| = 1$;*

(v) *f is normal and the co-occurence graph $G(f)$ is P_4-free.*

Proof. ■

10 Linear Read-Once Functions

10.1 Specifying Sets and Specification Number

10.2 Essential Points

10.3 The Number of Essential Points and the Number of Extremal Points

10.4 Positive Functions and the Number of Extremal Points

10.4.1 A Property of Extremal Points

10.4.2 Canalysing Functions

10.4.3 Non-Canalysing Functions with Canalysing Restrictions

10.4.4 Non-Canalysing Functions Containing Non-Canalysing Restrictions

10.5 Chow and Read-Once Functions

10.6 Threshold Functions and Specification Number

10.6.1 Minimal Non-LRO Functions

10.6.2 Non-LRO Threshold Functions with Minimum Specification Number

11 Partially-Defined Boolean Functions and Logical Analysis of Data

11.1 Extensions of PDBFs

11.2 Patterns and Theories of PDBFs

11.3 Roles of Theories and Co-Theories

11.4 Decision Trees and PDBFs

12 Pseudo-Boolean Functions

12.1 Pseudo-Boolean Optimisation

12.2 Posiform Transformations and Conflict Graphs

12.2.1 The Struction