# Modular Decomposition of Boolean Functions

Jan C. Bioch



| ERIM Report Series *Research in Management* | |
|---|---|
| ERIM Report Series reference number | ERS-2002-37-LIS |
| Publication | April 2002 |
| Number of pages | 36 |
| Email address corresponding author | bioch@few.eur.nl |
| Address | Erasmus Research Institute of Management (ERIM)<br>Rotterdam School of Management / Faculteit Bedrijfskunde<br>Erasmus Universiteit Rotterdam<br>P.O. Box 1738<br>3000 DR Rotterdam, The Netherlands<br>Phone: +31 10 408 1182<br>Fax: +31 10 408 9640<br>Email: info@erim.eur.nl<br>Internet: www.erim.eur.nl |

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:
www.erim.eur.nl

**E**RASMUS **R**ESEARCH **I**NSTITUTE OF **M**ANAGEMENT

# REPORT SERIES
*RESEARCH IN MANAGEMENT*

| BIBLIOGRAPHIC DATA AND CLASSIFICATIONS | | |
|---|---|---|
| Abstract | Modular decomposition is a thoroughly investigated topic in many areas such as switching theory, reliability theory, game theory and graph theory. Most applications can be formulated in the framework of Boolean functions. In this paper we give a uni_ed treatment of modular decomposition of Boolean functions based on the idea of generalized Shannon decomposition. Furthermore, we discuss some new results on the complexity of modular decomposition. We propose an O(mn)-algorithm for the recognition of a modular set of a monotone Boolean function f with m prime implicants and n variables. Using this result we show that the the computation of the modular closure of a set can be done in time O(mn2). On the other hand, we prove that the recognition problem for general Boolean functions is coNP-complete. | |
| Library of Congress Classification (LCC) | 5001-6182 | Business |
| | 5201-5982 | Business Science |
| | HB 143 | Mathematical Programming |
| Journal of Economic Literature (JEL) | M | Business Administration and Business Economics |
| | M 11<br>R 4 | Production Management<br>Transportation Systems |
| | C 69 | Mathematical Methods and Programming |
| European Business Schools Library Group (EBSLG) | 85 A | Business General |
| | 260 K<br>240 B | Logistics<br>Information Systems Management |
| | 250 A | Mathematics |
| Gemeenschappelijke Onderwerpsontsluiting (GOO) | | |
| Classification GOO | 85.00 | Bedrijfskunde, Organisatiekunde: algemeen |
| | 85.34<br>85.20 | Logistiek management<br>Bestuurlijke informatie, informatieverzorging |
| | 31.10 | Logica, verzamelingenleer |
| Keywords GOO | Bedrijfskunde / Bedrijfseconomie | |
| | Bedrijfsprocessen, logistiek, management informatiesystemen | |
| | Boolean-functions, wiskundige ontleding, modulaire functies, modulaire matrices, speltheorie, betrouwbaarheid | |
| Free keywords | Boolean functions, committees, computational complexity, decomposition algorithm, modular decomposition, modular sets, substitution decomposition, game theory, reliability theory, switching theory. | |

# Modular Decomposition of Boolean Functions

Jan C. Bioch
Dept. of Computer Science, Erasmus University Rotterdam,
P.O. Box 1738, 3000 DR Rotterdam.
email: bioch@few.eur.nl

**Abstract**

Modular decomposition is a thoroughly investigated topic in many areas such as switching theory, reliability theory, game theory and graph theory. Most applications can be formulated in the framework of Boolean functions. In this paper we give a unified treatment of modular decomposition of Boolean functions based on the idea of generalized Shannon decomposition. Furthermore, we discuss some new results on the complexity of modular decomposition. We propose an $O(mn)$-algorithm for the recognition of a modular set of a monotone Boolean function $f$ with $m$ prime implicants and $n$ variables. Using this result we show that the the computation of the modular closure of a set can be done in time $O(mn^2)$. On the other hand, we prove that the recognition problem for general Boolean functions is coNP-complete.

Keywords: Boolean functions, committees, computational complexity, decomposition algorithm, modular decomposition, modular sets, substitution decomposition, game theory, reliability theory, switching theory.

## 1 Basic concepts and applications

### 1.1 Disjunctive decompositions

Let $f : \{0,1\}^n \mapsto \{0,1\}$ be a Boolean function and $V = \{1, 2, \cdots, n\}$. Identify each $i \in V$ with the variable $x_i$. Then $f$ is said to be a function defined on $V$. Furthermore, if $V = A_1 \cup A_2 \cup \cdots A_m$ is a partition of $V$ ($A_i \cap A_j = \emptyset,\ i \neq j$), then we will denote this by $x_V = (x_{A_1}, \cdots, x_{A_m})$ and $f(x_V) = f(x_{A_1}, \cdots, x_{A_m})$. Let $F(y_I)$ and $g_i(x_{A_i})$ be Boolean functions defined on the mutually disjoint sets $I = \{1, \cdots, m\}$ and $A_i,\ i \in I$, and let $V = \cup_{i=1}^{m} A_i$. Then the Boolean function defined by

$$f(x_V) = F(g_1(x_{A_1}), \cdots, g_m(x_{A_m})),$$

is called the *composition* of the functions $F$ and $g_i,\ i \in I$, obtained by *substitution* of the variables $y_i$ in $F$ by the functions $g_i,\ i \in I$. This composition is denoted by $F[g_i,\ i \in I]$. A composition is called *proper* if $|I| > 1$ and $|A_i| > 1$ for some $i \in I$. A Boolean function is said to be *decomposable* if it has a representation as a proper composition. Otherwise, the function $f$ is called *indecomposable* or *prime*. If $F[g_i,\ i \in I]$ is a decomposition of the function $f$ then the partition $\pi = \{A_i,\ i \in I\}$ is called a *congruence partition* and $F$ is called the *quotient* of $f$ modulo $\pi$. This is denoted by $F = f/\pi$. From the definition of decomposition it follows easily that

$$f = F[g_i,\ i \in I] \Leftrightarrow f^d = F^d[{g_i}^d,\ i \in I]. \tag{1}$$

Therefore, we have $F = f/\pi \Leftrightarrow F^d = f^d/\pi$. Moreover, we will show that the functions $g_i,\ i \in I$, are determined modulo complementation of the functions, and that the quotient $F$ is determined modulo complementation of the variables (see corollary 1.1). It appears (cf [18]) that each decomposition of a Boolean function $f$ can be obtained by a series of so called *simple disjunctive decompositions.* These are decompositions of the form

$$f(x_V) = F(x_A, g(x_B)),$$

where $\pi = \{A, B\}$ is a partition of $V$. It is known ([18, 15]) that the study of non-disjunctive decompositions can also be reduced to that of simple disjunctive decompositions. However, a discussion of non-disjunctive decompositions is outside the framework of this chapter.

**Definition 1** *Let $f$ be a Boolean function defined on $V$. Then $A \subseteq V$ is called a modular set of $f$ if $f$ has a simple disjunctive decomposition of the form $f(x_V) = F(g(x_A), x_B)$. The function $g$ is called a component of $f$.*

From this definition it follows that the base set $V$ and its singleton subsets are modular. Before we proceed we introduce some notations used in this chapter.

### Notations

A Boolean function $f$ is called *trivial* if it is constant: $f \equiv 0$ (denoted by $f = \bot$) or $f \equiv 1$ (denoted by $\top$). Otherwise $f$ is called *non-trivial.* We say that a set $A \subseteq V$ is *essential* for $f$ or that $f$ *depends on* $A$, if $A$ contains at least one essential variable of $f$. The set of all modular sets of $f$ is denoted by $\mu_0(f)$ and the set of all modular sets of $f$ that are essential for $f$ is denoted by $\mu(f)$ . A congruence partition $\pi$ is called *essential* if all the classes of $\pi$ are essential for $f$. Furthermore, the set of all essential congruence partitions of $f$ is denoted by $\Gamma(f)$.

The following theorem shows that the modular sets of a function $f$ are precisely the classes of the congruence partitions of $f$.

**Theorem 1** *Let $f$ be a Boolean function and let $\pi = \{A_i, i \in I\}$ be a partition of $f$. Then $\pi \in \Gamma(f) \Leftrightarrow A_i \in \mu(f)$ for all $i \in I$.*

**Proof.** If $f(x_V) = F[g_i(x_{A_i}), i \in I]$, then obviously for all $i \in I$ there exists a Boolean function $F_i$ such that $f(x_V) = F_i(g_i(x_{A_i}), x_{\bar{A}_i})$, implying that $A_i \in \mu(f)$. To prove the converse we first assume that $|I| = 2$. If $A, B \in \mu(f)$, then $f(x_A, x_B) = F(g(x_A), x_B) = G(h(x_B), x_A)$. Since $A$ and $B$ are essential for $f$, there exist binary vectors $a(y)$ and $b(y)$ such that $y = g(a(y)) = h(b(y))$, where $y \in \{0, 1\}$. Now we define the function $H$ by:

$$H(y_1, y_2) = H(g(a(y_1)), h(b(y_2)) := f(a(y_1), b(y_2)).$$

To prove that $f(x_A, x_B) = H(g(x_A), h(x_B))$, we note that $g(a_1) = g(a_2)$ implies $f(a_1, x_B) = F(g(a_1), x_B) = F(g(a_2), x_B) = f(a_2, x_B)$. Furthermore, $h(b_1) = h(b_2)$ implies $f(x_A, b_1) = f(x_A, b_2)$. Therefore, we conclude $f(a_1, b_1) = f(a_2, b_1) = f(a_2, b_2)$. Conclusion: if $A, B \in \mu(f)$, then $\pi = \{A, B\} \in \Gamma(f)$. The case $|I| \geq 2$ is a straightforward generalization. ■

## 1.2 Applications

Substitution decomposition has been studied thoroughly by researchers in many different contexts such as switching theory, game theory, reliability theory, network theory, graph theory and hypergraph theory. Not surprisingly, the concept of a modular set is rediscovered several times under various names: bound sets, autonomous sets, closed sets, stable sets, clumps, committees, externally related sets, intervals, nonsimplifiable subnetworks, partitive sets and modules, see [12, 21] and references therein. An excellent survey for the various applications of substitution decomposition and connections with combinatorial optimization is given by Möhring and Radermacher in [21, 22]. The decomposition of monotone Boolean functions has been studied in several contexts: game theory (decomposition of $n$-person games [28]), reliability theory (decomposition of coherent systems [7]) and set systems (clutters [6]).

In switching theory decomposition of general Boolean functions is still an important tool in the design and analysis of circuits. Some applications of decompositions of positive Boolean functions to be discussed briefly here are in the areas of reliability theory, game theory and combinatorial optimization.

**Application 1 (Reliability theory)** *In reliability theory a system $S$ consisting of $n$ components is modeled by a positive (monotone) Boolean function $f_S$ called the structure function of $f_S$. This function indicates whether system $S$ is operating or not depending on the states of the $n$ components: operative ($x_i = 1$) or failed ($x_i = 0$). Modular sets play a role in the design and analysis of a complex system $S$ because they reflect the decomposability possibilities of $S$ in subsystems.*

**Application 2 (Game theory)** *The concept of an $n$-person simple game (or voting game) $G$ can be modeled by a positive Boolean function $f_G$ such that the winning coalitions of $G$ correspond to the prime implicants of $f_G$. Factorisation of compound simple games studied by Shapley [28] is equivalent to decomposition of the associated positive Boolean function.*

**Application 3 (Clutters)** *Combinatorial optimization over set systems has initiated the research on decomposition of clutters (Sperner families) (see e.g. [21, 6]). The interface between a clutter $C$ and its associated positive function $f_C$ is given by the correspondence between the elements of $C$ and the prime implicants of $f_C$.*

## 2 Generalized Shannon decomposition

Let $f$ be a Boolean function on V. Then for all $j \in V$ the following decomposition holds:

$$f = \bar{x}_j f_{x_j=0} \vee x_j f_{x_j=1}. \tag{2}$$

Equation (2) is known as a *Shannon decomposition* of $f$. Now consider the simple disjunctive decomposition

$$f(x_V) = F(g(x_A), x_B). \tag{3}$$

Then by applying equation (2) to $F$ we get:

$$f(x_V) = \bar{g}(x_A)F_0(x_B) \vee g(x_A)F_1(x_B), \tag{4}$$

where $F_0(x_B) = F(x_B, 0)$ and $F_1(x_B) = F(x_B, 1)$.
Conversely, let $g$ and $h_0, h_1$ be arbitrary Boolean functions defined respectively on $A$ and

$B$ such that $f = \bar{g}h_0 \vee gh_1$, and let the function $F$ be defined by $F(y, x_B) := \bar{y}h_0 \vee yh_1$. Then $f(x_V) = F(g(x_A), x_B)$ is a simple disjunctive decomposition of $f$, where $F_0(x_B) = h_0$ and $F_1(x_B) = h_1$. Therefore, we have proved the following fundamental lemma:

**Lemma 1** *Let $f$ be a Boolean function on $V$. Then $A \subseteq V$ is a modular set of $f$ iff there exists a Boolean function $g$ on $V$ and functions $h_0$ and $h_1$ on $B = V \setminus A$ such that $f = \bar{g}h_0 \vee gh_1$.*

We call the decomposition in the previous lemma *a generalized Shannon decomposition*. In particular, we call the decomposition in equation (4) a *generalized Shannon representation* of the simple disjunctive decomposition (3). If $A$ is a modular set of the function $f$ such that $A$ contains at least one essential variable of $f$, then it follows from the decomposition

$$f = \bar{g}h_0 \vee gh_1, \tag{5}$$

that the function $g$ is non-trivial and that the functions $h_0$ and $h_1$ are not identical. Therefore, there exists a binary vector $b_0$ such that either $g(x_A) = f(x_A, b_0)$ or $\bar{g} = f(x_A, b_0)$. Since $\bar{g}$ is a component of $f$ iff $g$ is a component of $f$ we may assume that the function $g$ is a subfunction of $f$.

**Definition 2** *Let $A$ be a modular set of $f$. Then a non-trivial subfunction $f(x_A, b_0)$ is denoted by $f_A(x_A)$. For general Boolean functions this subfunction is determined modulo complementation. For monotone Boolean functions the function $f_A(x_A)$ is uniquely determined and called the contraction of $f$ with respect to to $A$.*

In general, equation (5) shows that if $b$ is a fixed vector then the function $f(x_A, b)$ is either trivial or identical to $g$ of identical to $\bar{g}$. It is not difficult to see that the converse holds also. Therefore, the following theorem holds:

**Theorem 2** *Let $f$ be a Boolean function defined on $V$. If $A \subseteq V$ contains at least one essential variable of $f$, then the following statements are equivalent:*

*a) $A$ is modular*

*b) There exists a vector $b_0$ such that the function $g(x_A) := f(x_A, b_0)$ is non-trivial and for all fixed $b$ the function $f_b := f(x_A, b)$ is either trivial or identical to either $g$ or $\bar{g}$.*

**Corollary 1** *Suppose $f(x_V) = F(g(x_A), x_B) = G(h(x_A), x_B)$, and that $A$ is essential for $f$. Then either $g = h$ and $F = G$ or $g = \bar{h}$ and $F(y, x_B) = G(\bar{y}, x_B)$.*

**Proof.** We leave this as an exercise. ∎

## 3 Properties of modular sets

In this section we derive a number of properties of modular sets by proving decomposition theorems such as in ([15]). The main tool we use here is the Shannon representation of a simple decomposition and theorem (2).

**Lemma 2** *Let $f(x, y)$ be a Boolean function depending on $x$ and $y$.*
*Then $f(x, y) = y_1 \star y_2$, where $y_1 = x$ or $\bar{x}$, $y_2 = y$ or $\bar{y}$ and $\star$ denotes $\vee, \wedge$, or $\oplus$.*

**Proof.** Consider the decomposition $f(x, y) = \bar{y}f(x, 0) \vee yf(x, 1)$. Since $x$ is a component of $f$ we have according to theorem (2) to consider the following cases: $x = f(x, 0), x = f(x, 1), \bar{x} = f(x, 0)$ or $\bar{x} = f(x, 1)$. If $x = f(x, 0)$ then $f(x, 1) \in \{0, 1, \bar{x}\}$, implying that $f(x, y) \in \{x\bar{y}, x \vee y, x \oplus y\}$. If $x = f(x, 1)$ then $f(x, y) \in \{xy, x \vee \bar{y}, x \oplus \bar{y}\}$. Both cases together can be expressed as $f(x, y) \in \{x \star y, x \star \bar{y}\}$. Similarly, the other two cases yield $f(x, y) \in \{\bar{x} \star y, \bar{x} \star \bar{y}\}$. ■

**Corollary 2** *There are ten Boolean functions functions of two essential variables.*

**Theorem 3** *Suppose $A \in \mu(f)$ and $\bar{A}$ is essential for $f$. Then $\bar{A} \in \mu(f) \Leftrightarrow f$ has a decomposition $f(x_V) = g(x_A) \star h(x_{\bar{A}})$, where $\star$ denotes $\vee, \wedge$, or $\oplus$.*

**Proof.** Suppose $B = \bar{A} \in \mu(f)$. Then by theorem (1) $f$ can be written as $f(x_A, x_B) = F(g_1(x_A), h_1(x_B))$. Since $A$ and $B$ are essential for $f$, the function $F$ has two essential variables. So, by lemma (2) it follows that $f(x_V) = g(x_A) \star h(x_{\bar{A}})$, where $g$ and $h$ are respectively equal to $g_1$ and $h_1$ modulo complementation. The converse is obvious. ■

**Theorem 4** *Let $A \in \mu(f)$ and let $g$ be a component of $f$ defined on $A$. Then $\mu_0(g) = \{C \subseteq A \mid C \in \mu_0(f)\}$. In addition, if $f$ depends on all the variables in $A$, then $\mu(g) = \{C \subseteq A \mid C \in \mu(f)\}$.*

**Proof.** Wlog we may assume that $g = f_A$. So, $f(x_A, x_{\bar{A}}) = F(g(x_A), x_{\bar{A}})$ and there exists a vector $b$ such that $g(x_A) = f(x_A, b)$. If $C \subseteq A$ and $C \in \mu_0(f)$, then we also have $f(x_C, x_{\bar{C}}) = G(h(x_C), x_{\bar{C}})$. Therefore, $g(x_A) = G(h(x_C), x_{A\setminus C}, b)$. Let the function $H$ be defined by $H(y, y_{A\setminus C}) := G(y, x_{A\setminus C}, b)$. Then $g(x_C, x_{A\setminus C}) = H(h(x_C), x_{A\setminus C})$, so we have $C \in \mu_0(g)$. If in addition $f$ depends on all variables in $A$ then $C \in \mu(g)$.

Conversely, suppose $C \in \mu_0(g)$. Then $g(x_A) = G(h(x_C), x_{A\setminus C})$. Therefore, $f(x_A, x_{\bar{A}}) = F(G(h(x_C), x_{A\setminus C}), x_{\bar{A}})$. Let the function $H$ be defined by $H(y_C, y_{\bar{C}}) := F(G(y, y_{A\setminus C}), y_{\bar{A}})$.

Then $f(x_C, x_{\bar{C}}) = H(h(x_C), x_{\bar{C}})$, so $C \in \mu_0(f)$. If in addition $f$ depends on all variables in $A$, then we have $C \in \mu(f)$. ∎

**Proposition 1** *Let $\pi = \{A_i, i \in I\} \in \Gamma(f)$ and let $F = f/\pi$. Suppose $\emptyset \neq J \subset I$. If $B = \bigcup\{A_j, j \in J\}$, then $J \in \mu(F) \Leftrightarrow B \in \mu(f)$.*

**Proof.** Suppose $f(x_V) = F[g_i(x_{A_i}), i \in I]$ and $J \in \mu(F)$. Wlog we may assume $J = \{1, 2, \cdots, l\} \subset I = \{1, 2, \cdots, m\}$, where $1 < l < m$. Then $F(y_I) = G(h(y_J), y_{\bar{J}})$ and

$$f(x_V) = G(h(g_1(x_{A_1}), \cdots, g_l(x_{A_l})), g_{l+1}(x_{A_{l+1}}), \cdots, g_m(x_{A_m})). \tag{6}$$

Let the functions $k$ and $H$ be defined by:

$$k(x_B) := h[g_j(x_{A_j}), j \in J]$$

and

$$H(y, x_{\bar{B}}) := G(y, g(x_{A_{l+1}}), \cdots, g(x_{A_m})),$$

where $B = \bigcup\{A_j, j \in J\}$. Then equation (6) implies $f(x_V) = H(k(x_B), x_{\bar{B}})$, so that $B \in \mu(f)$.

Conversely, let $B \in \mu(f)$, where $B = \bigcup\{A_j, j \in J\}$, and $J = \{1, 2, \cdots, l\}$. Then according to theorem (1) $[B, A_{l+1}, \cdots, A_m\} \in \Gamma(f)$, so that $f$ can be written as:

$$f(x_V) = G(g(x_{A_1}, \cdots, x_{A_l}), g_{l+1}(x_{A_{l+1}}), \cdots, g(x_{A_m})). \tag{7}$$

Since $f$ depends on $A_i, i \in I$, for all $i \in I$ and $y \in \{0, 1\}$ there exists a binary vector $a_i(y)$ such that $y = g_i(a_i(y))$. If $h$ is the function defined by

$$h(y_1, \cdots, y_l) = h(g_1(a_1(y_1)), \cdots, g_l(a_l(y_l))) := g(a_1(y_1), \cdots, a_l(y_l)),$$

then equation (7) implies

$$\begin{aligned} F(y_I) &= F[g_i(a_i(y_i))), i \in I] = f[a_i(y_i)\ i \in I] \\ &= G(g(a_1(y_1), \cdots, a_l(y_l)), y_{l+1}, \cdots, y_m) \\ &= G(h(y_J), y_{\bar{J}}). \end{aligned}$$

This shows that $J \in \mu(F)$. ∎

**Theorem 5** *Let $f$ be a Boolean function defined on the partition $\{A, B, C\}$. Let $A, B, C$ be essential for $f$. If $A \cup B$ and $B \cup C$ are modular sets of $f$, then $A, B, C \in \mu(f)$ and $f = f_A \star f_B \star f_C$, where $\star$ denotes $\vee, \wedge$, or $\oplus$.*

**Proof.** We may assume that $f(x_V) = F(g(x_A, x_B), x_C) = G(x_A, h(x_B, x_C))$, where $g = f_{A\cup B}$ and $h = f_{B\cup C}$. According to theorem (2) there exists a $c$ such that $g(x_A, x_B) = f(x_A, x_B, c) = G(x_A, h(x_B, c)) = G(x_A, k(x_B))$, where $k(x_B) = h(x_B, c)$. Therefore, $B$ is a modular set of the component $g$ of $f$ and since $B$ is essential for $f$ theorem (4) implies that $B \in \mu(f)$. Similarly, there exist a vector $a$ such that $h(x_B, x_C) = f(a, x_B, x_C) = F(g(a, x_B), x_C) = F(G(a, k(x_B)), x_C)$. Furthermore, we have $G(a, k(x_B)) = f(a, x_B, c) = g(a, x_B) = h(x_B, C) = k(x_B)$. From this we conclude that $k = f_B$ and that

$$f(x_V) = F(G(x_A, k(x_B)), x_C) = G(x_A, F(k(x_B), x_C)). \tag{8}$$

Since $C$ is essential for $F$ there exists a vector $d$ such that $F_d(y) := F(y, d) \neq y$. Therefore, $F_d(y) \in \{0, 1, \bar{y}\}$. We now consider the following three cases:

1) $F_d(y) = 0$. Then equation (8) implies $G_0(x_A) := G(x_A, 0) = 0$. Therefore, we have $g(x_A, x_B) = \bar{k}(x_B)G_0(x_A) \vee k(x_B)G_1(x_A) = k(x_B)G_1(x_A)$, where $G_1(x_A) := G(x_A, 1)$. There exists a vector $b$ such that $k(b) = 1$. So, we can write $G_1$ as $G_1(x_A) = g(x_A, b) = f(x_A, b, c)$. Therefore $G_1 = f_A$.

2) $F_d(y) = 1$. In this case we have $G(x_A, 1) = 1$ implying that $g(x_A, x_B) = k(x_B) \vee G_0(x_A)$, and $G_0 = f_A$.

3) $F_d(y) = \bar{y}$. In this case equation (8) yields $\bar{G}(x_A, k(x_B)) = G(x_A, \bar{k}(x_B))$. In particular, since the function $k$ is not identical to one, we have $\bar{G}_0(x_A) = G_1(x_A)$. Therefore, $g(x_A, x_B) = k(x_B) \oplus G_0(x_A)$, and $G_0 = f_A$.

Note, that the cases $G_0 = 0, G_1 = 1$ and $\bar{G}_0 = G_1$ are mutually exclusive. For example, if $G_0 = 0$ and $G_1 = 1$, then $g(x_A, x_B) = k(x_B)$, contrary to our assumption that $f$ depends on $A$. Conclusion: $A$ and $B$ are modular sets of $f$ and $g = f_B \star f_A$, where $\star$ denotes $\vee, \wedge$, or $\oplus$. Similarly, $C$ is a modular set of $f$ and exactly one of the following cases occurs: $F_0 = 0, F_1 = 1$ and $\bar{F}_0 = F_1$. Now consider the following decompositions:

$$f = \bar{g}F_0 \vee gF_1, \quad g = \bar{k}G_0 \vee kG_1 \tag{9}$$

and the cases:

a) $F_0 = G_0 = 0$. Then (9) implies $f = kG_1F_1 = f_Bf_Af_C$.

b) $F_1 = G_1 = 1$. Then $f = k \vee G_0 \vee F_0 = f_B \vee f_A \vee f_C$.

c) $\bar{G}_0 = G_1$ and $\bar{F}_0 = F_1$. Then $f = k \oplus G_0 \oplus F_0 = f_B \oplus f_A \oplus f_C$.

To show that there are no other possible cases we consider the cases:

d) If $F_0 = 0$ and $G_1 = 1$, then:

$$h(x_B, x_C) = k(x_B)F_1(x_C) \text{ and } g(x_A, x_B) = k(x_B) \vee G_0(x_A). \tag{10}$$

Since $f$ depends on $C$ there exists a vector $c$ such that $F_1(c) = 0$. Now assume that $k(b) = 1$ holds. Then by (10) $h(b, c) = 0$ and $g(x_A, b) = 1$, implying that $G_0 = G(x_A, h(b, c)) = F(g(x_A, b), c) = F_1(c) = 0$. This contradicts the assumption $G_1 = 1$. Therefore, $F_1(c)$ implies $\forall b : k(b) = 0$. Contradiction.

e) If $F_0 = 0$ and $\bar{G}_0 = G_1$, then:

$$h(x_B, x_C) = k(x_B)F_1(x_C) \text{ and } g(x_A, x_B) = k(x_B) \oplus G_0(x_A). \tag{11}$$

There exists a vector $c$ such that $F_1(c) = 1$. Now assume that $k(b) = 0$ holds. Then by (11) $h(b, c) = 0$ and $g(x_A, b) = G_0(x_A)$, implying that

$$G_0 = G(x_A, h(b, c)) = F(g(x_A, b), c) = F(G_0(x_A), c) = 0. \tag{12}$$

Since $f$ depends on $A$ there exists a vector $a$ such that $G_0(A) = 0$. Then (12) implies $F_0(c) = 0$ contrary to our assumption $F_1(c) = 1$. From this we conclude: $\forall b :\ k(b) = 1$. Contradiction.

The cases $F_0 = 1$ and $G_0 = 0$, and $F_0 = 1$ and $F_0 = 0$ and $\bar{G}_0 = G_1$, are symmetrical with d) and e). Similarly, the case $\bar{G}_0 = G_1$, and $\bar{F}_0 = F_1$, also leads to a contradiction (we leave this as an exercise).

Conclusion: Cases a), b) and c) are the only possible ones. Therefore, we have proved that $f = f_A \star f_B \star f_C$, where $\star$ is uniquely determined as $\vee, \wedge$, or $\oplus$. ■

**Theorem 6** *Suppose $f$ is a Boolean function defined on the partition $\{A, B, C, D\}$, and $f$ depends on $A, B$ and $C$. If $A \cup B$ and $B \cup C$ are modular sets of $f$, then $A, B, C$ and $A \cup C, A \cup B \cup C \in \mu(f)$. Moreover, $f_{A \cup B \cup C} = f_A \star f_B \star f_C$, where $\star$ denotes $\vee, \wedge$, or $\oplus$.*

**Proof.** Since $A \cup B, B \cup C \in \mu(f)$ there exist functions $F, G$ and $h$ such that $f(x_V) = F(g(x_A, x_B), x_C, x_D) = G(x_A, h(x_B, x_C), x_D)$, where $g = f_{A\cup B}$ and $h = f_{B\cup C}$. Moreover, $g$ depends on $A$ and $B$ and $h$ depends on $B$ and $C$. Since $f$ depends on $A, B$ and $C$ there exists at least one vector $d$ such that the function $f_d = f(x_A, x_B, x_C, d) \notin \{\top, \bot\}$. We first prove the following

Claim 1): If $f_d \notin \{\top, \bot\}$, then $f_d$ depends on $A, B$ and $C$.

Suppose $F_d(y, x_C)$ depends on the variable $y$. Then since $g$ depends on $A$ and $B$ and $f_d = F_d(g(x_A, x_B), x_C)$, the sets $A, B$ and $C$ are essential for $f_d$. Similarly, if $G_d(z, x_A, x_D)$ depends on $z$, then $f_d$ depends on $B$ and $C$. Now assume that $F_d(y, x_C)$ does not depend on $y$. Then we will derive a contradiction as follows: Since $f_d = F_d(g(x_A, x_B), x_C)$ we have: $f_d$ depends on $C$. Therefore, since $f_d = G(x_A, h(x_B, x_C))$, the function $G_d$ depends on the variable $z$, implying that $f_d$ depends on $B$ and $C$. Consequently, $f_d$ depends on $y$, contrary to our assumption. Conclusion: $F_d$ and $G_d$ depend respectively on $y$ and $z$, and $f_d$ depends on $A, B$ and $C$.

Claim 2): If $f_d, f_e \notin \{\top, \bot\}$, then $f_e \in \{f_d, \bar{f}_d\}$.

Suppose $f_d \notin \{\top, \bot\}$. Then theorem (5) implies $f_d = \phi_1 \star \phi_2 \star \phi_3$, such that $g = \phi_1 \star \phi_2$ and $h = \phi_2 \star \phi_3$, where $\star$ is uniquely determined as $\vee, \wedge$, and $\oplus$. Since $\phi_1 \in \mu(g)$ and $g \in \mu(f)$, theorem (4) implies that $\phi_1 \in \mu(f)$. Since $g$ and $h$ are subfunctions of $f$ we have $\phi_1 = f_A$, $\phi_2 = f_B$ and $\phi_3 = f_C$. Similarly, if $f_e \notin \{\top, \bot\}$, then $f_e = \psi_1 \circ \psi_2 \circ \psi_3$ where $\circ$ is uniquely determined as $\vee, \wedge$, and $\oplus$. Moreover, $g = \psi_1 \circ \psi_2$, $h = \psi_2 \circ \psi_3$ and $\psi_i \in \{\phi_i, \bar{\phi}_i\}$. These constraints imply that $f_e \in \{f_d, \bar{f}_d\}$. Therefore, we have proved that $\forall e:\ f_e \in \{\top, \bot, f_d, \bar{f}_d\}$. According to theorem (2) this is equivalent to $A \cup B \cup C$ is a modular set of $f$. So by theorem (5) we have $f_{A\cup B\cup C} = f_A \star f_B \star f_C$, where $\star$ denotes $\vee, \wedge$, or $\oplus$, and $A \cup C \in \mu(f)$.

■

Let $A, B \subseteq V$. Then $A$ and $B$ are called *overlapping* iff $A$ and $B$ are not comparable and $A \cap B \neq \emptyset$. The following theorem is a useful reformulation of theorems (5) and (6) :

**Theorem 7** *Let $f$ be a Boolean function. Suppose $A$ and $B$ are overlapping modular sets of $f$ and that $f$ depends on $AB, A\bar{B}$, and $\bar{A}B$. Then $AB, A\bar{B}, \bar{A}B, A\bar{B} \cup \bar{A}B$ and $A \cup B$ are modular sets of $f$, and $f_{A\cup B} = f_{A\bar{B}} \star f_{A\cap B} \star f_{\bar{A}B}$, where $\star$ is either $\wedge$, $\vee$ or $\oplus$ .*

Theorem (7) is a famous result called the *Three Modules Theorem* of Ashenhurst [2]. But as far as we know this result is due to Singer [26]. For monotone Boolean functions this theorem is reproved in game theory and reliability theory [25]. This fundamental theorem is proved in the literature by using Ashenhurst decomposition charts, expansions of Boolean functions or differential calculus ([1, 2, 15, 18, 17]).

**Example 1** *Let $f$ be function defined by $f = x_1x_3x_4 \vee x_2x_3x_4 \vee x_1x_3x_5 \vee x_2x_3x_5$. Let $A = \{1,2,3\}$, and $B = \{3,4,5\}$. Then $A, B$, $A \cap B, A\bar{B}$ and $\bar{A}B$ are modular and $f = (x_1 \vee x_2)x_3(x_4 \vee x_5)$.*

Let $f$ be a Boolean function defined on the set $V$ and let $\pi = \{A_i \mid i \in I\}$ be a congruence partition of $f$. The set of classes of $\pi$ will also denoted by $V/\pi$. The quotient $F = f/\pi$ is a function defined on the set $I$. By identifying $I$ and $V/\pi$ we define the natural mapping $\theta_\pi : V \mapsto V/\pi$ by: $\theta_\pi(j) = i \Leftrightarrow j \in A_i$. Furthermore, we define the *completion* of a set $C \subseteq V$ as $\pi(C) := \bigcup\{A_i \mid C \cap A_i \neq \emptyset\}$.

**Proposition 2** *If $\pi \in \Gamma(f)$ and $B \in \mu(f)$ then $\pi(B) \in \mu(f)$.*

**Proof.** Let $\pi = \{A_i \mid i \in I\}$. Then by definition $\pi(B) = \bigcup\{A_i \mid B \cap A_i \neq \emptyset\}$. If $B \subseteq A_i$ for some $i$ then $\pi(B) = A_i$. Furthermore, if $\pi(B) = \bigcup\{A_j \mid j \in J \subseteq I\}$, then $\pi(B) = B$. In all other cases there exists a $j$ such that $B$ and $A_j$ are overlapping. According to theorem (7) we have $B \cup A_j \in \mu(f)$. Therefore, $\pi(B) = B \cup \{A_j \mid B$ and $A_j$ are overlapping$\} \in \mu(f)$. ■

**Theorem 8** *If $\pi \in \Gamma(f)$ and $B \in \mu(f)$ then $\theta_\pi(B) \in \mu(f)$.*

**Proof.** Since $\theta_\pi(B) = \theta_\pi(\pi(B))$, this follows from proposition (1) and proposition (2) ■

**Theorem 9** *If $J \in \mu(F)$ then $\theta_\pi^{-1}(J) \in \mu(f)$.*

**Proof.** This follows from proposition (1) ■

We will now collect a number of properties of modular sets proved thusfar:

**Theorem 10** *Let $f$ be a Boolean function defined on $V$ depending on all its variables and let $g$ be a component of $f$ defined on $C \in \mu(f)$. Suppose $\pi \in \Gamma(f)$ and let $F$ be the quotient $f/\pi$. Then:*

$M_0$ : *$\mu(f) = \mu(f^d)$.*

$M_1$ : *$V \in \mu(f)$ and $\{i\} \in \mu(f)$ for all $i \in V$.*

$M_2$ : *If $A, B \in \mu(f)$ are overlapping then the sets $A\bar{B}, AB, B\bar{A}, AB$ and $A\bar{B} \cup B\bar{A}$ all belong to $\mu(f)$.*

$M_3$: *$\mu(g) = \{B \in \mu(f) \mid B \subseteq C\}$.*

*$M_4$: If $B \in \mu(f)$ then $\pi(B) \in \mu(f)$.*

*$M_5$: If $B \in \mu(f)$ then $\theta_\pi(B) \in \mu(f)$.*

*$M_6$: If $J \in \mu(F)$ then $\theta_\pi^{-1}(J) \in \mu(f)$.*

**Proof.** $M_0$ follows from equation (1). $M_1$ is an immediate consequence of the definition of modular sets. The other properties are respectively proved in theorems (4, 6, 2) and theorems (8, 9).

# 4 The set of congruence partitions

Let $f$ be a Boolean function defined on $V$. The set of partitions on $V$ will be denoted by $\Pi(f) = \Pi(V)$. In this section we briefly discuss the structure of the set of congruence partitions $\Gamma(f) \subseteq \Pi(f)$. It is known ([8]) that $\Pi(f)$ is a finite lattice with ordering relation $\pi_1 \leq \pi_2$ denoting that each class of $\pi_1$ is contained in a class of $\pi_2$. In that case $\pi_1$ is called a *refinement* of $\pi_2$, and $\pi_2$ is called a *coarsening* of $\pi_2$. The least uppper bound respectively greatest lower bound of $\pi_1$ and $\pi_2$ are denoted by $\pi_1 \vee \pi_2$ and $\pi_1 \wedge \pi_2$. The partition $\pi_1 \wedge \pi_2$ consists of all non-empty intersections of a class of $\pi_1$ and a class of $\pi_2$. The partition $\pi_1 \vee \pi_2$ is the intersection of all partitions $\pi$ containing $\pi_1$ and $\pi_2$. It is easy to see that if $R$ is a class of $\pi_1 \vee \pi_2$ and if $C$ is a either class of $\pi_1$ or of $\pi_2$ then $C \cap R \neq \emptyset \Rightarrow C \subseteq R$. Therefore each class $R$ of $\pi_1 \vee \pi_2$ can be written as

$$R = P_1 \cup Q_2 \cup P_3 \cup Q_4 \cup \cdots \cup P_l, \tag{13}$$

where respectively the $P_i$ are (possibly not different) classes of $\pi_1$ and the $Q_j$ are (possibly not different) classes of $\pi_2$. Furthermore, $P_i \cap Q_{i+1} \neq \emptyset$ and $Q_i \cap P_{i+1} \neq \emptyset$.

**Proposition 3** *$\Gamma(f)$ is a sublattice of $\Pi(f)$.*

**Proof.** We have to prove that $\Gamma(f)$ is closed under the meet and join operations of $\Pi(f)$. Suppose $\pi_1, \pi_2 \in \Gamma(f)$. Then each class of $\pi_1 \wedge \pi_2$ is the non-empty intersection of two modular sets. Therefore, it follows theorem (7) that $\pi_1 \wedge \pi_2 \in \Gamma(f)$. Similarly, theorem (7) and equation (13) imply that $\pi_1 \vee \pi_2 \in \Gamma(f)$. ■

$\Pi(f)$ and $\Gamma(f)$ contain a finest partition $\pi^0$ consisting of all singleton subsets of $V$ and a coarsest partition $\pi^1$ consisting of a single class namely the set $V$. Let $\pi_1, \pi_2$ be two partitions with $\pi_1 < \pi_2$ then $\pi_2$ *covers* $\pi_1$ if for all partitions $\sigma$ with $\pi_1 < \sigma < \pi_2$ we have either $\sigma = \pi_1$ or $\sigma = \pi_2$. A partition $\pi \in \Gamma(f)$ is called an *atom* of $\Gamma(f)$ if $\pi$ covers $\pi^0$. It is easy to see that $\pi$ is an atom of $\Gamma(f)$ iff $\pi = \{A, \{i\} \mid i \in V \setminus A\}$, where $A \in \mu(f)$ and $f_A$ is prime. Let $\mathcal{P}$ be a finite poset (partially ordered set). If $a, b \in \mathcal{P}$ and $a < b$ then a sequence $a = a_0, a_1, \cdots a_n = b$ is called a *chain* between the endpoints $a$ and $b$ of length $n$ if $a_{i-1} < a_i$ for $i = 1, 2, \cdots, n$. Moreover, a chain is called *maximal* if $a_i$ covers $a_{i-1}$, for $i = 1, 2, \cdots, n$. The poset $\mathcal{P}$ satisfies the *Jordan-Dedekind chain condition* if all maximal chains in $\mathcal{P}$ between two endpoints $a$ and $b$ have the same length.

**Definition 3** *A finite lattice $\mathcal{L}$ is called upper semi-modular if $a_i$ covers $a_1 \wedge a_2$, $i$ =1,2, implies that $a_1 \vee a_2$ covers both $a_1$ and $a_2$.*

**Theorem 11** *Let $f$ be a Boolean function. Then $\Gamma(f)$ is upper semi-modular.*

**Proof.** Suppose $\pi_1, \pi_2 \in \Gamma(f)$ cover $\pi_1 \wedge \pi_2 = \{P_j \mid j \in J\}$. Then there exists exactly one class $A_i$ of $\pi_i$ such that $A_i$ is a union of classes of $\pi_1 \wedge \pi_2$ and $\pi_i = \{A_i, P_k \mid k \in J_i\}$, where $J_i \subset J$, $i = 1, 2$ and $f_{A_i}$ is prime. If $J_1 \cap J_2 = \emptyset$, then $\pi_1 \vee \pi_2 = \{J_1, J_2, P_j \mid j \in J \setminus (J_1 \cup J_2)\}$. If $J_1 \cap J_2 \neq \emptyset$, then $\pi_1 \vee \pi_2 = \{J_1 \cup J_2, P_j \mid j \in J \setminus (J_1 \cup J_2)\}$. Therefore, in both cases $\pi_1 \vee \pi_2$ covers $\pi_1$ and $\pi_2$.

■

The following theorem is proved in the literature, see ([8]).

**Theorem 12** *If $\mathcal{L}$ is a finite upper semi-modular lattice, then $\mathcal{L}$ satisfies the Jordan-Dedekind chain condition.*

As a corollary of theorems (11, 12) we have:

**Theorem 13** *Let $f$ be a Boolean function. Then $\Gamma(f)$ satisfies the Jordan-Dedekind condition.*

Other algebraic properties of congruence partitions are discussed in [21, 22].

# 5 Composition trees

In this section we assume that all Boolean functions depend on all their variables. Composition trees have been studied first by Shapley ([28]) in the context of simple games (monotone Boolean functions) . These trees represent in a compact way all the information on the modular sets of a Boolean function. Although the number of modular sets maybe exponential in the number $n$ of variables, it appears that that the number of nodes in a composition trees is linear in $n$. Let $f$ be a Boolean function defined on $V$. Then $C \in V$ is called a *maximal modular set* of $f$ if $C \in \mu(f)$ and for all $B$ with $C \subset B \neq V$, we have $B \notin \mu(f)$. The set of all maximal modular sets is denoted by $m(f)$. A function $f$ is of *composition type I* if no two maximal modular sets have a non-empty intersection, otherwise $f$ is of *composition type II.* We will show that in the latter case the set of complements of the maximal modular sets, denoted by $\pi_\star$, is a partition of $V$

**Definition 4** *A Boolean function $f$ defined on $V$ is called degenerated iff every non-empty-set $A \subseteq V$ is a modular set of $f$.*

**Theorem 14** *If $f$ is a function on $V = \{x_1, x_2, \cdots, x_n\}$. Then $f$ is degenerated iff $f = y_1 \oplus y_2 \oplus \cdots \oplus y_n$, where $y_i = x_i$ or $\bar{x}_i$.*

**Definition 5** *Let $f$ be a Boolean function. Then $\Delta(f)$ is the set of all congruence partitions $\pi$ such that the quotient $f/\pi$ is degenerated.*

**Corollary 3** *Let $f \in \Delta(f)$ and let $\pi \in \Gamma(f)$. Then every union of classes of $\pi$ belongs to $\mu(f)$. In particular, if $A \in \pi$ then $\bar{A} \in \mu(f)$.*

**Theorem 15** *If $f$ is of type II, then $|m(f)| \geq 3$ and $\pi_\star \in \Delta(f)$.*

**Proof.** Suppose $A, B \in m(f)$ and $A \cap B \neq \emptyset$. Then we have respectively: $A$ and $B$ are overlapping, $V = A \cup B$, $\bar{A} \cap \bar{B} \neq \emptyset$ and $\bar{A} \cup \bar{B} = A\bar{B} \cup B\bar{A} \in \mu(f)$. Since $A \cap B \neq \emptyset$ implies $\bar{A} \cup \bar{B} \neq V$, there exists a $C \in m(f)$ such that $C \supseteq \bar{A} \cup \bar{B}$. From this we conclude that $|m(f)| \geq 3$ and that $\forall C, D \in m(f)$ the following holds: $CD \neq \emptyset, \bar{C}\bar{D} = \emptyset, \bar{C} \cup \bar{D} \neq V$ and $\bar{C} \cup \bar{D} \in \mu(f)$. According to theorem (7) this implies that $U := \bar{C}_1 \cup \bar{C}_2 \cup \cdots \cup \bar{C}_m \in \mu(f)$. Moreover, we claim that $U = V$, for otherwise there would exist a $C_j \in m(f)$ such that $U \leq C_j$, which is clearly a contradiction. Therefore, $\pi_\star$ is a partition of $f$. Now let $J \subseteq I = \{1, 2, \cdots, m\}$. Then $\bigcup\{\bar{C}_j \mid j \in J\} = \overline{\bigcup\{\bar{C}_i \mid i \in \bar{J}\}} = \bigcap\{C_i \mid i \in \bar{J}\} \in \mu(f)$. This proves that $\pi_\star \in \Delta(f)$. ∎

**Theorem 16** *A Boolean function $f$ is of type II iff $|m(f| \geq 3$ and $\Delta(f) \neq \emptyset$.*

**Proof.** The if-part of this theorem follows from theorem (15). Conversely, assume $|m(f)| \geq 3$ and $\pi \in \Delta(f)$. Suppose $f$ is of type II and $\pi^*$ is the maximal disjoint congruence partition of $f$. Let $A$ and $B$ be two classes of $\pi$ such that $A \subseteq C_i$ and $B \subseteq C_j$, where $C_i$ and $C_j$ are different classes of $\pi^*$. Since $A \cup B \in \mu(f)$ and $\forall\ C_k \in \pi^* : A \cup B \not\subseteq C_k$, we have a contradiction. Therefore, $f$ is of type II. ∎

**Corollary 4** *A Boolean function $f$ is of type I iff $|m(f| = 2$ or $\Delta(f) = \emptyset$.*

**Theorem 17** *Suppose $f$ is of type II and let $\pi_\star = \{C_1, C_2, \cdots, C_m\}$. Furthermore, let $\mu_\star(f)$ denote the set of all unions of classes in $\pi_\star$. Then:*

$$\mu(f) = \mu(f_{C_1}) \cup \mu(f_{C_2}) \cup \cdots \cup \mu(f_{C_m}) \cup \mu_\star(f).$$

**Proof.** Let $B \in \mu(f)$. Assume that $B$ is not properly contained in a class of $\pi_\star$ and $B$ is not a union of classes of $\pi_\star$. Then there exists a class $C \in \pi_\star$ such that either $B \supset \bar{C}$ or $B$ and $\bar{C}$ are overlapping, implying $B \cup \bar{C} \in \mu(f)$. However, this contradicts the maximality of $\bar{C}$. ■

**Theorem 18** *If $f$ is of type II and $\pi \in \Delta(f)$ then $\pi_\star \leq \pi$.*

**Proof.** Suppose $\pi \in \Delta(f)$ and $A$ is a class of $\pi$. Then by corollary (3) $\bar{A} \in \mu(f)$. Therefore, the class $A$ cannot be properly contained in a class $C$ of $\pi_\star$. For otherwise $\bar{C} \subset \bar{A}$, contrary to the maximality of $\bar{C}$. So according to theorem (17) every class of $\pi$ is a union of classes of $\pi_\star$, implying that $\pi_\star \leq \pi$. ■

The following theorem shows that $\pi_star$ is the finest partition such that $f/\pi$ is degenerated.

**Theorem 19** *If $\pi_1, \pi_2 \in \Delta(f)$, then $\pi_1 \wedge \pi_2 \in \Delta(f)$.*

**Proof.** Suppose $\pi_1, \pi_2 \in \Delta(f)$. Then $\pi_\star \leq \pi_1 \wedge \pi_2$. Therefore, $f/\pi_1 \wedge \pi_2$ is a quotient of $f/\pi_\star$. Since $f/\pi_\star$ is degenerated, this implies that $\pi_1 \wedge \pi_2 \in \Delta(f)$. ■

Based on the two composition types we can construct a *composition tree* $\mathcal{T}(f)$ for a Boolean function $f$ defined on $V$ :

1) The root of $\mathcal{T}$ is the set $V$. Each node of $\mathcal{T}$ is a modular set of $f$.

2) If $C$ is a node and $f_C$ is of type I, then $f_C$ has a maximal disjoint decomposition $\pi^\star = \{C_1, C_2, \cdots, C_m\}$. Then $C_1, C_2, \cdots, C_m$ are the children of node $C$, and node $C$ is labeled with $P(rime)$.

3) If $f_C$ is of type II, then $\Delta(f_C)$ has a finest partition $\pi_\star = \{C_1, C_2, \cdots, C_m\}$, with $m \geq 3$. Then $C_1, C_2, \cdots, C_m$ are the children of node $C$, and node $C$ is labeled with $D(egenerated) : D_\wedge, D_\vee$ or $D_\oplus$.

4) The leaves of $\mathcal{T}(f)$ are the singleton sets $\{i\} \in \mu(f)$.

**Example 2** *Let $f$ and $g$ be positive functions defined by:*
$f = (x_1 \vee x_2)x_3(x_4 \vee x_5) = x_1x_3x_4 \vee x_2x_3x_4 \vee x_1x_3x_5 \vee x_2x_3x_5$, *and*
$g = x_1x_2x_4 \vee x_1x_3x_4 \vee x_2x_3x_4 \vee x_1x_2x_5x_6 \vee x_1x_3x_5x_6 \vee x_2x_3x_5x_6 \vee x_4x_5x_6 \vee$
$\vee x_1x_2x_7 \vee x_1x_3x_7 \vee x_2x_3x_7 \vee x_4x_7$.
*Then* $m(f) = \{\{1,2,3\},\{3,4,5\},\{1,2,4,5\}\}$ *and* $\pi_\star(f) = \{\{4,5\},\{1,2\},\{3\}\}$.
*Moreover,* $m(g) = \{\{1,2,3\},\{4\},\{5,6,7\}\}$. *The modular trees of these functions are given in figure (1). Note that although a function with $|m(f)| = 2$ is prime, the corresponding node in the tree is labeled as $D$.*

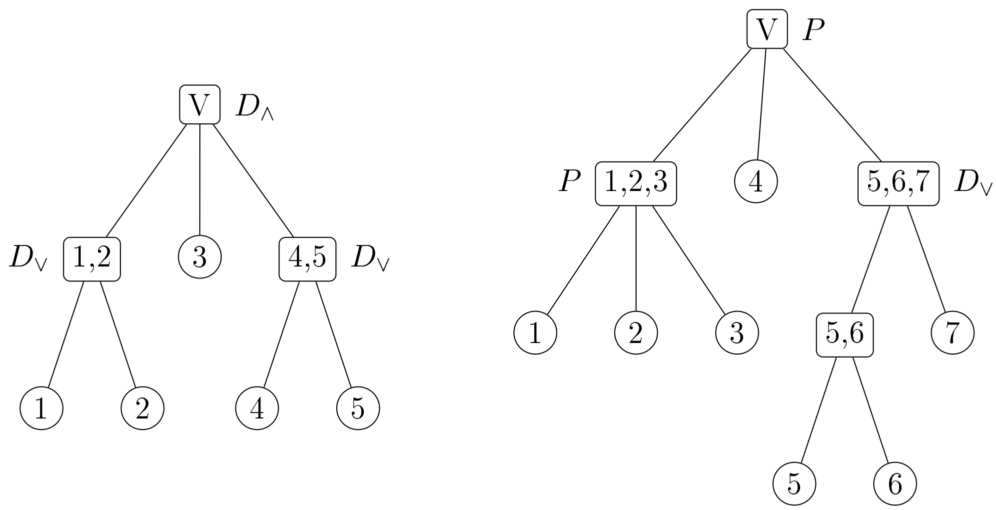


Figure 1: Modular decomposition of $f$ and $g$

**Theorem 20** *Let $f$ be a Boolean function defined on $V$. A subset of $C \subseteq V$ is a modular set of $f$ iff one of the following holds:*

*a) $C$ is a node of $\mathcal{T}(f)$.*

*b) $C$ is the union of children of a node of type $D$.*

**Proof.** This follows from theorem (10:3) and theorem (17). ∎

**Theorem 21** *Let $f$ be a Boolean function defined on $V$ and let $|\mathcal{T}|$ denote the number of nodes of the modular tree $\mathcal{T}$. Then $|\mathcal{T}(f)| \leq 2|V| - 1$.*

**Proof.** We use induction on $n = |V|$. The assertion is trivial for $n = 2$. Suppose the the assertion is true for all Boolean functions with $n \leq k$. Now consider a function $f$ on $V$ with $|V| = k + 1$. Let $C_1, C_2, \cdots, C_m$ be the children of node $V$, where $m \geq 2$. Then $\sum_{i=1}^{m} |C_i| = |V|$. We then have

$$|\mathcal{T}(f)| = 1 + \sum_{i=1}^{m} |\mathcal{T}(C_i)| \leq 1 + \sum_{i=1}^{m}(2|C_i| - 1) = 1 + 2|V| - m \leq 2|V| - 1.$$

■

# 6 The complexity of recognizing modular sets

In this section we prove that for general Boolean functions the problem of recognizing modular sets (called MODULAR) is coNP-complete. In switching theory this complexity has not been discussed. In this context modular sets and decompositions are based on the evaluation of Ashenhurst decomposition charts or by using differential calculus [1, 2, 15, 18, 17]. It has been shown in [22, 21] that the algorithms for the determination of modular sets is exponential in the number of variables. However, here we will study the complexity of the recognition problem of Boolean functions given in DNF-form. In particular we will discuss the following problems:

**Problem MODULAR**
*Given*: A Boolean function $f$ in DNF defined on $V$ and a set $C \subset V$ that contains at least one essential variable of $f$.
*Question*: Is $C$ a modular set of $f$ ?

We relate this problem to the following recognition problem:

**Problem COMPLEMENT**
*Given*: Boolean functions $f$ and $g$ in DNF.
*Question*: $f = \bar{g}$ ?

It is easy to see that this problem is (polynomial) equivalent to the the problem whether two functions $f$ and $g$ are mutually dual: $f = g^d$. It is well known that this problem is coNP-complete, see e.g. ([3]).

**Theorem 22** *Problem MODULAR is coNP-complete.*

**Proof.** Suppose $g_1$ and $g_2$ are Boolean functions given in DNF on $A = \{x_1 \cdots, x_n\}$. Define the function $f$ on $A \cup \{x, y\}$ as:

$$f = xg_1 \vee yg_2. \tag{14}$$

If $g_2 = \bar{g}_1$, then $A$ is a modular set of $f$. Conversely, suppose $A$ is modular and $A$ contains essential variables of $f$. Then there exists a pair of binary values $(x_0, y_0)$ such that the function $g$ defined by $g = f(x_0, y_0, x_A)$ is non-trivial. Furthermore, according to theorem (2) for all *fixed* $x$ and $y$ the function $h(x_A) = f(x, y, x_A)$ is constant or identical to the function $g$ or its complement. From equation (14) it follows that $h \in \{\perp, g_2, g_1, g_1 \vee g_2\}$. Therefore, we have $g_2 = \bar{g}_1$. Conclusion: $g_2 = \bar{g}_1 \Leftrightarrow A$ is modular. This shows that the problem MODULAR is coNP-hard. To prove that this problem is in coNP we note that according to theorem (2) $A$ is a modular set of $f$ iff for all binary vectors $b$ the function $f_b := f(x_A, b) \in \{\top, \perp, g, \bar{g}\}$, where $g$ is a component of $f$ on $A$. Therefore, the set $A$ is *not* modular iff there exist binary vectors $b_1$ and $b_2$ such that $f_{b_1}, f_{b_2} \notin \{\top, \perp\}$ and $f_{b_1} \neq \bar{f}_{b_2}$. Equivalently, the set $A$ is not modular iff there exist three different binary vectors $a, a_1, a_2$, and two different vectors $b_1, b_2$ such that $f_{b_1}(a) = f_{b_2}(a) \neq f_{b_1}(a_1) = f_{b_2}(a_2)$. This shows that problem MODULAR is in coNP. ∎

# 7 Decomposition of monotone Boolean functions

## 7.1 Introduction

If $f$ is a monotone Boolean function defined on $N$ and $f(x_N) = F(g(x_A), x_B)$, then there exist uniquely determined monotone Boolean functions $G$ and $h$ such that $f(x_N) = G(h(x_A), x_B)$. This shows that the decomposibility of a monotone function within the class of monotone functions is the same as in the class of general Boolean functions. Therefore, we will restrict the decompositions of a monotone Boolean function to the class of all monotone Boolean functions. This implies that if $A \in \mu(f)$, then a (monotone) component $g$ of $f$ defined on $A$ is uniquely determined. Furthermore, we show that a set $A$ is a modular set of a monotone function $f$ iff there exists a monotone Boolean function $g$ such that for all binary vectors $b$ we have $f_b(x_A) := f(x_A, b) \in \{\perp, \top, g\}$. The other results obtained in the previous sections also apply to monotone Boolean functions. In particular, the decomposition tree contains all the available information about the modular sets of a monotone function $f$. However, if $f$ is monotone function and if a node of the composition tree is of type II, then only the cases $D_\vee$ or $D_\wedge$ can occur.

## 7.2 Preliminaries

**Notations:** Let $f$ be a positive function defined on $N$. Then a subset $A \subseteq N$ will be represented frequently by its characteristic vector $a := char(A) \in \{0,1\}^n$, with $n = |N|$. If $A = \emptyset$ then this will be denoted by $a = \mathbf{0}$, where $\mathbf{0}$ is the all-zero vector. If $A \subseteq N$, then the functions $f(a = 0)$ and $f(a = 1)$ are the *restrictions* of $f$ defined on the set $\bar{A}$ by setting all variables in $A$ to 0 respectively 1. Similarly, the function $f(\bar{a} = 1)$ is the restriction of $f$ to $A$ defined by setting all variables in $\bar{A}$ to 1, see example (4). However, where needed we will consider all these restrictions of $f$ as functions defined on $N$ by adding dummy (non-essential) variables. Furthermore, the set of all essential variables of $f$ is called the *support set* of $f$. This set is denoted by $S(f)$, and the vector $char(S(f))$ is denoted by $\sigma(f)$.
As known a positive Boolean function has a unique irredundant DNF consisting of all prime implicants. The set of prime implicants correspond to the set of minimal true vectors of $f$, denoted by $\min T(f)$. It is well-known that $\min T(f^d)$ represents the set of minimal transversals of $\min T(f)$. The complement of a false vector is a transversal: $f(x) = 0 \Leftrightarrow f^d(\bar{x}) = 1$.
If $v, w \in \{0,1\}^n$, then $v \wedge w$ (also denoted by $vw$ ), and $v \vee w$ denote respectively the vectors obtained by applying component-wise the and-operation and the or-operation to the vectors $v$ and $w$. Finally, we will denote the variables of a positive function by their index and $+$ denotes the $\vee$-operation.

**Example 3** *Let $f$ be the function defined by $f(x) = x_1x_2 \vee x_2x_3$. Then :*
*$f$ is denoted as: $f = 12 + 23$. Furthermore, $f^d = (1+2)(2+3) = 2 + 13$, $minT(f) = \{110, 011\}$, and $minT(f^d) = \{010, 101\}$ is the set of the minimal transversals of $minT(f)$. Moreover, 001 is a false vector of $f$ and its complement 110 is a transversal of $minT(f)$.*

The following lemma is easy to prove:

**Lemma 3** *Let $f$ be a monotone function. Then $\sigma(f^d) = \sigma(f)$.*

# 8 Decompositions of monotone Boolean functions

In this and the next sections (only) we frequently use the following definition.

**Definition 6** *A Boolean function $f$ is called monotone if $f$ is monotone non-decreasing and positive if $f$ is monotone and non-trivial.*

**Definition 7** *Let $f$ be a positive function defined on $N$ and let $A \subseteq N$. If $f$ depends on $A$ (i.e $\sigma(f) \wedge a \neq \mathbf{0}$), then the positive function $f^a$ on $A$ is defined by: $minT(f^a) = \{v \mid v \in minT(f), v \wedge a \neq \mathbf{0}\}$, where $a = char(A)$. Otherwise $f^a := \perp$.*

From this definition it follows that every positive Boolean function $f$ can be decomposed as:

$$f = f(a = 0) \vee f^a, \quad \text{where} \quad A \subseteq N. \tag{15}$$

Furthermore, for a monotone Boolean function $f$ Shannon's decomposition has the form:

$$f(x) = f(x_j = 0) \vee x_j f(x_j = 1). \tag{16}$$

**Definition 8** *Let $f$ be a positive function defined on $N$, and $A \subseteq N$. Then the contraction $f_a$ of $f$ on $N$ is defined by $f_a(x_A) = f^a(\bar{a} = 1)(x_A)$, where $a = char(A)$.*

**Example 4** *Let $f$ be the positive function on $\{1, 2, \cdots, 6\}$ defined by:*
$f = 1245 + 126 + 2345 + 236 + 46$ *and let* $A = \{1, 2, 3\}$. *Then* $a = char(A) = 111000$, $f(a = 0) = 46, f^a = 1245 + 126 + 2345 + 236,$ *and* $f_a = 12 + 23$.

It is easy to verify that the following lemma holds:

**Lemma 4** *Let $f$ be a positive function defined on $N$, and let $A \subseteq N$.*
*Then $f_a(x) = 1 \Leftrightarrow \exists v \in minT(f)$ such that $x \geq v \wedge a > \mathbf{0}$.*

The following characterization of the contraction is well-known, see [25]:

**Theorem 23** *Let $f$ be a positive function defined on $N$ and let $A \subseteq N$. Suppose that $a = char(A)$ and $x \leq a$. Then: $f_a(x) = 1 \Leftrightarrow \exists y \leq \bar{a}$ such that $f(y) = 0$ and $f(x \vee y) = 1$.*

**Proof.** Suppose that $x \leq a$ and that $f_a(x) = 1$. Then by lemma (4) $\exists v \in \min T(f)$ such that $x \geq v \wedge a > \mathbf{0}$. Let $y = v \wedge \bar{a}$. Then $y \leq \bar{a}$ and $x \vee y \geq v$. This implies that $f(x \vee y) = 1$. Moreover, since $v \wedge a > \mathbf{0}, wehavev \wedge \bar{a} < v$. Therefore: $f(y) = 0$.
Conversely, suppose $\exists y \leq \bar{a}$ such that $f(y) = 0$ and $f(x \vee y) = 1$. Then $x \vee y \geq v$ for some $v \in \min T(f)$. From this we conclude that $x \geq v \wedge a$ and that $y = y \wedge \bar{a} \geq v \wedge \bar{a}$. From this we derive that $v \not\geq \bar{a}$, for otherwise we would have $y \geq v$, contrary to our assumption that $f(y) = 0$. Conclusion: $\exists v$ such that $v \wedge a \neq \mathbf{0}$. and $x \geq v \wedge a$. According to lemma (4) this is equivalent to $f_a(x) = 1$. ∎

The following theorem shows that if $f$ is a positive function and if $A \in \mu(f)$, then the component $g(x_A)$ of $f$ is just the contraction of $f$ on $A$.

**Theorem 24** *Let $f$ be a positive Boolean function defined on $N$ and let $A \subseteq N$. Then $A$ is modular iff $f^a = f^a(a=1)f_a$.*

**Proof.** If $f$ does not depend on $A$, then $f^a = \perp$, so the theorem is obviously true. If $A \in \mu(f)$, then by definition $s(f) \wedge a \neq \mathbf{0}$ and $f = F(g(x_A), x_B)$, where $\{A, B\}$ is a partition of $N$. Then Shannon's decomposition: $F(y, x_B) = F(y=0) \vee yF(y=1)$, implies the fundamental equation:

$$f = f(a=0) \vee gf(a=1). \tag{17}$$

Furthermore, according to equation (15), $f(a=1) = f^a(a=1) \vee f(a=0)$. Therefore, equation (17) implies that $f^a = gf^a(a=1)$, and that:

$$f = f(a=0) \vee gf^a(a=1). \tag{18}$$

Using the fact that the functions $f(a=0)$ and $f^a(a=1)$ only depend on $B = N \setminus A$, equation (18) implies:

$$f_a(x_A) = f^a(\bar{a}=1)(x_A) = g(x_A).$$

Therefore, we have the decomposition:

$$f = f(a=0) \vee f^a(a=1)f_a. \tag{19}$$

However, equation (19) is equivalent to

$$f^a = f^a(a=1)f_a. \tag{20}$$

Conversely, if equation (20) holds, then $A$ is modular. ■

**Corollary 5** *Let $f$ be a positive function defined on $N$ such that $f$ depends on $A \subseteq N$. If $f(x_N) = F(g(x_A), x_B)$, where $F$ and $g$ are positive functions, then $g = f_a$, with $a = char(A)$.*

**Corollary 6** *Let $f$ be a positive function defined on $N$ and $A \subseteq N$. Then the following assertions are equivalent:*

*a)* $A \in \mu(f)$

*b)* *There exists a positive function $g$ defined on $A$ such that $\forall b : f_b(x_A) := f(x_A, b) \in \{\perp, \top, g\}$.*

**Proof.** If $A \in \mu(f)$, then equation (17) shows that there exists a positive function $g$ on $A$ such that $g(x_A) = f(\bar{a} = 0)(x_A)$. This shows that $g$ is a subfunction of $f$. The corollary is therefore a consequence of theorem (2). ■

**Remark 1** *Note, that the problem of deciding whether a set $A$ is modular or not can be solved in time $O(m^2n^2)$ by checking the equation $f^a = f^a(a = 1)f_a$ !*

**Example 5** *Consider the function $f$ of example (4), and let $A = \{1, 2, 3\}$. Then: $f^a = f^a(a = 1)f_a = (45 + 6)(12 + 23)$.*

### Characterizations of modular sets

The following characterizations of a modular set (except e)) are well-known, see e.g. ([25]):

**Theorem 25** *Suppose that $f$ is a positive function defined on $N$, and $A \subseteq N$. Furthermore, let $\sigma(f) \wedge a \neq \mathbf{0}$, where $a = char(A)$. Then the following assertions are equivalent:*

**a)** *$A$ is a modular set of $f$*

**b)** *$f_a$ is a component of $f$*

**c)** *$A$ is a modular set of $f^a$*

**d)** $(f^d)_a = (f_a)^d$

**e)** *There exists a positive function $g$ defined on $A$ such that*
$\forall b : f_b(x_A) := f(x_A, b) \in \{\bot, \top, g\}$.

**f)** $\forall v, w \in \min T(f^a) : f(va \vee w\bar{a}) = 1$

**g)** $\min T(f^a) = \{va \vee w\bar{a} \mid v, w \in \min T(f^a)\}$

**h)** $\sigma(((f^a)^d)^a) = \sigma(f) \wedge a$.

**Proof.** a) $\Leftrightarrow$ b)$\Leftrightarrow$ c) $\Leftrightarrow$ d) $\Leftrightarrow$ e) $\Leftrightarrow$ f) The equivalence of the assertions a), b), c), e) and f) follows from theorem (24). The equivalence of a) and d) follows from b) and the fact that $g$ is a component of $f$ on $A$ $\Leftrightarrow g^d$ is a component of $f^d$ on $A$ (cf xx).

f) $\Leftrightarrow$ g) Obviously, g) implies f). Conversely, suppose assertion f) holds true, and $z = xa \vee y\bar{a} \in T(f^a)$, with $x, y \in minT(f^a)$. If $z \notin minT(f^a)$, then $z > v$ for some $v \in minT(f^a)$. So at least one of the following inequalities holds true: $xa > va$ or $y\bar{a} > v\bar{a}$.

However, the first inequality implies that $x = xa \vee x\bar{a} > va \vee x\bar{a} \in T(f^a)$, contrary to the minimality of $x$. Similarly, $y\bar{a} > v\bar{a}$, implies that $y = ya \vee y\bar{a} > ya \vee v\bar{a} \in T(f^a)$, contrary to the minimality of $y$. From this we conclude that f) and g) are equivalent.

a) $\Leftrightarrow$ h) Finally, we note that according to theorem (24) $A \in \mu(f) \Leftrightarrow f^a = gh$, where $g$ and $h$ are monotone functions with $\sigma(g) = \sigma(f) \wedge a$, and $\sigma(h) \leq \bar{a}$. However, $f^{ad} = (f^a)^d = g^d \vee h^d$ implies that $f^{ada} = g^d$. Therefore, if $A \in \mu(f)$, then $\sigma(f^{ada}) = \sigma(g^d) = \sigma(g) = \sigma(f) \wedge a$. Conversely, if $\sigma(f^{ada}) = \sigma(f) \wedge a$, then there exists monotone functions $g$ and $h$ such that $f^{ad} = g^d \vee h^d$, with $\sigma(g) = \sigma(f) \wedge a$, and $\sigma(h) \leq \bar{a}$, implying that $f^a = gh$. This establishes the equivalence of the assertions a) and h). ■

**Example 6** *Consider the function* $f = (12 + 23)(45 + 6) = 1245 + 126 + 2345 + 236$. *If* $A = \{1, 2\}$ *or* $A = \{1, 2, 3\}$, *then* $f^{ad} = 2 + 13 + 46 + 56$. *If* $A = \{1, 2, 3\}$, *then* $\sigma(f^{ada}) = a$. *However, if* $A = \{1, 2\}$, *then* $A$ *is not modular because* $\sigma(f^{ada}) \neq a$.

**Proposition 4** *Let* $f$ *be a positive function defined on* $N$, $A \subseteq N$, *and* $a = char(A)$. *If* $A \in \mu(f)$, *then* $minT(f^a(\bar{a} = 1)) = \{va \mid v \in minT(f^a)\}$ *and* $minT(f^a(a = 1)) = \{v\bar{a} \mid v \in minT(f^a)\}$.

**Proof.** Suppose $v, w \in minT(f^a)$ and $v\bar{a} > w\bar{a}$. If $A \in \mu(f)$, then by theorem (25) $v = v\bar{a} \vee va > w\bar{a} \vee va \in T(f^a)$, contrary to the minimality of $v$. Similarly, $va > wa$ cannot be true if $v$ and $w$ are minimal. ■

# 9 The modular closure

Unless stated otherwise we assume that a positive function $f$ depends on all its variables. A central step in the determination of the modular tree of a positive function is the computation of the modular closure of a set. It is proved by Singer (see (5)) that a non-empty intersection of two modular sets of a Boolean function is again modular. Therefore, each subset $A$ of variables is contained in a smallest modular set called the *modular closure* of $A$. The modular closure of a set was first introduced by Billera [6] in the context of clutters.

**Definition 9** *Let* $f$ *be a Boolean function defined on* $N$. *The closure of* $A \subseteq N$ *is defined by:* $Cl_{(f)}(A) = \cap\{B \mid A \subseteq B,\ B \text{ is a modular set of } f\}$.

**Proposition 5** *Let $f$ be a positive function on $N$ and $A \subseteq B \subseteq S(f^a)$, where $a = char(A)$. Then $B \in \mu(f^a) \Leftrightarrow \forall v \in minT(f^{ad}) : b \geq v$ or $b \leq \bar{v}$, where $b = char(B)$.*

**Proof.** Let $L$ and $R$ denote respectively the right side and left side of the equivalence of the proposition. Suppose $R$ is false, then $\exists v \in minT(f^{ad})$ such that $v \wedge b \neq \mathbf{0}$ and $v \wedge \bar{b} \neq \mathbf{0}$. This implies that $f^{ad}(bv) = f^{ad}(b\bar{v}) = 0$. Therefore, $f^a(\overline{vb}) = f^a(\overline{v\bar{b}}) = 1$, and according to theorem (25.f) $\exists x, y \in minT(f^a)$ such that $x \leq \overline{vb}$ and $y \leq \overline{v\bar{b}}$. Let $z = xb \vee y\bar{b}$. Then it is easy to verify that $z \leq \bar{v}$. Suppose $B \in \mu(f^a)$. Then $z \in minT(f^a)$, implying that $f^a(\bar{v}) = 1$. This contradicts the fact that $f^{ad}(v) = 1$. Conclusion: $L \Rightarrow R$. Conversely, suppose that $R$ is true. If $B \notin \mu(f^a)$, then $\exists\ x, y \in minT(f^a)$ such that $z := xb \vee y\bar{b} \notin T(f^a)$. Therefore, $\bar{z} = \bar{x}b \vee \bar{y}\bar{b} \notin T(f^{ad})$. From this it follows that $\exists w \in minT(f^{ad})$ such that $w \leq \bar{x}b \vee \bar{y}\bar{b}$. Since we assume that $R$ is true, we have that either $b \geq w$ or $b \leq \bar{w}$. This means that at least one of the vectors $\bar{x}$ or $\bar{y}$ belongs to $T(f^{ad})$, contrary to the fact that $x, y \in T(f^a)$. Conclusion: $R \Rightarrow L$.

■

**Definition 10** *Let $f$ be a positive function on $N$ and $\emptyset \neq A \subseteq N$. Then we define an equivalence relation $\theta$ on $N$ by: $i\theta j \Leftrightarrow i = j$ or there exists a sequence $i = i_1, \cdots i_k = j$, with $k \geq 2$, such that $i_l$ and $i_{l+1}$ both occur in some $v \in minT(f^{ad})$, $l = 1, \cdots, k-1$.*

**Proposition 6** *Let $f$ be a positive function on $N$ and $\emptyset \neq A \subseteq N$. Then we have:*

$$Cl_{f^a}(A) = \{i : i \in N \text{ and } i\theta j \text{ for some } j \in A\}.$$

**Proof.** Let $B = Cl_{f^a}(A)$ and $R = \{i : i \in N$ and $i\theta j$ for some $j \in A\}$. Then $A \subseteq B \subseteq S(f^a)$, and $B \in \mu(f^a)$. According to proposition (5) we have: $\forall v \in minT(f^{ad})$ either $v \leq b$ or $v \leq \bar{b}$, where $b = char(B)$. Using definition (10) we conclude that $R \subseteq B$. On the other hand this definition implies that: $\forall v \in minT(f^{ad})$ either $v \leq r$ or $v \leq \bar{r}$, where $r = char(R)$. Since $A \subseteq R \subseteq S(f^{ad})$, proposition (5) implies that $R \in \mu(f^a)$. Therefore, we have $R \supseteq B$. This shows that $R = B$. ■

The following theorem [25] relates the modular closure of $f^a$ to the dual of $f^a$:

**Theorem 26** *Let $f$ be a positive function on $N$ and $A \subseteq B \subseteq S(f^a)$, with $a = char(A)$. Then $A \subseteq S(f^{ada}) \subseteq Cl_{f^a}(A) \subseteq Cl_f(A)$.*

**Proof.** Since by assumption $f$ depends on all variables in $N$ we have $A \subseteq S(f^{ada})$. Furthermore, it is easy to verify that proposition (5) implies we that $S(f^{ada}) \subseteq Cl_{f^a}(A)$.

Finally, since $Cl_f(A) \in \mu(f)$, we note that $A \subseteq Cl_f(A) \in \mu(f^a)$. This implies that $Cl_{f^a}(A) \subseteq Cl_f(A)$.

■

**Theorem 27** *Suppose $f$ is a positive function and $u, v \in minT(f^a)$. If $f(ua \vee v\bar{a}) = 0$, then the vector $t := \bar{u}a \vee \bar{v}\bar{a} \in T(f^{ad})$. Furthermore, $\forall w \in minT(f^{ad})$ such that $w \leq t$ we have $\mathbf{0} \not\leq w\bar{a} \leq \sigma(f^{ada})$.*

**Proof.** It is easy to see that $\bar{t} = ua \vee v\bar{a}$, so $t \in T(f^{ad})$. Furthermore, the assumptions imply $w \leq \bar{u}a \vee \bar{v}\bar{a}$, and $\bar{u}, \bar{v} \in F(f^{ad})$. Therefore, since $w \in minT(f^{ad})$ we conclude $w \not\leq \bar{u}a$ and $w \not\leq \bar{v}\bar{a}$, implying $w\bar{u}a \neq \mathbf{0}$ and $w\bar{v}\bar{a} \neq \mathbf{0}$. From this we conclude that $w \leq \sigma(f^{ada})$ and that $w\bar{a} \neq \mathbf{0}$. ■

Given $t$, then a vector $w$ in theorem (27) can be determined in time $O(mn^2)$, since it is known that a minimal transversal $w$ can be obtained from a transversal $t$ in $O(n)$ steps. Therefore, if $A$ is not modular, then the last theorem shows that we can determine an element in $Cl_f(A) \setminus A$ given $t$ in time $O(mn^2)$, see also remark (2).

**Definition 11** *Suppose $\exists u, v \in \min T(f^a)$ such that $f(ua \vee v\bar{a}) = 0$. Then we call the vector $ua \vee v\bar{a}$ a culprit of $f$ with respect to $a$.*

The following lemma is of independent interest:

**Lemma 5** *Suppose $f$ is a positive Boolean function and $f^d(w) = 1$.
Let $v \in argmin\{|uw| \mid u \in minT(f)\}$. Then for all unit vectors $e \leq wv$ there exits a vector $w_0 \in minT(f^d)$ such that $e \leq w_0 \leq w$.*

**Proof.** Since $w\bar{v} \wedge v = \mathbf{0}$ and $v \in minT(f)$ we conclude that $w\bar{v} \notin T(f^d)$. On the other hand we claim that

$$w\bar{v} \vee e \in T(f^d). \tag{21}$$

To prove this claim we suppose that $u \in minT(f)$ but $(w\bar{v} \vee e) \wedge u = \mathbf{0}$. Then we have $e \not\leq u$ and $w\bar{v}u = \mathbf{0}$. However, the last equality implies $wu \leq v$, implying

$$\mathbf{0} \neq wu \leq wv. \tag{22}$$

By the minimality assumption we then have $wu = wv$. Since $e \not\leq u$ and $e \leq wv$, this is a contradiction. This proves our claim (21). Furthermore we claim that:

$$\forall w_0 \in minT(f^d) \text{ such that: } w_0 \leq w\bar{v} \vee e, \text{ we have } e \leq w_0. \tag{23}$$

To prove claim (23), assume $e \not\leq w_0$. Then we would have: $w_0 \leq w\bar{v}$. However, $w\bar{v} \notin T(f^d)$, so $w_0 \not\leq w\bar{v}$. Contradiction. This finishes our proof. ■

The following lemma is a reformulation of a proposition in [25]:

**Lemma 6** *Suppose $f$ is a positive Boolean function and $f^d(w) = 1$. Let $c$ be a vector such that $U = \{u \in minT(f) \mid uwc = \mathbf{0}\} \neq \emptyset$. Let $v \in argmin_{u \in U}\{\ |uw|\ \}$. Then for all unit vectors $e \leq wv$ there exits a vector $w_0 \in minT(f^d)$ such that $e \leq w_0 \leq w$.*

**Proof.** Note, that the inequality (22) implies: $wuc \leq wvc = \mathbf{0}$, so $u \in U$. Using this observation the proof of this lemma is the same as the proof of lemma (5). ■

The following fundamental theorem is a variation of a theorem in [25]:

**Theorem 28** *Let $f$ be a positive function. Suppose $t$ is the complement of a culprit of $f$ with respect to to $a$. Then $U = \{u \in minT(f^a) \mid uta = \mathbf{0}\} \neq \emptyset$. Furthermore, if $u_0 \in argmin_{u \in U}\{\ |ut|\ \}$, then $\mathbf{0} \neq u_0 t = u_0 t\bar{a} \leq Cl_f(a)$.*

**Proof.** Since $t$ is the complement of a culprit we have $\exists v, w \in minT(f^a)$ such that $t = \bar{v}a \vee \bar{w}\bar{a}$, and $f^{ad}(t) = 1$. Furthermore, since $\bar{v}a \notin T(f^{ad})$ there must exist a vector $u_0 \in minT(f^a)$ such that $u_0\bar{v}a = \mathbf{0}$. From $u_0 ta = u_0\bar{v}a = \mathbf{0}$ it follows that $u_0 \in U$. Now suppose $u_0 \in argmin_{u \in U}\{\ |ut|\ \}$, then according to lemma (6): for all unit vectors $e \leq u_0 t$ we have: $\exists t_0 \in minT(f^{ad})$ such that $e \leq t_0 \leq t$. Now theorem (26) implies $\mathbf{0} \neq t_0\bar{a} \leq \sigma(f^{ada})$. Therefore, we have: $\mathbf{0} \neq u_0 t = u_0 t\bar{a} \leq Cl_f(a)$. ■

**Remark 2** *The vector $u_0 t$ can be determined in $O(mn)$ time. Therefore, if a culprit is known, then we can determine in $O(mn)$ time an element in $Cl_f(A) \setminus A$.*

# 10 Computational aspects

We have already seen that the recognition problem MODULAR for general Boolean functions is coNP-complete. For positive Boolean functions the situation is quite different. Various decomposition algorithms (in different contexts) are known. Therefore, we briefly discuss the computational aspects of the decomposition of positive Boolean functions. A unified treatment of all algorithms (up to 1990) related to modular sets known in game theory, reliability theory and set systems (clutters) is given by Ramamurthy [25].

### Historical remarks

Let $f$ be a positive function defined on the set $N$, where $|N| = n$, and let $m$ be the number of prime implicants of $f$. Then according to Möhring and Radermacher [21] the modular tree can be computed in time $O(n^3 T(m,n))$, where $T(m,n)$ is the complexity of computing the modular closure of a set $A \subseteq N$. The first known algorithm to compute the modular closure due to Billera [6] is based on computing the dual of $f$. Although this problem is NP-hard in general, for positive functions the complexity of the dualization problem is still not known, although this problem is unlikely to be NP-hard, see e.g [3]. An improvement of Billera's algorithm by Ramamurthy and Parthasarathy [23] also based on dualization has a similar complexity. The first polynomial algorithm given by Möhring and Radermacher (1984) reduced the complexity to $T(m,n) = O(m^3 n^4)$. Subsequently, the complexity was further reduced by Ramamurthy and Parthasarathy [23] and Ramamurthy [25] to respectively $T(m,n) = O(m^3 n^2)$ and $T(m,n) = O(m^2 n^2)$. It is easy to see that the determination of the modular closure can be solved by solving $O(n)$ times the following problem:

**Problem PMODULAR**
*Input*: A Boolean function $f$ with $m$ prime implicants defined on $N$, where $|N| = n$ and $A \subseteq N$.
*Output*: "A is modular" if $A$ is modular. An element $x \in Closure(A) \setminus A$ otherwise.

In the next section we show that the search problem PMODULAR can be solved in time: $O(mn)$. Therefore, the modular closure of a set can be determined in time $T(m,n) = O(mn^2)$.

## 10.1 Solving PMODULAR in time $O(mn)$

Before we solve problem PMODULAR we first show that for positive functions the recognition problem whether a set $A$ is modular or not can be solved in time $O(mn)$.

### Recognition of modular sets

Let $f$ be positive Boolean function $f$ on $N$, $\emptyset \neq A \subseteq N$, and $a = char(A)$. Then we denote $M = \min T(f^a) = \{v_1, \cdots, v_m\}$, $S = \{va \mid v \in M\}$, $T = \{v\bar{a} \mid v \in M\}$, $p = |S|$ and $q = |T|$. Furthermore, without loss of generality we may assume that $M \neq \emptyset$ and that $\forall v \in M = \min T(f^a)$ we have $v \not\leq a$. For each $v \in M$ we can write $v = va \vee v\bar{a}$ as a

$2n$-vector: $(va|v\bar{a})$. Note, that by assumption both vectors $va$ and $v\bar{a}$ are non-zero. We now consider the list of all (column-)vectors:

$$\left| \begin{array}{ccccc} v_1 a & v_2 a & \cdots & \cdots & v_m a \\ v_1 \bar{a} & v_2 \bar{a} & \cdots & \cdots & v_m \bar{a} \end{array} \right| .$$

According to [27], the set of all these $2n$-vectors can be lexicographically sorted in time $O(mn)$.

**Example 7** *Let* $f = 15 + 16 + 245 + 35 + 36 + 46$, *and* $A = \{1, 2, 3, 4\}$. *Then* $f^a = f$ *and the sorted list is given by* $\mathcal{S} = \left| \begin{array}{cccccc} 1 & 1 & 24 & 3 & 3 & 4 \\ 5 & 6 & 5 & 5 & 6 & 6 \end{array} \right|$ . *Note here, that the* $2n$*-vector* $(va|v\bar{a})$ *is denoted by a pair of subsets, e.g. the third column-vector* $(010100|000010)$ *is denoted by* $(24|5)$.

**Theorem 29** *A is modular iff the sorted list of all* $2n$*-vectors has the following structure:*

$$\mathcal{S} = \left| \begin{array}{c} s_1 \cdots s_1 \\ t_1 \cdots t_q \end{array} \right| \begin{array}{c} s_2 \cdots s_2 \\ t_1 \cdots t_q \end{array} \left| \begin{array}{c} \cdots\cdots \\ \cdots\cdots \end{array} \right| \begin{array}{c} s_p \cdots s_p \\ t_1 \cdots t_q \end{array} \right| \text{, where } s_i \in S \text{ and } t_j \in T,$$

*and we have:* $S = minT(f_a)$ *and* $T = minT(f^a(a = 1))$. *So if* $A$ *is modular, then the list* $\mathcal{S}$ *consists of* $p$ *segments of length* $q$, *and* $m = pq$.

**Proof.** According to theorem (25), we have: $A \in \mu(f) \Leftrightarrow f^a = f_a f(a = 1) \Rightarrow S = minT(f_a)$ and $T = minT(f^a(a = 1))$. Furthermore, if $v_1, v_2, w_1, w_2 \in minT(f^a)$, then $v_1 a \vee w_1 \bar{a} = v_2 a \vee w_2 \bar{a} \Leftrightarrow v_1 a = v_2 a$ and $w_1 \bar{a} = w_2 \bar{a}$. ■

**Example 8** *Let* $f$ *be the function of example (4), and let* $A = \{1, 2, 3\}$. *Then we have:* $f^a = 126 + 236 + 1245 + 2345$, *and the sorted list is given by* $\mathcal{S} = \left| \begin{array}{cc} 12 & 12 \\ 45 & 6 \end{array} \right| \begin{array}{cc} 23 & 23 \\ 45 & 6 \end{array} \right|$ . *Therefore,* $A \in \mu(f)$ *and* $p = q = 2$. *Similarly, it can be checked that* $\{1, 3\} \in \mu(f)$.

It is easy to see that the structure $\mathcal{S}$ can be identified in time $O(mn)$, by scanning the list $\mathcal{S}$ from left to right. Therefore, it can be determined in time $O(mn)$ whether a set $A$ is modular or not. However, the more difficult part is to detect an element $x \in Closure(A) \setminus A$ in time $O(mn)$ if $A$ is not modular. According to theorem (7) this can be done in time $O(mn)$ if we can find a culprit in time $O(mn)$.

**Finding a culprit in time $O(mn)$**

Recall that the vector $va \vee w\bar{a}$, with $v, w \in minT(f^a)$ is called a culprit with respect to to $A$ if $f(va \vee w\bar{a}) = 0$. The next basic lemma is used several times in order to find a culprit if it exists. In this lemma the following notations are used: $v \sim w \Leftrightarrow (v < w$ or $v > w)$, and $v \simeq w \Leftrightarrow (v \leq w$ or $v > w)$.

**Lemma 7** *Let $(s_1|t_1)$ and $(s_2|t_2)$ denote any two different columns of the list S. Then:*

**a)** $s_1 \simeq s_2 \Rightarrow t_1 \not\simeq t_2$

**b)** $t_1 \simeq t_2 \Rightarrow s_1 \not\simeq s_2$

**c)** *If $s_1 \sim s_2$ then either $s_1 \vee t_2$ or $s_2 \vee t_1$ is a culprit*

**d)** *If $t_1 \sim t_2$, then either $s_1 \vee t_2$ or $s_2 \vee t_1$ is a culprit*

**e)** *If the 2n-vector $(s_1|t_2)$ does not occur in the list S and $s_1$ and $t_2$ are minimal, then $s_1 \vee t_2$ is a culprit.*

**Proof.** Let $v$ and $w$ be minimal vectors of $f^a$ such that $s_1 = va, s_2 = wa, t_1 = v\bar{a}$ and $t_2 = w\bar{a}$.

c) Suppose $s_1 \sim s_2$, e.g $va > wa$. Then $v = va \vee v\bar{a} > wa \vee v\bar{a}$. Since $v$ is a minimal vector of $f^a$, the vector $wa \vee v\bar{a}$ is a culprit: $f(wa \vee v\bar{a}) = 0$, see theorem (25.f). The assertions a), b) and d) are proved similar.

e) Suppose that the vector $va \vee w\bar{a}$ is not a culprit. then $f(va \vee w\bar{a}) = 1$. Hence, there exists a vector $u \in min(T(f^a)$ such that $u \leq va \vee w\bar{a}$. This implies $ua \leq va$ and $u\bar{a} \leq w\bar{a}$. Since by assumption $va$ and $w\bar{a}$ are minimal, we have $ua = va$ and $u\bar{a} = w\bar{a}$. Therefore, the vector $(va|w\bar{a}) = (ua|u\bar{a})$ is a column-vector of $\mathcal{S}$, contrary to our assumption. So the vector $va \vee w\bar{a}$ is a culprit. ■

Suppose that $(s_1|t_2)$ does not occur in the list $\mathcal{S}$. Then we can check in $O(mn)$ time whether the elements $s_1$ and $t_2$ are minimal. If both elements are minimal then we can apply assertion e) of lemma (7). Otherwise, we can apply either c) or d). Therefore, we have the following corollary:

**Corollary 7** *If $(s_1|t_2)$ does not occur in the list S, then a culprit can be found in time $O(mn)$.*

**Example 9** *Consider the sorted list in example (7):* $\mathcal{S} = \left| \begin{array}{cc|cccc} 1 & 1 & 24 & 3 & 3 & 4 \\ 5 & 6 & 5 & 5 & 6 & 6 \end{array} \right|$ . *Then the first segment has length* $q = 2$. *Since the first element of the fourth column is not equal to* 24 *we detect that the column* $(24|6)$ *is not in* $\mathcal{S}$. *However, 246 (=010101) is* not *a culprit, because the element 24 is not minimal. By scanning the first row we discover that 4 is comparable with 24. Hence, by lemma (7.c) applied to the third and last column, either 246 or 45 is not a true vector of* $f^a$. *In this case 45 (=000110) is a culprit, because* $(4|5)$ *is not in* $\mathcal{S}$ *(see lemma (7.a)) and the elements 4 and 5 are minimal.*

We will now describe our algorithm to decide if a set $A$ is modular or otherwise to find a culprit, given the sorted listed $\mathcal{S}$. The overall algorithm is given in the procedure Modular.

Modular($\mathcal{S}$, **var** *culprit*):
    $flag := false; culprit := false$
    call FirstSegment
    **while** $flag = true$ **do** call NextSegment

The procedure Firstsegment scans the list $\mathcal{S}$ from left to right, by comparing the elements in the first row with the first element $s_1$. The procedure first deals with the special case $s_1 \neq s_2$. If $s_1 \not\sim s_2$ and the length of the first segment $q > 1$, then $(s_1 \mid t_{j_0})$ is not in $\mathcal{S}$, where $j_0 := \min\{j \mid t_j \neq t_1\}$. In that case we return a culprit by applying corollary (7). Note, that we will indicate the application of this corollary in the procedures Firstsegment and Nextsegment by : return culprit*. On the other hand if one of these procedures detect two comparable elements in $\mathcal{S}$ then application of lemma (7.c(d)) is sufficient to find a culprit. This will be denoted by: return culprit. In the procedure Firstsegment we determine the length of the first segment and the first element in the next segment. The i-th column of each next segment is denoted by $(S_i|T_i)$. In particular the beginning of each next segment is given by $(S_1 \mid T_1)$. While there is a next segment, i.e if there is an element $s_i \neq S_1$ and if $s_i \not\sim S_1$ we set $flag = true$, and we start the procedure Nextsegment. However, if $s_i \sim S_1$, then we apply lemma (7.c) to determine a culprit. Both procedures determine the beginning of the next segment by updating the variable *index*.

FirstSegment($\mathcal{S}$, **var** $index, flag, culprit, p, q$):
    **if** $s_1 \neq s_2$ **then**
        **if** $s_1 \sim s_2$ **then** return *culprit*
            **else if** $\forall j > 1\ t_j = t_1$ **then** return $(q = 1, p = m)$
                **else** $j_0 := \min\{j \mid t_j \neq t_1\}$

(so $(s_1|t_{j_0})$ is not in $\mathcal{S}$) return *culprit*∗
**else if** $\forall i > 2\ s_i = s_1$ **then** return $(p = 1; q = m)$
**else** $i_0 := \min\{i \mid s_i \neq s_1\}$;
**if** $s_{i_0} \sim s_1$ **then** return *culprit*
**else** return $(q = i_0 - 1, p = m/q, index = q + 1, flag = true)$

In the next example we have $s_1 \sim s_2$.

**Example 10** *Suppose that* $A = \{1, 2, 3, 4\}$ *and that* $f^a = 15 + 124 + 234 + 345$. *Then* $\mathcal{S} = \left|\begin{array}{c|ccc} 1 & 14 & 34 & 34 \\ 5 & 2 & 2 & 5 \end{array}\right|$. *In this case* $q = 1$. *However* $14 = s_2 \sim s_1 = 1$. *By applying lemma (7.a) it follows that 12 (= 11000) is a culprit.*

The procedure Nextsegment also detects whether the length $l$ of each next segment is equal to $q$. If $l < q$, then $(S_1 \mid T_{l+1})$ is not in the list $\mathcal{S}$. If $l > q$ then $(s_1 \mid T_{q+1})$ is not in the list $\mathcal{S}$. In both cases we apply corollary (7) to find a culprit.

NextSegment($\mathcal{S}$, **var** $index, flag, culprit$):
$flag := false; i := 2; S_1 := s_{index}$
**while** $S_i = S_1$ **do** $i := i + 1$
$l := i - 1$
**if** $l \neq q$ **then** (note: either $(S_1 \mid T_{l+1})$ or $(s_1 \mid T_{q+1})$ is not in $\mathcal{S}$) return *culprit*∗
**else** call Compare
**if** $S_{q+1} \sim S_1$ **then** return *culprit*
**else** return $(flag = true, index = q + 1)$

The next example shows a list $\mathcal{S}$ with an 'incomplete' segment:

**Example 11** *Suppose* $\mathcal{S} = \left|\begin{array}{cc|c} 1 & 1 & 2 \\ 3 & 4 & 3 \end{array}\right|$. *Then* $q = 2$. *However, the second segment is 'incomplete'. In this case procedure Nexsegment detects that* $(2|4)$ *is not in the list and that 24 (=0101) is a culprit.*

Even if all the elements in the first row of a segment are equal (implying that the elements of the second row of that segment are all different), we still have to compare all the elements of the second row with those of the first segment. This comparison is made in the procedure Compare called in the procedure Nextsegment.

Compare ($T$, **var** *culprit*):

  $culprit := false$

  **if** $\forall j \in \{1, \cdots q\}$ $T_j = t_j$ **then** return

    **else** $j_0 := \min\{j \mid T_j \neq t_j\}$

  **if** $T_{j_0} \sim t_{j_0}$ **then** return *culprit*

    **else** $((s_1 \mid T_{j_0})$ or $(S_{j_0} \mid t_1)$ is not in $\mathcal{S})$ return $culprit*$

**Example 12** *Suppose* $\mathcal{S} = \left| \begin{matrix} 1 & 1 \\ 3 & 4 \end{matrix} \right| \begin{matrix} 2 & 2 \\ 3 & 45 \end{matrix} \right|$ . *Then* $q = 2$ *and* $p = 2$*. However, the second row of the second segment differs from the second row of the first segment. In this case procedure Compare detects that 24 (=0101) is a culprit.*

The preceding arguments and theorem (7) show that we have proved the following theorem:

**Theorem 30** *Procedure Modular checks in time* $O(mn)$ *whether a set* $A$ *is modular or not. If* $A$ *is not modular then procedure modular returns a culprit in time* $O(mn)$*. Therefore, if* $A$ *is not modular, an element in* $Cl_f(A) \setminus A$ *can be detected in time* $O(mn)$*.*

### 10.1.1 Computing the modular tree

In this subsection we assume that $f$ is a positive function defined on $N$ with $n = |N| \geq 2$ and $\sigma(f) = N$. Furthermore, we assume that $f$ is given by $minT(f)$ and that $|minT(f)| = m$. Recall that $m(f)$ denotes the set of all maximal modular sets of $f$. We also refer to the results in section (5).

**Lemma 8** *We can determine a* $C \in m(f)$ *in time* $O(nT(m, n))$*.*

**Proof.** Let $i \in N$, then we can construct the series of modular closures:
$\{i\} = C_0 \subset C_1 \cdots \subset C_k = C$, where $C_{i+1} = Cl_f(C_i \bigcup \{j\})$ by choosing some $j \in \bar{C}_i$, with $Cl_f(C_i \bigcup \{j\}) \neq N$. If such an element $j$ does not exist, then $C = C_i \in m(f)$. Since $k \leq n$, the set $C$ can be computed in time $O(nT(m, n))$. ■

**Proposition 7** *The set* $m(f)$ *can be determined in time* $O(n^2T(m, n))$*.*

**Proof.** We first construct a $C_1 \in m(f)$ using the procedure discussed in lemma (8). in the same way we can construct a maximal modular set $C_2$ using and element $i \in \bar{C}_1$. Suppose $C_1 \bigcap C_2 = \emptyset$, and $|m(f)| = k$. If $C_1, C_2 \cdots, C_l \in m(f)$, then $l < k$ iff $D :=$

$C_1 \bigcup C_2 \cdots, \bigcup C_l \subset N$. Let $j \in \bar{D} \neq \emptyset$. Then we determine $C_{l+1} \in m(f)$ such that $j \in C_{l+1}$. If $C_1 \bigcap C_2 \neq \emptyset$, then $l < k$ iff $E := \bar{C}_1 \bigcup \bar{C}_2 \cdots, \bigcup \bar{C}_l \subset N$. Therefore, if $l < k$ and $C \in m(f) \setminus \{C_1, C_2 \cdots, C_l\}$, then $C \supseteq E$. Now we construct $C_{l+1} \in m(f)$ such that $C_{l+1} \supseteq E$. Since $|m(f)| \leq n$, it follows that $m(f)$ can be generated in time $O(n^2T(m,n))$. ■

**Theorem 31** *The modular tree of $f$ can be determined in time $O(n^3T(m,n))$.*

**Proof.** Let $\mathcal{T}(f)$ denote the modular tree of $f$. We have already established in theorem (21) that $|\mathcal{T}(f)| \leq 2n-1$. Since the leaves of $\mathcal{T}(f)$ are the singleton sets of $N$, it follows that the number of internal nodes of the tree is less than or equal to $n-1$. Suppose $C$ is an internal node and $m(f_c) = \{C_1, C_2, \cdots, C_k\}$, where $k \leq n$, and $c = char(C)$. Then $m(f_c)$ can be determined in time $O(n^2T(m,n))$. Note here that if $C \in \mu(f)$, then $minT(f_c)$ can be determined in time $O(mn)$, see proposition (4). If $C_1 \bigcap C_2 = \emptyset$, then the children of $C$ are the nodes $C_1, C_2, \cdots, C_k$. Otherwise, the children of $C$ are $\bar{C}_1, \bar{C}_2, \cdots, \bar{C}_k$. Since there are at most $n-1$ internal nodes, it follows that $\mathcal{T}(f)$ can be determined in time $O(n^3T(m,n))$. ■

Finally, since we can compute the modular set of a non-empty set $A \subset N$ in time $T(m,n) = O(mn^2)$, we have the following result:

**Corollary 8** *The modular tree of $f$ can be generated in time $O(mn^5)$.*

# 11 Conclusions and future research

Compared with the set theoretic approach used in the literature it appears that the Boolean function approach to modular decomposition is more transparent. Moreover, the approach using generalized Shannon decomposition enabled us to give a unified treatment of many results scattered in the literature. We also derived new results on the complexity of modular decomposition. For monotone Boolean functions the recognition of modular sets and therefore the computation of the modular closure and the modular tree can be reduced with a factor $O(m)$. On the other hand we have proved that for general Boolean functions the recognition problem is coNP-complete.

Since partially defined Boolean functions [11, 10, 19] play an important role in many data mining tasks and in switching theory we consider decomposition theory in data mining also as an important task for further research. Finally decompositions with components restricted to a certain class, e.g. self-dual functions [4] (committees in game theory), matroids [16], regular functions etc. are an interesting topic for future research.

**Acknowledgement:**

I would like to thank Peter Hammer, Endre Boros and Yves Crama for some stimulating discussions and pointers to the literature.

# References

[1] Ashenhurst, R.L. (1952): The Decomposition of Switching Functions. Bell Laboratories' Report No. BL-1(II) (reprinted in [15])

[2] Ashenhurst, R.L. (1959): The decomposition of switching functions. Proc. International Symposium on the Theory of Switching, Part I (vol. XXIX, Ann. Computation Lab. Harward), Harward University Press, Cambridge, 75–116

[3] Bioch, J.C., Ibaraki, T. (1995): Complexity of Identification and Dualization of Positive Boolean Functions. Information and Computation 123, 50–63

[4] Bioch, J.C., Ibaraki, T., Makino, K. (1999): Minimum self-dual decompositions of positive dual-minor Boolean functions. Discrete Applied Mathematics, 307-326.

[5] Bioch, J.C. (2002): The algorithmic complexity of modular decomposition, Erim Report Series ERS-2001-30-LIS, 14p. , Erasmus University Rotterdam, www.erim.nl.

[6] Billera, L.J. (1970): On the composition and decomposition of clutters. Journal of Combinatorial Theory 11, 234–245

[7] Birnbaum, Z.W., Essary, J.D. (1965): Modules of coherent systems. SIAM Journal of Applied Mathematics 13, 444–462

[8] Birkhoff, G. (1967): Lattice theory. AMS Colloquium Publications, Vol. 25.

[9] Bonizzoni, P., Vedova G.D. (1999): An algorithm for the Modular Decomposition of Hypergraphs. Journal of Algorithms 32, 65–86

[10] Boros, E., Gurvich, V., Hammer, P.L., Ibaraki, T., Kogan, A. (1994): Decomposition of partially defined Boolean functions. Dimacs Technical Report 94-09, Rutgers University, Rutcor Research Report, RRR-13-94, Discrete Applied Math., 62, 51–75

[11] Boros, E., Gurvich, V., Hammer, P.L., Ibaraki, T., Kogan, A. (2001): Structural analysis of partially defined Boolean functions. Annals of Operations Research. Vol. Optimal partitioning of combinatorial structures.

[12] Brandt, Le, V.B., Spinrad, J.P. (1999): Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications

[13] Cournier, A., and Habib M. (1994): A new linear algorithm for modular decomposition, LNCS, vol. 787, 68-84, Springer-Verlag, Berlin

[14] Butterworth, R.W. (1972): A set theoretic treatment of coherent systems. SIAM Journal of Applied Mathematics 22, 590–598

[15] Curtis, H.A. (1962): A New Approach to the Design of Switching Circuits. Van Nostrand, Princeton

[16] Crama, Y., Hammer P.L. (1989) Bimatroidal independence systems, Zeitschrift fr Operations Research 33 (1989) 149-165.

[17] Davio, M., Deschamps, J.P., Thayse, A. (1978): Discrete and Switching Functions, McGraw-Hill

[18] Hu, S.T.(1968): Mathematical theory of switching circuits and automata, University of California Press, Berkely and Los Angeles

[19] Makino, K., Yano, K., Ibaraki, T. (1995) Positive and Horn Decomposibility of Partially Defined Boolean Functions, Rutgers University, Rutcor Research Report, RRR-29-95.

[20] McConnell, R.M., Spinrad J.P. (1996): Linear-time modular decomposition of undirected graphs and efficient orientation of comparabitlity graphs, Proc. of the 5th ACM-SIAM Symposium on Discrete Algorithms, SODA'96, 536-545

[21] Möhring, R.H., Radermachter, F.J. (1984): Substitution decomposition of discrete structures and connections to combinatorial optimization. Annals of Discrete Mathematics, 19, 257–356

[22] Möhring, R.H. (1985/86): Algorithmic aspects of the substitution of decomposition in optimization over relations, set systems and Boolean Functions. Annals of Operations Research 4, 195–225

[23] Ramamurthy, K.G., Parthasarathy, T. (1986): An algorithm to find the smallest committee containing a given set. Opsearch, 23, 1–6

[24] Ramamurthy, K.G. (1988): A new algorithm to find the smallest committee containing a given set of players. Opsearch 25, 49–56

[25] Ramamurthy, K.G. (1990): Coherent Structures and Simple Games. Theory and Decision Library, Series C, Kluwer, Dordrecht

[26] Singer, T. (1953): The Decomposition Chart as a Theoretical Aid. Bell Laboratories' Report No., III-1-III-28, 1953. (reprinted in [15])

[27] Sedgewick R. (1990): Algorithms in C. Addison-Wesley

[28] Shapley, L.S. (1967): On committees. In New Methods of Thought and Procedure, edited by F. Zwicky and A.G. Wilson, Springer-Verlag, New York, 246–270

## Publications in the Report Series Research* in Management

**ERIM Research Program: "Business Processes, Logistics and Information Systems"**

**2002**

*The importance of sociality for understanding knowledge sharing processes in organizational contexts*
Niels-Ingvar Boer, Peter J. van Baalen & Kuldeep Kumar
ERS-2002-05-LIS

*Crew Rostering for the High Speed Train*
Ramon M. Lentink, Michiel A. Odijk & Erwin van Rijn
ERS-2002-07-LIS

*Equivalent Results in Minimax Theory*
J.B.G. Frenk, G. Kassay & J. Kolumbán
ERS-2002-08-LIS

*An Introduction to Paradigm*
Saskia C. van der Made-Potuijt & Arie de Bruin
ERS-2002-09-LIS

*Airline Revenue Management: An Overview of OR Techniques 1982-2001*
Kevin Pak & Nanda Piersma
ERS-2002-12-LIS

*Quick Response Practices at the Warehouse of Ankor*
R. Dekker, M.B.M. de Koster, H. Van Kalleveen & K.J. Roodbergen
ERS-2002-19-LIS

*Harnessing Intellectual Resources in a Collaborative Context to create value*
Sajda Qureshi, Vlatka Hlupic, Gert-Jan de Vreede, Robert O. Briggs & Jay Nunamaker
ERS-2002-28-LIS

*Version Spaces and Generalized Monotone Boolean Functions*
Jan C. Bioch & Toshihide Ibaraki
ERS-2002-34-LIS

*Periodic Review, Push Inventory Policies for Remanufacturing*
B. Mahadevan, David F. Pyke, Moritz Fleischman
ERS-2002-35-LIS

*Modular Decomposition of Boolean Functions*
Jan C. Bioch
ERS-2002-37-LIS

---

* A complete overview of the ERIM Report Series Research in Management:
http://www.ers.erim.eur.nl

ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Marketing
F&A Finance and Accounting
STR Strategy and Entrepreneurship

**2001**

*Bankruptcy Prediction with Rough Sets*
Jan C. Bioch & Viara Popova
ERS-2001-11-LIS

*Neural Networks for Target Selection in Direct Marketing*
Rob Potharst, Uzay Kaymak & Wim Pijls
ERS-2001-14-LIS

*An Inventory Model with Dependent Product Demands and Returns*
Gudrun P. Kiesmüller & Erwin van der Laan
ERS-2001-16-LIS

*Weighted Constraints in Fuzzy Optimization*
U. Kaymak & J.M. Sousa
ERS-2001-19-LIS

*Minimum Vehicle Fleet Size at a Container Terminal*
Iris F.A. Vis, René de Koster & Martin W.P. Savelsbergh
ERS-2001-24-LIS

*The algorithmic complexity of modular decompostion*
Jan C. Bioch
ERS-2001-30-LIS

*A Dynamic Approach to Vehicle Scheduling*
Dennis Huisman, Richard Freling & Albert Wagelmans
ERS-2001- 35-LIS

*Effective Algorithms for Integrated Scheduling of Handling Equipment at Automated Container Terminals*
Patrick J.M. Meersmans & Albert Wagelmans
ERS-2001-36-LIS

*Rostering at a Dutch Security Firm*
Richard Freling, Nanda Piersma, Albert P.M. Wagelmans & Arjen van de Wetering
ERS-2001-37-LIS

*Probabilistic and Statistical Fuzzy Set Foundations of Competitive Exception Learning*
J. van den Berg, W.M. van den Bergh, U. Kaymak
ERS-2001-40-LIS

*Design of closed loop supply chains: a production and return network for refrigerators*
Harold Krikke, Jacqueline Bloemhof-Ruwaard & Luk N. Van Wassenhove
ERS-2001-45-LIS

*Dataset of the refrigerator case. Design of closed loop supply chains: a production and return network for refrigerators*
Harold Krikke, Jacqueline Bloemhof-Ruwaard & Luk N. Van Wassenhove
ERS-2001-46-LIS

*How to organize return handling: an exploratory study with nine retailer warehouses*
René de Koster, Majsa van de Vendel, Marisa P. de Brito
ERS-2001-49-LIS

*Reverse Logistics Network Structures and Design*
Moritz Fleischmann
ERS-2001-52-LIS

*What does it mean for an Organisation to be Intelligent? Measuring Intellectual Bandwidth for Value Creation*
Sajda Qureshi, Andries van der Vaart, Gijs Kaulingfreeks, Gert-Jan de Vreede, Robert O. Briggs & J. Nunamaker
ERS-2001-54-LIS

*Pattern-based Target Selection applied to Fund Raising*
Wim Pijls, Rob Potharst & Uzay Kaymak
ERS-2001-56-LIS

*A Decision Support System for Crew Planning in Passenger Transportation using a Flexible Branch-and-Price Algorithm*
Richard Freling, Ramon M. Lentink & Albert P.M. Wagelmans
ERS-2001-57-LIS

*One and Two Way Packaging in the Dairy Sector*
Jacqueline Bloemhof, Jo van Nunen, Jurriaan Vroom, Ad van der Linden & Annemarie Kraal
ERS-2001-58-LIS

*Design principles for closed loop supply chains: optimizing economic, logistic and environmental performance*
Harold Krikke, Costas P. Pappis, Giannis T. Tsoulfas & Jacqueline Bloemhof-Ruwaard
ERS-2001-62-LIS

*Dynamic scheduling of handling equipment at automated container terminals*
Patrick J.M. Meersmans & Albert P.M. Wagelmans
ERS-2001-69-LIS

*Web Auctions in Europe: A detailed analysis of five business-to-consumer auctions*
Athanasia Pouloudi, Jochem Paarlberg & Eric van Heck
ERS-2001-76-LIS

*Models and Techniques for Hotel Revenue. Management using a Roling Horizon.*
Paul Goldman, Richard Freling, Kevin Pak & Nanda Piersma
ERS-2001-80-LIS

**2000**

*A Greedy Heuristic for a Three-Level Multi-Period Single-Sourcing Problem*
H. Edwin Romeijn & Dolores Romero Morales
ERS-2000-04-LIS

*Integer Constraints for Train Series Connections*
Rob A. Zuidwijk & Leo G. Kroon
ERS-2000-05-LIS

*Competitive Exception Learning Using Fuzzy Frequency Distribution*
W-M. van den Bergh & J. van den Berg
ERS-2000-06-LIS

*Models and Algorithms for Integration of Vehicle and Crew Scheduling*
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-14-LIS

*Managing Knowledge in a Distributed Decision Making Context: The Way Forward for Decision Support Systems*
Sajda Qureshi & Vlatka Hlupic
ERS-2000-16-LIS

*Adaptiveness in Virtual Teams: Organisational Challenges and Research Direction*
Sajda Qureshi & Doug Vogel
ERS-2000-20-LIS

*Assessment of Sustainable Development: a Novel Approach using Fuzzy Set Theory*
A.M.G. Cornelissen, J. van den Berg, W.J. Koops, M. Grossman & H.M.J. Udo
ERS-2000-23-LIS

*Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice*
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-31-LIS

*An NPV and AC analysis of a stochastic inventory system with joint manufacturing and remanufacturing*
Erwin van der Laan
ERS-2000-38-LIS

*Generalizing Refinement Operators to Learn Prenex Conjunctive Normal Forms*
Shan-Hwei Nienhuys-Cheng, Wim Van Laer, Jan Ramon & Luc De Raedt
ERS-2000-39-LIS

*Classification and Target Group Selection bases upon Frequent Patterns*
Wim Pijls & Rob Potharst
ERS-2000-40-LIS

*Average Costs versus Net Present Value: a Comparison for Multi-Source Inventory Models*
Erwin van der Laan & Ruud Teunter
ERS-2000-47-LIS

*Fuzzy Modeling of Client Preference in Data-Rich Marketing Environments*
Magne Setnes & Uzay Kaymak
ERS-2000-49-LIS

*Extended Fuzzy Clustering Algorithms*
Uzay Kaymak & Magne Setnes
ERS-2000-51-LIS

*Mining frequent itemsets in memory-resident databases*
Wim Pijls & Jan C. Bioch
ERS-2000-53-LIS

*Crew Scheduling for Netherlands Railways. "Destination: Curstomer"*
Leo Kroon & Matteo Fischetti
ERS-2000-56-LIS