A Study on Monotone Self-Dual Boolean Functions

Mustafa Altun[a] and Marc D. Riedel[b]
[a] *Electronics and Communication Engineering, Istanbul Technical University, Istanbul , Turkey 34469*
[b] *Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA 55455*
Corresponding Email: altunmus@itu.edu.tr

Abstract

This paper shows that monotone self-dual Boolean functions in irredundant disjuntive normal form (IDNF) do not have more variables than disjuncts. Monotone self-dual Boolean functions in IDNF with the same number of variables and disjuncts are examined. An algorithm is proposed to test whether a monotone Boolean function in IDNF with n variables and n disjuncts is self-dual. The runtime of the algorithm is $\mathcal{O}(n^3)$.

1. Introduction

The problem of testing whether a monotone Boolean function in irredundant disjuntive normal form (IDNF) is self-dual is one of few problems in circuit complexity whose precise tractability status is unknown (Eiter et al., 2008). This famous problem is called the *monotone self-duality problem*. It impinges upon many areas of computer science, such as artificial intelligence, distributed systems, database theory, and hypergraph theory (Makino, 2003; Eiter and Gottlob, 2002).

Consider a monotone Boolean function f in IDNF. Suppose that f has k variables and n disjuncts:

$$f(x_1, x_2, \ldots, x_k) = D_1 \vee D_2 \vee \cdots \vee D_n$$

where each disjunct D_i is a prime implicant of f, $i = 1, \ldots n$. In this study, we show that if f is self-dual then $k \leq n$. We consider the monotone self-duality problem for Boolean functions with the same number of variables and disjuncts (i.e., $n = k$). For such functions, we propose an efficient recursive algorithm that runs in $\mathcal{O}(n^3)$ time.

In their seminal work, Fredman and Khachiyan showed that the duality of monotone Boolean functions f and g both in IDNF can be determined in quasi-polynomial time (Fredman and Khachiyan, 1996). They constructed their work on the relations between the parameters n, m, and k where k is the number of variables of f (or g), and n and m are the number of disjuncts of f and g both in IDNF, respectively. They showed that the number of variables is upper bounded by the product of the numbers of disjuncts in f and g; $k \leq n \times m$. From their result, it is apparent that if f is self-dual (i.e., $f = g$) then $k \leq n^2$. This result holds true for both monotone and non-monotone Boolean functions and it has been widely used in the literature regarding the monotone self-duality problem (Gaur and Krishnamurti, 2000, 2008; Eiter and Gottlob, 2002; Eiter et al., 2008). We improve on this result for monotone Boolean functions. In Section 2, by Corollary 1, we show that if f is self-dual then $k \leq n$.

Our result can also be applied to dual monotone Boolean functions by using the statement in Lemma 4: Boolean functions f and g are dual iff a Boolean function $af \vee bg \vee ab$ is self-dual where a and b are two additional Boolean variables. In Section 2 by Corollary 2, we show that if f and g are monotone dual functions both in IDNF then $k \leq n+m-1$ where k is the number of variables, and n and m are the numbers of disjuncts of f and g, respectively. Prior work has shown that if f and g are dual functions both in IDNF then $k \leq n \cdot m$ (Fredman and Khachiyan, 1996; Elbassioni, 2008; Eiter et al., 2008).

Fredman and Khachiyan's algorithm runs in $N^{o(logN)}$ time to test whether monotone Boolean functions f and g both in IDNF are dual, where N is the total number of disjuncts in f and g (i.e., $N = n + m$). Since the case $f = g$ makes the algorithm check the self-duality of f, the monotone self-duality problem can also be solved in quasi-polynomial time that makes the problem unlikely to be NP hard. The exact time complexity of the problem (in terms of polynomial time solvability) has not been known yet. However, there are polynomial-time algorithms in the literature for the sub-classes of the problem (Boros et al., 1997, 2004; Makino, 2003; Eiter et al., 2008; Elbassioni and Rauf, 2010; Karasan, 2011; Gottlob, 2012). In this study, we define a new sub-class of this famous problem: monotone Boolean functions with the same number of variables and disjuncts (i.e.,$n = k$). For such functions, we propose an efficient recursive algorithm that runs in $\mathcal{O}(n^3)$ time. The algorithm and the underlying mathematics of it are presented in Section 3.

1.1. Definitions

Definition 1 *Consider k independent* **Boolean variables**, $x_1, x_2, \ldots, x_k$. **Boolean literals** *are Boolean variables and their complements, i.e.,* $x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_k, \bar{x}_k$.

Definition 2 *A* **disjunct (D)** *of a Boolean function f is an AND of literals, e.g., $D = x_1\bar{x}_3x_4$, that implies f.* **A disjunct set (SD)** *is a set containing all the disjunct's literals, e.g., if $D = x_1\bar{x}_3x_4$ then $SD = \{x_1, \bar{x}_3, x_4\}$. A* **disjunctive normal form (DNF)** *is an OR of disjuncts.*

Definition 3 *A* **prime implicant (PI)** *of a Boolean function f is a disjunct that implies f such that removing any literal from the disjunct results in a new disjunct that does not imply f.*

Definition 4 *An* **irredundant disjunctive normal form (IDNF)** *is a DNF where each disjunct is a PI of a Boolean function f and no PI can be deleted without changing f.*

Definition 5 *If a Boolean function f* **covers** *another Boolean function g then $g = 1$ makes $f = 1$, i.e., g implies f. For example, $f = x_1x_2$ covers (is implied by) $g = x_1x_2x_3$.*

Definition 6 *Boolean functions f and g are* **dual pairs** *iff $f(x_1, x_2, \ldots, x_k) = g^D = \bar{g}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k)$. A Boolean function f is* **self-dual** *iff $f(x_1, x_2, \ldots, x_k) = f^D = \bar{f}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k)$.*

Given an expression for a Boolean function in terms of AND, OR, NOT, 0, and 1, its dual can also be obtained by interchanging the AND and OR operations as well as interchanging the constants 0 and 1. For example, if $f(x_1, x_2, x_3) = x_1x_2 \vee \bar{x}_1x_3$ then $f^D(x_1, x_2, x_3) = (x_1 \vee x_2)(\bar{x}_1 \vee x_3)$. A trivial example is that for $f = 1$, the dual is $f^D = 0$.

Definition 7 *A Boolean function f is* **monotone** *if it can be constructed using only the AND and OR operations (specifically, if it can constructed without the NOT operation).*

Definition 8 **The Fano plane** *is the smallest finite projective plane with seven points and seven lines such that and every pair of its lines intersect in one point.* **A Boolean function that represents the Fano plane** *is a monotone self-dual Boolean function with seven variables and seven disjuncts such that every pair of its disjuncts intersect in one variable. An example is* $f = x_1x_2x_3 \vee x_1x_4x_5 \vee x_1x_6x_7 \vee x_2x_4x_6 \vee x_2x_5x_7 \vee x_3x_4x_7 \vee x_3x_5x_6$.

Definition 9 *Consider a Boolean function* f. *A Boolean function* $f_{x_j=x_i}$ *can be obtained by replacing every* x_j *with* x_i *in* f *where* x_i *and* x_j *are any two variables of* f. *For example, if* $f = x_1x_2 \vee x_1x_3 \vee x_2x_3x_4$ *then* $f_{x_2=x_1} = x_1$ *and* $f_{x_4=x_3} = x_1x_2 \vee x_1x_3 \vee x_2x_3$.

2. Number of disjuncts versus number of variables

Our main contribution in this section is Theorem 1. It defines a necessary condition for monotone self-dual Boolean functions. For such functions, there exists a *matching* between its variables and disjuncts, i.e., every variable can be paired to a distinct disjunct that contains the variable. From this theorem we derive our two main results, presented as Corollary 1 and Corollary 2.

2.1. Preliminaries

We define the **intersection property** as follows. A Boolean function f satisfies the *intersection property* if every pair of its disjuncts has a non-empty intersection.

Lemma 1 *(Fredman and Khachiyan, 1996) Consider a monotone Boolean function* f *in IDNF. If* f *is self-dual then* f *satisfies the intersection property.*

Proof of Lemma 1: The proof is by contradiction. Consider a disjunct D of f. We assign 1's to the all variables of D and 0's to the other variables of f. This makes $f = 1$. If f does not satisfy the intersection property then there must be a disjunct of f having all assigned 0's. This makes $f^D = 0$, so $f \neq f^D$. This is a contradiction. □

Lemma 2 *Consider a monotone Boolean function* f *in IDNF satisfying the intersection property. Suppose that we obtain a new Boolean function* g *by removing one or more disjuncts from* f. *There is an assignment of 0's and 1's to the variables of* g *such that every disjunct of* g *has both a 0 and a 1.*

Proof of Lemma 2: Consider one of the disjuncts that was removed from f. We focus on the variables of this disjunct that are also variables of g. Suppose that we assign 1's to all of these variables of g and 0's to all of the other variables of g. Since f is in IDNF, the assigned 1's do not make $g = 1$. Therefore $g = 0$; every disjunct of g has at least one assigned 0. Since f satisfies the intersection property, every disjunct of g has at least one assigned 1. As a result, every disjunct of g has both a 0 and a 1. □

We define a **matching** between a variable x and a disjunct D as follows. There is a matching between x and D iff x is a variable of D. For example, if $D = x_1x_2$ then there is a matching between x_1 and D as well as x_2 and D.

Lemma 3 *Consider a monotone Boolean function* f *in IDNF satisfying the intersection property. Suppose that* f *has* k *variables and* n *disjuncts. Consider* b *of* f*'s* k *variables where* $b < k$ *and* $b < n$. *If each of these* b *variables can be matched with a distinct disjunct of* f, *and all other unmatched disjuncts of* f *do not have any of the matched* b *variables, then* f *is not self-dual.*

Proof of Lemma 3: Lemma 3 is illustrated in Table 1. Note that a variable x_i is matched with a disjunct D_i for every $i = 1, \ldots, b$. To prove that f is not self-dual, we assign 0's and 1's to the variables of f such that every disjunct of f has both 0 and 1. This results in $f = 0$ and $f^D = 1$; $f \neq f^D$. We first assign 0's and 1's to the variables of $D_{b+1} \vee \ldots \vee D_n$ to make each disjunct of $D_{b+1} \vee \ldots \vee D_n$ have both a 0 and a 1. Lemma 2 allows us to do so. Note that none of the variables $x_1, \ldots, x_b$ has an assignment yet. Since f satisfies the intersection property, each disjunct of $D_1 \vee \ldots \vee D_b$ should have at least one previously assigned 0 or 1. If a disjunct of $D_1 \vee \ldots \vee D_b$ has a previously assigned 1 then we assign 0 to its matched (circled) variable; if a disjunct of $D_1 \vee \ldots \vee D_b$ has a previously assigned 0 then we assign 1 to its matched (circled) variable. As a result, every disjunct of f has both a 0 and a 1; therefore f is not self-dual.

D_1	D_2	....	D_{b-1}	D_b	D_{b+1}...............D_n
.	.	....	.	.	....
.	.	....	.	.	....
x_1	x_2	....	x_{b-1}	x_b	....
					no $x_1, \ldots\ldots, x_b$

Table 1: An illustration of Lemma 3.

□

Lemma 4 *(Eiter and Gottlob, 1995) Boolean functions f and g are dual pairs iff a Boolean function $af \vee bg \vee ab$ is self-dual where a and b are two additional Boolean variables.*

Proof of Lemma 4: From the definition of duality, if $af \vee bg \vee ab$ is self-dual then $(af \vee bg \vee ab)_{a=1,\ b=0} = f$ and $(af \vee bg \vee ab)_{a=0,\ b=1} = g$ are dual pairs. From the definition of duality, if f and g are dual pairs then $(af \vee bg \vee ab)^D = (a^D \vee f^D)(b^D \vee g^D)(a^D \vee b^D) = (a \vee g)(b \vee f)(a \vee b) = (af \vee bg \vee ab)$. □

2.2. The Theorem

Theorem 1 *Consider a monotone Boolean function f in IDNF. If f is self-dual then each variable of f can be matched with a distinct disjunct.*

Before proving the theorem we elucidate it with examples.

Example 1 *Consider a monotone self-dual Boolean function in IDNF*

$$f = x_1x_2 \vee x_1x_3 \vee x_2x_3.$$

The function has three variables x_1, x_2, and x_3, and three disjuncts $D_1 = x_1x_2$, $D_2 = x_1x_3$, and $D_3 = x_2x_3$. As shown in Table 2, every variable is matched with a distinct disjunct; the circled x_1, x_2, and x_3 are matched with D_1, D_3, and D_2, respectively. We see that the theorem holds for this example. Note that the required matching – each variable to a distinct disjunct – might not be unique. For this example, another possibility is having x_1, x_2, and x_3 matched with D_2, D_1, and D_3, respectively.

Example 2 *Consider a monotone self-dual Boolean function in IDNF*

$$f = x_1x_2x_3 \vee x_1x_3x_4 \vee x_1x_5x_6 \vee x_2x_3x_6 \vee x_2x_4x_5 \vee x_3x_4x_6 \vee x_3x_5.$$

D_1	D_3	D_2
x_2	x_3	x_1
x_1	x_2	x_3

Table 2: An example to illustrate Theorem 1.

The function has six variables x_1, x_2, x_3, x_4, x_5, and x_6, and seven disjuncts $D_1 = x_1x_2x_3$, $D_2 = x_1x_3x_4$, $D_3 = x_1x_5x_6$, $D_4 = x_2x_3x_6$, $D_5 = x_2x_4x_5$, $D_6 = x_3x_4x_6$, and $D_7 = x_3x_5$. As shown in Table 3, every variable is matched with a distinct disjunct; the circled x_1, x_2, x_3, x_4, x_5, and x_6 are matched with D_1, D_4, D_2, D_5, D_3, and D_6, respectively. We see that the theorem holds for this example.

D_1	D_4	D_2	D_5	D_3	D_6	D_7
x_2	x_3	x_1	x_2	x_1	x_3	
x_3	x_6	x_4	x_5	x_6	x_4	x_3
x_1	x_2	x_3	x_4	x_5	x_6	x_5

Table 3: An example to illustrate Theorem 1.

Proof of Theorem 1: The proof is by contradiction. We suppose that at most a variables of f can be matched with distinct disjuncts, where $a < k$. We consider two cases, $n = a$ and $n > a$ where n is the number of disjuncts of f. For both cases, we find an assignment of 0's and 1's to the variables of f such that every disjunct of f has both a 0 and a 1. This results in a contradiction since such an assignment makes $f = 0$ and $f^D = 1$; $f \neq f^D$.

Case 1: $n = a$.

This case is illustrated in Table 4. To make every disjunct of f have both a 0 and a 1, we first assign 0 to x_1 and 1 to x_{a+1}. Then we assign a 0 or a 1 to each of the variables $x_2, \ldots, x_a$ step by step. In each step, if a disjunct has a previously assigned 1 then we assign 0 to its matched (circled) variable; if a disjunct has a previously assigned 0 then we assign 1 to its matched (circled) variable. After these steps, if every disjunct of f has both a 0 and a 1 then we have proved that f is not self-dual. If there remain disjuncts, these disjuncts should not have any previously assigned variables. Lemma 3 identifies this condition and it tells us that f is not self-dual. This is a contradiction.

D_1	D_2	....	D_{n-1}	D_n
.	.	....	.	.
x_{a+1}	.	....	.	.
x_1	x_2	....	x_{a-1}	x_a

Table 4: An illustration of Case 1.

Case 2: $n > a$

This case is illustrated in Table 5. We show that f always satisfies the condition in Lemma 3; accordingly f is not self-dual.

As shown in Table 5, the expression $D_{a+1} \vee \ldots \vee D_n$ does not have the variable x_1 or the variable x_{a+1}. If it had then at least $a+1$ variables would be matched; this would go against our assumption. For example, if $D_{a+1} \vee \ldots \vee D_n$ has x_1 then x_1 would be matched with a disjunct from $D_{a+1} \vee \ldots \vee D_n$ and x_{a+1} would be matched with D_1. So $a+1$ variables would be matched with distinct disjuncts.

D_1	D_2	....	D_{a-1}	D_a	D_{a+1} .... D_n
.	.	....	.	.	....
x_{a+1}	.	....	.	.	....
x_1	x_2	....	x_{a-1}	x_a	....
					no x_1 no x_{a+1}

Table 5: An illustration of Case 2.

If $D_{a+1} \vee \ldots \vee D_n$ does not have any of the variables $x_2, \ldots, x_a$ then f satisfies the condition in Lemma 3; f is not self-dual. If it does then the number of disjuncts not having x_1 or x_{a+1} increases. This is illustrated in Table 6. Suppose that $D_{a+1} \vee \ldots \vee D_n$ has variables $x_j, \ldots, x_{a-1}$ where $j \geq 2$. As shown in the table, $D_j \vee \ldots \vee D_n$ does not have x_1 or x_{a+1}. If it had then at least $a+1$ variables would be matched; this would go against our assumption. For example, if D_j had x_{a+1} then x_{a+1} would be matched with D_j and x_j would be matched with a disjunct from $D_{a+1} \vee \ldots \vee D_n$. So $a+1$ variables would be matched with distinct disjuncts.

D_1	D_2	......	D_{j-1}	D_j	......	D_{a-1}	D_a	D_{a+1} .... D_n
.	.	......	.	.	......	.	.	....
x_{a+1}	.	......	.	.	......	.	.	....
x_1	x_2	......	x_{j-1}	x_j	......	x_{a-1}	x_a	....
				no x_1 no x_{a+1}				

Table 6: An illustration of Case 2.

If $D_j \vee \ldots \vee D_n$ does not have any of the variables $x_2, \ldots, x_{j-1}$ then f satisfies Lemma 3; f is not self-dual. If it does have any of these variables then the number of disjuncts not having x_1 or x_{a+1} increases.

As a result the number of disjuncts not having x_1 or x_{a+1} increases unless the condition in Lemma 3 is satisfied. Since there must be disjuncts having x_1 or x_{a+1}, this increase should eventually stop. When it stops, the condition in Lemma 3 will be satisfied. As a result, f is not self-dual. This is a contradiction.

$\square$

Corollary 1 *Consider a monotone Boolean function f in IDNF. Suppose that f has k variables and n disjuncts. If f is self-dual then $k \leq n$.*

Proof of Corollary 1: We know that if f is self-dual then f should satisfy the matching defined in Theorem 1. This matching requires that f does not have more variables than disjuncts, so $k \leq n$. $\square$

Corollary 2 *Consider monotone Boolean functions f and g in IDNF. Suppose that f has k variables and n disjuncts and g has k variables and m disjuncts. If f and g are dual pairs then $k \leq n+m-1$.*

Proof of Corollary 2:

From Lemma 4 we know that the Boolean functions f and g are dual pairs iff a Boolean function $af \vee bg \vee ab$ is self-dual where a and b are Boolean variables. If neither a nor b is a variable of f (or of g) then $af \vee bg \vee ab$ has $n+m+1$ disjuncts and $k+2$ variables. From Corollary 1, we know that $k+2 \leq n+m+1$, so $k \leq n+m-1$. □

3. The self-duality problem

In this section we propose an algorithm to test whether a monotone Boolean function in IDNF with n variables and n disjuncts is self-dual. The runtime of the algorithm is $\mathcal{O}(n^3)$.

3.1. Preliminaries

Theorem 2 *(Altun and Riedel, 2010, 2012) Consider a disjunct D_i of a monotone self-dual Boolean function f in IDNF. For any variable x of D_i there exists at least one disjunct D_j of f such that $SD_i \cap SD_j = \{x\}$.*

Before proving the theorem we elucidate it with an example.

Example 3 *Consider a monotone self-dual Boolean function function in IDNF*

$$f = x_1x_2x_3 \vee x_1x_3x_4 \vee x_1x_5x_6 \vee x_2x_3x_6 \vee x_2x_4x_5 \vee x_3x_4x_6 \vee x_3x_5.$$

The function has seven disjuncts $D_1 = x_1x_2x_3$, $D_2 = x_1x_3x_4$, $D_3 = x_1x_5x_6$, $D_4 = x_2x_3x_6$, $D_5 = x_2x_4x_5$, $D_6 = x_3x_4x_6$, and $D_7 = x_3x_5$. Consider the disjunct $D_1 = x_1x_2x_3$. Since $SD_1 \cap SD_3 = \{x_1\}$, $SD_1 \cap SD_5 = \{x_2\}$, and $SD_1 \cap SD_6 = \{x_3\}$, the theorem holds for any variable of D_1. Consider the disjunct $D_2 = x_1x_3x_4$. Since $SD_2 \cap SD_3 = \{x_1\}$, $SD_2 \cap SD_4 = \{x_3\}$, and $SD_2 \cap SD_5 = \{x_4\}$, the theorem holds for any variable of D_2.

Proof of Theorem 2: The proof is by contradiction. Suppose that there is no disjunct D_j of f such that $SD_i \cap SD_j = \{x\}$. From Lemma 1, we know that D_i has a non-empty intersection with every disjunct of f. If we extract x from D_i then a new disjunct D_i' should also have a non-empty intersection with every disjunct of f. This means that if we assign 1's to the all variables of D_i' then these assigned 1's make $f = f^D = (1+\ldots)(1+\ldots)\ldots(1+\ldots) = 1$. So D_i' implies f; D_i' is a disjunct of f. This disjunct covers D_i. However, in IDNF, all disjuncts including D_i are irredundant, not covered by another disjunct of f. So we have a contradiction □

Lemma 5 *Consider a disjunct D of a monotone self-dual Boolean function f in IDNF. Consider a variable x of D. Suppose that f has exactly y disjuncts $D_1,\ldots,D_y$ such that $SD \cap SD_i = \{x\}$ for every $i = 1,\ldots,y$. A Boolean function $g = (D_{x=1})((D_1 \vee \ldots \vee D_y)_{x=1})^D$ implies (i.e., is covered by) f.*

Before proving the lemma we elucidate it with an example.

Example 4 *Consider a monotone self-dual Boolean function function in IDNF*

$$f = x_1x_2x_3 \vee x_1x_3x_4 \vee x_1x_5x_6 \vee x_2x_3x_6 \vee x_2x_4x_5 \vee x_3x_4x_6 \vee x_3x_5.$$

The function has seven disjuncts $D_1 = x_1x_2x_3$, $D_2 = x_1x_3x_4$, $D_3 = x_1x_5x_6$, $D_4 = x_2x_3x_6$, $D_5 = x_2x_4x_5$, $D_6 = x_3x_4x_6$, and $D_7 = x_3x_5$. Consider the disjunct $D_1 = x_1x_2x_3$. The disjunct $D_3 = x_1x_5x_6$ is the only disjunct that intersects D_1 in x_1. Since $g = \left((D_1)_{x_1=1}\right)\left((D_3)_{x_1=1}\right)^D = x_2x_3x_5 \vee x_2x_3x_6$ implies f, the lemma holds for this case. The disjuncts $D_6 = x_3x_4x_6$ and $D_7 = x_3x_5$ are the only disjuncts that intersect D_1 in x_3. Since $g = \left((D_1)_{x_3=1}\right)\left((D_6 \vee D_7)_{x_1=1}\right)^D = x_1x_2x_4x_5 \vee x_1x_2x_5x_6$ implies f, the lemma holds for this case.

Proof of Lemma 5: To prove the statement we check if $g = 1$ always makes $f = f^D = 1$ (by assigning 1's to the variables of g). Suppose that f has n disjuncts $D_1, \ldots, D_y, D, D_{y+2}, \ldots, D_n$. If $g = 1$ then both $(D_{x=1}) = 1$ and $((D_1 \vee \ldots \vee D_y)_{x=1})^D = 1$. From Lemma 1, we know that if $(D_{x=1}) = 1$ then every disjunct of $D_{y+2}, \ldots \vee, D_n$ has at least one assigned 1. From the definition of duality, we know that if $((D_1 \vee \ldots \vee D_y)_{x=1})^D = 1$ then every disjunct of $D_1, \ldots, D_y$ has at least one assigned 1. As a result, every disjunct of f has at least one assigned 1 making $f = f^D = (1 + \ldots) \ldots (1 + \ldots) = 1$. □

Lemma 6 *Consider a monotone self-dual Boolean function f in IDNF with k variables. Consider any b of f's k variables where $b < k$. A set of these b variables has a non-empty intersection with at least $b+1$ disjunct sets of f.*

Before proving the lemma we elucidate it with an example.

Example 5 *Consider a monotone self-dual Boolean function function in IDNF*

$$f = x_1x_2x_3x_4 \vee x_1x_5 \vee x_1x_6 \vee x_2x_5x_6 \vee x_3x_5x_6 \vee x_4x_5x_6.$$

The function has six disjuncts $D_1 = x_1x_2x_3x_4$, $D_2 = x_1x_5$, $D_3 = x_1x_6$, $D_4 = x_2x_5x_6$, $D_5 = x_3x_5x_6$, and $D_6 = x_4x_5x_6$. Consider a set of two variables $\{x_2, x_3\}$; $b = 2$. Since it has a non-empty intersection with three disjunct sets SD_1, SD_4, and SD_5, the lemma holds for this case. Consider a set of one variable $\{x_1\}$; $b = 1$. Since has a non-empty intersection with three disjunct sets SD_1, SD_2, and SD_3, the lemma holds for this case.

Proof of Lemma 6: The proof is by contradiction. From Theorem 1, we know that each of the k variables should be matched with a distinct disjunct, so a set of b variables of f should have a non-empty intersection with at least b disjunct sets of f. Suppose that a set of b variables of f has a non-empty intersection with exactly b disjunct sets of f. Lemma 3 identifies this condition and it tells us that f is not self-dual. This is a contradiction. □

Theorem 3 *Consider a monotone self-dual Boolean function f in IDNF with k variables. Consider any b of f's k variables where $b < k-1$. If every variable of f occurs at least three times then a set of the b variables has a non-empty intersection with at least $b+2$ disjunct sets of f where $b < k-1$.*

Proof of Theorem 3: The proof is by induction on b.
The base case: $b = 1$.

Since a variable of f occurs three times, a set of one variable should have a non-empty intersection with at least three disjunct sets of f.
The inductive step: Assume that the theorem holds for $b \le m$ where $m \ge 2$. We show that it also holds for $b = m+1$.

Consider a set of $m+1$ variables $S = \{x_1, \ldots, x_{m+1}\}$. Consider a disjunct D of f such that $SD \cap S = \{x_1, \ldots, x_c\}$. From Theorem 2, we know that there is at least one disjunct that intersects D in x_i for every $i = 1, \ldots, c$. We consider two cases.

For the cases we suppose that f does not have a disjunct set intersecting S in one variable; if it does then the theorem holds for S (by using the inductive assumption). Also we suppose that f does not have a disjunct set that is a subset of S; if it does then it is obvious that the theorem holds for S.

Case 1: There is only one disjunct that intersects D in x_i for every $i = 1, \ldots, c$.

Suppose that D_i is the only disjunct that intersects D in x_i for every $i = 1,\ldots,c$. Consider a variable set $SD_{x_1-x_c}$ of $((D_1)_{x_1=1} \vee \ldots \vee (D_c)_{x_c=1})$; $SD_{x_1-x_c}$ includes all variables of $((D_1)_{x_1=1} \vee \ldots \vee (D_c)_{x_c=1})$. From Lemma 5, we know that $\Big((D)_{x_i=1}\Big)\Big((D_i)_{x_i=1}\Big)^D$ implies f for every $i = 1,\ldots,c$. This means that f should have at least $|SD_{x_1-x_c} \cap S|$ disjuncts such that each of them has one distinct variable from $SD_{x_1-x_c} \cap S = \{x_{c+1}, x_{c+2}, \ldots, x_{m+1}\}$ and none of them is covered by $(D \vee D_1 \vee \ldots \vee D_c)$.

If $SD_{x_1-x_c} \cap S = \{x_{c+1}, x_{c+2}, \ldots, x_{m+1}\}$ then f has at least $|SD_{x_1-x_c} \cap S| = m - c + 1$ disjunct sets such that each of them intersects $\{x_{c+1}, x_{c+2}, \ldots, x_{m+1}\}$ in one variable. Therefore, including SD, SD_1, SD_2, ..., and SD_c, f has at least $m + 2$ disjunct sets such that each of them has a non-empty intersection with S. If f has exactly $m+2$ disjunct sets then each disjunct of f has a non-empty intersection with $(x_{c+1}x_{c+2}\ldots x_{m+1})(D_{x_1=1,\ldots,x_c=1})$. This means that f should have a disjunct that covers $(x_{c+1}x_{c+2}\ldots x_{m+1})(D_{x_1=1,\ldots,x_c=1})$. Since none of the $m+2$ disjuncts covers $(x_{c+1}x_{c+2}\ldots x_{m+1})(D_{x_1=1,\ldots,x_c=1})$, f needs one more disjunct to cover $(x_{c+1}x_{c+2}\ldots x_{m+1})(D_{x_1=1,\ldots,x_c=1})$ that has a non-empty intersection with S. This is a contradiction. As a result, f has at least $m + 3$ disjunct sets such that each of them has a non-empty intersection with S; the theorem holds for S.

If $SD_{x_1-x_c} \cap S = \{x_{c+1}, x_{c+2}, \ldots, x_n\}$ where $n < m + 1$ then from our inductive assumption we know that the variable set $\{x_{n+1}, x_{n+2}, \ldots, x_{m+1}\}$ intersects at least $m - n + 3$ disjunct sets. As a result, f has at least $(c + 1) + |SD_{x_1-x_c} \cap S| = (n - c) + (m - n + 3) = m + 4$ disjunct sets such that each of them has a non-empty intersection with S. So the theorem holds for S.

Case 2: For at least one of the variables of $x_1,\ldots,x_c$, say x_c, there are at least two disjuncts such that each of them intersects D in x_c.

The proof has c steps. In each step, we consider all disjuncts of f such that each of them intersects D in x_i where $1 \le i \le c$. We first consider disjuncts $D_1,\ldots,D_y$ such that each of them intersects D in x_1. Consider a variable set SD_{x_1} of $(D_1 \vee \ldots \vee D_y)_{x_1=1}$; SD_{x_1} includes all variables of $(D_1 \vee \ldots \vee D_y)_{x_1=1}$. From Lemma 5, we know that $(D_{x_1=1})((D_1 \vee \ldots \vee D_y)_{x_1=1})^D$ implies f. Therefore along with $D_1 \vee \ldots \vee D_y$, f should have disjuncts that cover $(D_{x_1=1})((D_1 \vee \ldots \vee D_y)_{x_1=1})^D$. This means that f includes a dual-pair of $(D_1 \vee \ldots \vee D_y)_{x_1=1}$ and $((D_1 \vee \ldots \vee D_y)_{x_1=1})^D$. From Lemma 4 and Lemma 6, we know that $SD_{x_1} \cap S$ requires at least $|SD_{x_1} \cap S| + 1$ disjunct sets of f such that each of them has a non-empty intersection with S.

We apply the same method for x_2, x_3, and x_{c-1}, respectively. Consider a variable set SD_{x_i} for every $i = 2,\ldots,c-1$; SD_{x_i} is obtained in the same way as SD_{x_1} was obtained in the first step. In each step if $SD_{x_i} \cap S$ has new variables that are the variables not included in $(SD_{x_1} \cup \ldots \cup SD_{x_{i-1}}) \cap S$, then these new variables result in new disjuncts. From Lemma 4 and Lemma 6, we know that the number of new disjuncts is at least one more than the number of the new variables. Therefore before the last step, including SD, f has at least $|(SD_{x_1} \cup \ldots \cup SD_{x_{c-1}}) \cap S| + (c-1) + 1$ disjunct sets (+1 is for SD) such that each of them has a non-empty intersection with S.

The last step corresponds to x_c. If $|(SD_{x_1} \cup \ldots \cup SD_{x_{c-1}}) \cap S| = ((m+1) - c)$ then SD_{x_c} does not have any new variables. Since there are at least two disjuncts such that each of them intersects D in x_c, f has at least $(m+1-c)+(c)+(2) = m+3$ disjunct sets such that each of them has a non-empty intersection with S. So the theorem holds for S. If $|(SD_{x_1} \cup \ldots \cup SD_{x_{c-1}}) \cap S| = n$ where $n < (m+1)-c$ then S has $(m-n-c+1)$ variables that are not included in $((SD_{x_1} \cup \ldots \cup SD_{x_{c-1}}) \cup SD)$. From our inductive assumption, we know that these $(m - n - c + 1)$ variables results in at least $(m-n-c+1+2)$ new disjunct sets. As a result, f has at least $(m+1-c)+(c)+(2) = m+3$ disjunct sets such that each of them has a non-empty intersection with S. So the theorem holds for S. □

Lemma 7 *Consider a monotone self-dual Boolean function f in IDNF with the same number of variables and disjuncts. If f has a variable occurring two times then f has at least two disjuncts of size two.*

Proof of Lemma 7: If a variable of f, say x_1, occurs two times then from Theorem 2, we know that two disjuncts that have x_1 should intersect in x_1. Consider the disjuncts $x_1x_{a1}\dots x_{an}$ and $x_1x_{b1}\dots x_{bm}$ of f. From Lemma 5, we know that both $g = (x_{a1}\dots x_{an})(x_{b1} \vee \dots \vee x_{bm})$ and $h = (x_{b1}\dots x_{bm})(x_{a1} \vee \dots \vee x_{an})$ should be covered by f. Note that g and h have total of $n+m$ disjuncts. These $n+m$ disjuncts should be covered by at most $n+m-2$ disjuncts of f; otherwise Lemma 6 is violated. For example, if $n+m$ disjuncts are covered by $n+m-1$ disjuncts of f then along with the disjuncts $x_1x_{a1}\dots x_{an}$ and $x_1x_{b1}\dots x_{bm}$ there are $n+m+1$ disjuncts having $n+m+1$ variables. This means that a set of the remaining variables, say b variables, has a non-empty intersection with at most b disjuncts of f, so Lemma 6 is violated.

Any disjunct of f with more than two variables can only cover one of the $m+n$ disjuncts of $g \vee h$. Therefore to cover $m+n$ disjuncts of $g \vee h$ with $m+n-2$ disjuncts, f needs disjuncts of size two. Since a disjunct of size two can cover at most two of the $m+n$ disjuncts of $g \vee h$, f should have at least two disjuncts of size two. □

Lemma 8 *Consider a monotone self-dual Boolean function f in IDNF with the same number of variables and disjuncts. If each variable of f occurs at least three times then f is a unique Boolean function that represents the Fano plane.*

Proof of Lemma 8: We consider two cases.
Case 1: A pair of disjuncts of f intersect in multiple variables.

We show that if a pair of disjuncts of f intersect in multiple variables then f is not self-dual. Consider two disjuncts D_1 and D_2 of f such that they intersect in multiple variables. Suppose that both D_1 and D_2 have variables x_1 and x_2. This case is illustrated in Table 7. Note that x_3, x_4, …, x_k are matched with D_3, D_4, …, D_k, respectively. This is called *perfect matching*. Hall's theorem describes a necessary and sufficient condition for this matching: a subset of b variables of $\{x_3,\dots,x_k\}$ has a non-empty intersection with at least b disjunct sets from $SD_3,\dots,SD_k$. From Theorem 3, we know that a set of b variables of f has a non-empty intersection with at least $b+2$ disjunct sets of f. This satisfies the necessary and sufficient condition for the perfect matching between $x_3,\dots,x_k$ and $D_3,\dots,D_k$.

We find an assignment of 0's and 1's to the variables of f such that every disjunct of f has both a 0 and a 1. To make every disjunct of f have both 0 and 1, we first assign 0 to x_1 and 1 to x_2. Then we assign a 0 or a 1 to each of the variables $x_3,\dots,x_k$ step by step. In each step, if a disjunct has a previously assigned 1 then we assign 0 to its matched (circled) variable; if a disjunct has a previously assigned 0 then we assign 1 to its matched (circled) variable. After these steps, if every disjunct of f has both a 0 and a 1 then we have proved that f is not self-dual. If there remain disjuncts, these disjuncts should not have any previously assigned variables. Lemma 3 identifies this condition and it tells us that f is not self-dual.
Case 2: Every pair of disjuncts of f intersect in one variable.

Suppose that a variable of f, say x_1, occurs three times. Consider disjuncts $D_1 = x_1x_{a1}\dots x_{an}$, $D_2 = x_1x_{b1}\dots x_{bm}$, and $D_3 = x_1x_{c1}\dots x_{cl}$ of f where $n \leq m \leq l$. From Lemma 5, we know that f

D_1	D_2	D_3	....	D_{n-1}	D_k
.	.	.	....	.	.
x_2	x_1	.	....	.	.
x_1	x_2	x_3	....	x_{k-1}	x_k

Table 7: An illustration of Case 1.

should cover $(x_{a1} \dots x_{an})(x_{b1} \vee \dots \vee x_{bm})(x_{c1} \vee \dots \vee x_{cl})$ where $n \le m \le l$. This means that f should cover $m{\cdot}l$ disjuncts. These disjuncts are covered by at least $m{\cdot}l$ disjuncts of f; otherwise the intersection property does not hold for f. Along with D_1, D_2, and D_3, f has $m{\cdot}l{+}3$ disjuncts having $m{+}n{+}l{+}1$ variables. From Lemma 1, we know that $m{\cdot}l{+}3 \le m{+}n{+}l{+}1$. The only solution of this inequality is that $n = 2$, $m = 2$, and $l = 2$. This results in a self-dual Boolean function representing the Fano plane, e.g., $f = x_1x_2x_3 \vee x_1x_4x_5 \vee x_1x_6x_7 \vee x_2x_4x_6 \vee x_2x_5x_7 \vee x_3x_4x_7 \vee x_3x_5x_6$.

If a variable of f occurs more than three times then the value on left hand side of the inequality $m \cdot l + 3 \le m + n + l$ increases more than that on the right hand side does, so there is no solution. □

Lemma 9 *A Boolean function f is self-dual iff $f_{x_b=x_a}$, $f_{x_c=x_a}$, and $f_{x_c=x_b}$ are all self-dual Boolean functions where x_a, x_b, and x_c are any three variables of f.*

Proof of Lemma 9: From the definition of duality, f is self-dual iff each assignment of 0's and 1's to the variables of f, corresponding to a row of the truth table, satisfies $f(x_1, x_2, \dots, x_k) = \bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k)$. Any dependency between variables of f only eliminates some rows of f's truth table. Therefore, if f is self-dual then $f_{x_b=x_a}$, $f_{x_c=x_a}$, and $f_{x_c=x_b}$ are all self-dual. For each row of f's truth table either $x_b = x_a$ or $x_c = x_a$, or $x_c = x_b$. Therefore, if $f_{x_b=x_a}$, $f_{x_c=x_a}$, and $f_{x_c=x_b}$ are all self-dual then f is self-dual. □

3.2. The Algorithm

We present a four-step algorithm:
Input: A monotone Boolean function f in IDNF with n variables n disjuncts.
Output: "YES" if f is self-dual; "NO" otherwise.

1. Check if f is a single variable Boolean function. If it is then return "YES".
2. Check if f represents the Fano plane. If it does then return "YES".
3. Check if the intersection property holds for f. If it does not then return "NO".
4. Check if f has two disjuncts of size two, x_ax_b and x_ax_c where x_a, x_b, and x_c are variables of f. If it does not then return "NO"; otherwise obtain a new function $f = f_{x_c=x_b}$ in IDNF. Repeat this step until f consists of a single variable; in this case, return "YES".

If f is self-dual then f should be in one of the following three categories: (1) f is a single variable Boolean function; (2) at least one variable of f occurs two times; (3) each variable of f occurs at least three times. From Theorem 2, we know that if f is self-dual and not in (1) then every variable of f should occur at least two times, so f should be in either (2) or (3). Therefore these three categories cover all possible self-dual Boolean functions.

The first step of our algorithm checks if f is self-dual and in (1). The second step of our algorithm checks if f is self-dual and in (3). From Lemma 8, we know that if f is self-dual and in (3) then f is a unique Boolean function that represents the Fano plane. The third and fourth steps of our algorithm check if f is self-dual and in (2). From Lemma 1, we know that if f is self-dual then f should satisfy the intersection property. From Lemma 7, we know that if f is self-dual and in (2) then f should have at least two disjuncts of size two, $x_a x_b$ and $x_a x_c$. From Lemma 9, we know that f is self-dual iff $f_{x_b=x_a}$, $f_{x_c=x_a}$, and $f_{x_c=x_b}$ are all self-dual. Since f satisfies the intersection property, both $f_{x_b=x_a} = x_a$ and $f_{x_c=x_a} = x_a$ are self-dual. This means that f is self-dual iff $f_{x_c=x_b}$ is self-dual. Note that $f_{x_c=x_b}$ in IDNF has $n-1$ variables and $n-1$ disjuncts. Since $f_{x_c=x_b}$ satisfies the intersection property and does not represent the Fano plane, we just need to repeat step four to check if the function is self-dual. Note that to check if f is self-dual and in (2), we need to repeat step four at most n times.

In step three of the algorithm, we check if the intersection property holds for f. For this task, we first sort the variables of each disjunct. The sorting algorithm runs in $\mathcal{O}(nlogn)$ time for each disjunct with an assumption that a disjunct of f has at most n variables (Papadimitriou, 2003). Since there are total of n disjuncts to be sorted, the total run time is $\mathcal{O}(n^2 logn)$. Secondly, we check the intersection property for every pair of sorted disjuncts of f. This task can be achieved by a comparison algorithm that checks the variables of sorted disjunct pairs. While comparing a disjunct pair , the algorithm starts with the first variables of the disjuncts (variables with the smallest subscripts in each disjunct). In each comparison, if compared variables are not same – otherwise the intersection property holds for f – then a variable with a smaller subscript is replaced by the next variable of the same disjunct. Thus, one variable is eliminated from further comparisons. Since f has n variables, there are at most $n-1$ comparisons. In the last comparison there are two variables left; the other $n-2$ variables are eliminated in the previous $n-2$ comparisons. This results in that the comparison algorithm runs in $\mathcal{O}(n)$ time. Since the number of the disjunct pairs of f is upper bounded by n^2, the total run time is $\mathcal{O}(n^3)$. As a result, the step three of the algorithm runs in $\mathcal{O}(n^2 logn) + \mathcal{O}(n^3) = \mathcal{O}(n^3)$ time. The step four of the algorithm is recursive and reduces the number of variables and disjuncts of f by one in each call. Since f has n variables, this step has n calls (for the worst case). In each call there is an elimination procedure of a variable for which every variable in every disjunct should be checked. Since each of the n disjuncts has at most n variables, each call runs in $\mathcal{O}(n^2)$ time, so the total runtime of step four is $\mathcal{O}(n^3)$.

The steps three and four of the algorithm both run in $\mathcal{O}(n^3)$ time. Therefore the run time of the algorithm is $\mathcal{O}(n^3)$.

4. Conclusion

In this paper, we investigate monotone self-dual Boolean functions. We present many new properties for these type of Boolean functions. Most importantly, we show that monotone self-dual Boolean functions in IDNF (with k variables and n disjuncts) do not have more variables than disjuncts; $k \leq n$. This is a significant improvement over the prior result showing that $k \leq n^2$. We focus on the famous problem of testing whether a monotone Boolean function in IDNF is self-dual. We examine this problem for monotone Boolean functions with the same number of variables and disjuncts; $k = n$. Our algorithm runs in $\mathcal{O}(n^3)$ time. As a future work, we plan to extend our results to testing self-duality of monotone Boolean functions with different $n-k > 0$ values.

References

Altun, M., Riedel, M. D., 2010. Lattice-based computation of Boolean functions. In: Design Automation Conference. pp. 609–612.

Altun, M., Riedel, M. D., 2012. Logic synthesis for switching lattices. IEEE Transactions on Computers 61 (11), 1588–1600.

Boros, E., Elbassioni, K., Gurvich, V., Khachiyan, L., 2004. Generating maximal independent sets for hypergraphs with bounded edge-intersections. LATIN 2004: Theoretical Informatics, 488–498.

Boros, E., Hammer, P., Ibaraki, T., Kawakami, K., 1997. Polynomial-time recognition of 2-monotonic positive boolean functions given by an oracle. SIAM Journal on Computing 26 (1), 93–109.

Eiter, T., Gottlob, G., 1995. Identifying the minimal transversals of a hypergraph and related problems. SIAM Journal on Computing 24 (6), 1278–1304.

Eiter, T., Gottlob, G., 2002. Hypergraph transversal computation and related problems in logic and AI. Lecture Notes in Computer Science 2424, 549–564.

Eiter, T., Makino, K., Gottlob, G., 2008. Computational aspects of monotone dualization: A brief survey. Discrete Applied Mathematics 156 (11), 1952–2005.

Elbassioni, K., 2008. On the complexity of monotone dualization and generating minimal hypergraph transversals. Discrete Applied Mathematics 156 (11), 2109–2123.

Elbassioni, K., Rauf, I., 2010. Polynomial-time dualization of r-exact hypergraphs with applications in geometry. Discrete Mathematics 310 (17), 2356–2363.

Fredman, M. L., Khachiyan, L., 1996. On the complexity of dualization of monotone disjunctive normal forms. Journal of Algorithms 21 (3), 618–628.

Gaur, D. R., Krishnamurti, R., 2000. Self-duality of bounded monotone boolean functions and related problems. In: Algorithmic Learning Theory. Springer, pp. 209–223.

Gaur, D. R., Krishnamurti, R., 2008. Self-duality of bounded monotone Boolean functions and related problems. Discrete Applied Mathematics 156 (10), 1598–1605.

Gottlob, G., 2012. Deciding monotone duality and identifying frequent itemsets in quadratic logspace. arXiv preprint arXiv:1212.1881.

Karasan, O. E., 2011. Incremental polynomial time dualization of quadratic functions and a subclass of degree-k functions. Annals of Operations Research 188 (1), 251–261.

Makino, K., 2003. Efficient dualization of O(log n)-term monotone disjunctive normal forms. Discrete Applied Mathematics 126, 305–312.

Papadimitriou, C. H., 2003. Computational complexity. John Wiley and Sons Ltd.