社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－AL－106 (1)
2006／5／18

最簡な論理式だけを生成するアルゴリズム

天野 一幸

群馬大学工学部情報工学科
376-8515 群馬県桐生市天神町 1-5-1
amano@cs.gunma-u.ac.jp

あらまし n 変数論理関数 f と m 変数論理関数 g に対して，$f \otimes g$ は $(f \otimes g)(x_1, \ldots, x_{nm}) = f(g(\tilde{x}_1), \ldots, g(\tilde{x}_n))$，ただし，$\tilde{x}_i = (x_{(i-1)m+1}, \ldots, x_{im})$ で定義される nm 変数論理関数を表すものとする．本稿では，以下の条件を満たす論理関数のクラス G を与える: 任意の $f = f_1 \otimes \cdots \otimes f_k$ $(f_1, \ldots, f_k \in G)$ に対して，"素直"な構成法によって与えられる論理式が常に f に対する最簡な論理式である．クラス G は，例えば，全ての 2 変数関数，3 変数関数 256 個のうち 134 個などを含む．この結果は，$n = 2^k$ 変数パリティ関数の最簡な論理式のサイズが丁度 n^2 であることを示した Khrapchenko による結果の拡張と見ることができる (全ての i に対して $f_i = x_1 \oplus x_2$ と考える)．また，このような最簡な論理式のみを生成する手続きをも与える．本結果は本質的には，近年，量子敵対者法 [1, 2, 8, 10] において提案された，ある複雑さの尺度を綿密に解析したものである．

A procedure that generates a class of optimal Boolean formulas

Kazuyuki Amano

Dept. of Computer Science, Gunma University
Tenjincyo 1-5-1, Kiryu, Gunma, 376-8515 Japan

Abstract In this paper, we investigate the size of Boolean formulas for composite functions. For two Boolean functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$, $f \otimes g$ denotes a Boolean function on nm variables defined by $(f \otimes g)(x_1, \ldots, x_{nm}) = f(g(\tilde{x}_1), \ldots, g(\tilde{x}_n))$ where $\tilde{x}_i = (x_{(i-1)m+1}, \ldots, x_{im})$. We give a class of base functions G such that for every function of the form $f = f_1 \otimes \cdots \otimes f_k$ with $f_1, \ldots, f_k \in G$, a "naive" construction yields an optimal formula for f. The class G contains every two-variable functions, 134 out of 256 three-variable functions, and more. This can be viewed as a generalization of Khrapchenko's result that says that the formula size of the parity function on $n = 2^k$ variables is n^2, which corresponds to the case $f_i = x_1 \oplus x_2$ for every i. We also give a procedure that recursively generates Boolean formulas whose optimality is guaranteed. Our results are based on a careful inspection of a recently proposed complexity measure SumPI [7], which originated from the quantum adversary method [1, 2, 8, 10].

1 Introduction

To derive a superlinear lower bound on the size of a Boolean circuit for a function in NP is still one of the most challenging open problems in theoretical computer science. The current best lower bound is $5n - o(n)$ [5]. A mild success has been archived for the size of a Boolean formula. A formula is a circuit in which every gate has fan-out exactly one. The current best lower bound for an explicitly defined function is $\Omega(n^{3-o(1)})$ by Håstad [3].

In this paper, we focus on the formula complexity of *composite functions*. For two Boolean functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$, $f \otimes g$ denotes a Boolean function on nm variables defined by $(f \otimes g)(x_1, \ldots, x_{nm}) = f(g(\tilde{x}_1), \ldots, g(\tilde{x}_n))$ where $\tilde{x}_i = (x_{(i-1)m+1}, \ldots, x_{im})$. The starting point of our work is the famous result of Khrapchenko [6] that says that the minimal size of a formula computing the parity function on $n = 2^k$ variables is exactly n^2. The parity function on 2^k variables is expressed as $(x_1 \oplus x_2) \otimes \cdots \otimes (x_1 \oplus x_2)$. Obviously, an optimal formula for $x_1 \oplus x_2$ is $(x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)$, which has size 4. By using this recursively, we get a formula for the parity function on 2^k variables whose size is $4^k = n^2$. Khrapchenko's result implies that such a "naive" construction yields an optimal formula when the base function is $x_1 \oplus x_2$. Obviously, the same property holds when a base function is a read-once formula. So the natural question is then : what happens when we consider other base functions?

One of the motivations for considering the formula complexity of composite functions is as follows: If we have a base function f on n variables with such a property and whose formula complexity is $L(f)$, then the formula complexity of the iterated function of f is $N^{\log_n L(f)}$, where N is the total number of input variables. For small values of n, computing $L(f)$ may be feasible with the aid of a computer. This would bring us a good lower bound on the formula size of an explicit Boolean function.

In this paper, we give a set of functions G each of which has such a composite property, formally for every function of the form $f = f_1 \otimes \cdots \otimes f_k$ with $f_1, \ldots, f_k \in G$, the formula complexity of f is equal to the product of the complexities of f_i's. The class G contains every two-variable functions, 134 out of

256 three-variable functions, 2144 out of 2^{16} four-variable functions, and more. We also give a procedure that recursively generates Boolean formulas whose optimality is guaranteed (Theorem 7).

Our results are essentially based on a careful inspection of a recently proposed complexity measure SumPI [7], which originated from the quantum adversary method [1, 2, 8, 10]. In particular, we mainly use the dual of the semidefinite version of the definition of this measure. This enables us to obtain a lower bound on the size of a formula *constructively*, i.e., a lower bound is obtained by giving a feasible solution to a certain semidefinite program.

The organization of the paper is as follows: In Section 2, we present some basic notations and definitions on Boolean formula complexity. In Section 3, we first introduce two equivalent formulations of the complexity measure SumPI. Then we analyze them to give a set of functions having a composite property as well as a procedure that recursively generates optimal Boolean formulas. In Section 4, we discuss some open problems concerning our work.

2 Preliminaries

A *Boolean formula* is a binary tree where each internal node is labeled with $\wedge$ or $\vee$, and each leaf is labeled with a literal. A Boolean formula computes a Boolean function in an obvious way. The *size* of a formula is the number of leaves in it. For a Boolean function f, the *formula complexity* of f, denoted by $L(f)$, is defined as the size of a smallest formula that computes f. A Boolean formula F is said to be *optimal* if the size of F is equal to $L(f)$ where f is the function computed by F.

For a natural number n, $[n]$ denotes the set $\{1,\ldots,n\}$. For a binary sequence $x \in \{0,1\}^n$, x_i denotes the i-th bit of x. The set of real numbers is denoted by $\mathbf{R}$. For a matrix A, $A[x,y]$ denotes its (x,y) element. Let $\mathrm{tr}(A)$ be the *trace* of A, i.e., $\mathrm{tr}(A) = \sum_x A[x,x]$. For a diagonal matrix A, $A[x,x]$ is simply denoted by $A[x]$. Let $A \cdot B$ denote the scalar product of A and B, i.e., $A \cdot B = \sum_{x,y} A[x,y]B[x,y]$, and $A \circ B$ denote the Hadamard product of A and B, i.e., $(A \circ B)[x,y] = A[x,y]B[x,y]$. Let $A \geq B$ denote the componentwise comparison, formally $A[x,y] \geq B[x,y]$ for every x and y. Let $A \succeq B$ denote that $A - B$ is positive semidefinite, formally for every real vector v, $v^T(A-B)v \geq 0$, or equivalently, all eigenvalues of $(A-B)$ are non-negative.

Let Parity_n denote the parity function on n variables, i.e., $\mathsf{Parity}_n(x_1,\ldots,x_n) = \sum_i x_i \bmod 2$, and let Maj_n denote the majority function on n variables, i.e., $\mathsf{Maj}_n(x_1,\ldots,x_n) = 1$ iff $\sum_i x_i \geq \lceil n/2 \rceil$. For $S \subseteq [n]$, $\mathsf{Eq}_{n,S}$ denotes the function on n variables such that $\mathsf{Eq}_{n,S}(x_1,\ldots,x_n) = 1$ iff $\sum_i x_i \in S$.

3 Formula Size for Composite Functions

For two Boolean functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$, $f \otimes g$ denotes a Boolean function on nm variables defined as

$$(f \otimes g)(x_1,\ldots,x_{nm}) \;=\; f(g(\tilde{x}_1),\ldots,g(\tilde{x}_n)),$$

where $\tilde{x}_i = (x_{(i-1)m+1},\ldots,x_{im})$. The d-th iterated function

$$\overbrace{f \otimes f \otimes \cdots \otimes f}^{d}$$

is simply denoted by f^d.

As we described in Introduction, an optimal formula for $(\mathsf{Parity}_2)^d \equiv \mathsf{Parity}_{2^d}$ can be obtained by a recursive use of an optimal formula for Parity_2.

Let us now consider Parity_3. A simple examination shows an optimal formula for Parity_3 is

$$(x_1\overline{x}_2 \vee \overline{x}_1 x_2)\overline{x}_3 \vee (x_1 x_2 \vee \overline{x}_1\overline{x}_2)x_3,$$

and $L(\mathsf{Parity}_3) = 10$. By "squaring" this formula, we have a formula for $\mathsf{Parity}_9 = (\mathsf{Parity}_3)^2$ of size $10^2 = 100$. However, this is obviously not optimal. If we construct a formula for Parity_9 by following an expression:

$$\begin{aligned}\mathsf{Parity}_9 &= \mathsf{Parity}_5 \oplus \mathsf{Parity}_4 \\ &= (\mathsf{Parity}_3 \oplus \mathsf{Parity}_2) \oplus (\mathsf{Parity}_2 \oplus \mathsf{Parity}_2),\end{aligned}$$

we can obtain a formula for Parity_9 of size $4\{(10+4)+(4+4)\} = 88$, which is smaller than $100 = L(\mathsf{Parity}_3)^2$. This motivates us to classify functions according to whether it has such a composition property.

Definition 1 *A Boolean function f is said to be good if $L(f^d) = L(f)^d$ for every $d \geq 1$; otherwise, it is called bad.*

By the above discussion, we have known that Parity_2 is good and Parity_3 is bad. Obviously, every read-once formula f is good. In order to learn more, we investigate a recently proposed complexity measure SumPI [7], which originated from the quantum adversary method [1, 2, 8, 10].

For $x \in \{0,1\}^n$, let $p_x : [n] \to \mathbf{R}$ be a probability distribution, that is $p_x(i) \geq 0$ and $\sum_i p_x(i) = 1$. Let $p = \{p_x \mid x \in \{0,1\}^n\}$. Then

$$\mathsf{SumPI}(f) \quad = \quad \min_{p} \max_{\substack{x,y \\ f(x)\neq f(y)}} \frac{1}{\sum_{i:x_i\neq y_i} \sqrt{p_x(i)p_y(i)}}. \tag{1}$$

Laplante, Lee and Szegedy [7] proved that $\mathsf{SumPI}^2(f)$ lower bounds the formula complexity of f, i.e., $L(f) \geq \mathsf{SumPI}^2(f)$ for every f. Several equivalent definitions of SumPI are known [9]. We show here the semidefinite version of the definition of SumPI, which is the most suitable form for our purpose.

Theorem 1 *[9] Let F be a $2^n \times 2^n$ binary matrix such that $F[x,y] = 1$ iff $f(x) \neq f(y)$, and let D_i be a $2^n \times 2^n$ binary matrix such that $D_i[x,y] = 1$ iff $x_i \neq y_i$ and $f(x) \neq f(y)$.[1] Let μ_{min} be the minimal solution of the following semidefinite program:*

$$\begin{array}{lrcl} \textit{minimize} & \mu = tr(\Delta) & & \\ \textit{subject to} & \Delta & \textit{is} & \textit{diagonal} \\ & Z & \geq & 0 \\ & Z \cdot F & = & 1 \\ & \forall i : \Delta - Z \circ D_i & \succeq & 0. \end{array} \tag{2}$$

Then $\mathsf{SumPI}(f) = 1/\mu_{min}$.

Note that the above is in fact the *dual* of the semidefinite formulation of the minimax version of $\mathsf{SumPI}(f)$ in Eq. (1). The merit of using this formulation is that we can prove a lower bound constructively. Every feasible solution (Δ, Z) of SDP (2) gives a lower bound of $1/\mathrm{tr}(\Delta)$ on $\mathsf{SumPI}(f)$.

The measure $\mathsf{SumPI}(f)$ has a strong composition property. Ambainis proved that $\mathsf{SumPI}(f)^d \geq \mathsf{SumPI}(f^d)$ [1]. Laplante, Lee and Szegedy [7] proved that this lower bound is tight by showing $\mathsf{SumPI}(f)^d \leq \mathsf{SumPI}(f^d)$. By a careful inspection of their proof, it is not hard to generalize this to the following form, which was also stated in [4].

Theorem 2 *[1, 7] For every Boolean functions f and g,*

$$\mathsf{SumPI}(f \otimes g) \quad = \quad \mathsf{SumPI}(f) \cdot \mathsf{SumPI}(g). \qquad \triangleleft$$

The above theorem implies that if $\mathsf{SumPI}^2(f) = L(f)$ and $\mathsf{SumPI}^2(g) = L(g)$, then a formula for $f \otimes g$ of size $L(f) \cdot L(g)$ is guaranteed to be optimal.

In what follows, a Boolean function f is said to be *tight* if $\mathsf{SumPI}^2(f) = L(f)$. By the definition, every tight function is also good. Moreover, if G is a set of tight functions, then $L(f_1 \otimes \cdots \otimes f_k) = L(f_1) \cdots L(f_k)$ for every $f_1, \ldots, f_k \in G$. We do not know whether the converse is true or not.

Let B_n be the set of all 2^{2^n} Boolean functions on n variables. The set B_n can be divided into equivalence classes, which usually called NPN-equivalence classes, by considering the following three operations: (i) input inversion, (ii) input permutation and (iii) output inversion. Two Boolean functions are NPN-equivalent if one can be transformed into the other by permuting and/or negating the inputs and/or negating the output. Let $F_n \subseteq B_n$ be a set of representatives of all NPN-equivalence classes. An easy computation shows that

$$\begin{aligned} F_1 &= \{0, x_1\}, \\ F_2 &= F_1 \cup \{x_1 x_2, x_1 \oplus x_2\}, \end{aligned}$$

[1] We added the condition that $f(x) \neq f(y)$ here to the definition of D_i, which is not exist in [9]. However, it is easy to see that these two definitions are equivalent.

$$F_3 = F_2 \cup \{x_1x_2x_3, (x_1 \oplus x_2)x_3, x_1x_2 \vee x_3, x_1x_2 \vee \overline{x_1}x_3, \mathsf{Maj}_3, \mathsf{Parity}_3, \mathsf{Eq}_{3,\{2\}}, \mathsf{Eq}_{3,\{0,3\}}, x_1x_2x_3 \vee \overline{x_1}(\overline{x_2} \vee \overline{x_3}), (x_1x_2 \vee x_3)(\overline{x_2} \vee \overline{x_3})\}.$$

Note that the number of NPN-equivalence classes in B_n for $n = 3, 4$ and 5 are 14, 222 and 616126, respectively.

The following fact is obvious but useful.

Fact 1 *Suppose that f and g belong to a same NPN-equivalence class. Then $L(f) = L(g)$ and $\mathsf{SumPI}(f) = \mathsf{SumPI}(g)$.* ◁

It is easy to check that every function in F_2 is tight. The next theorem gives a way to generate a tight function recursively.

Theorem 3 *Suppose that $f(x_1, \ldots, x_n) = x_1 \cdot g(x_2, \ldots, x_n)$. Then $\mathsf{SumPI}^2(f) = \mathsf{SumPI}^2(g) + 1$.*

Proof.(i) $\mathsf{SumPI}^2(f) \leq \mathsf{SumPI}^2(g) + 1$.

We use the minimax version of the definition of $\mathsf{SumPI}(f)$. Let $p = \{p_x \mid x \in \{0,1\}^{n-1}\}$ be a set of probability distributions on $[n-1]$ that attains $\mathsf{SumPI}(g)$ in Eq. (1).

Put $z = \mathsf{SumPI}(g)$. We will define a set of probability distributions $q = \{q_x \mid x \in \{0,1\}^n\}$ on $[n]$ such that the RHS in Eq. (1) is at most $\sqrt{z^2+1}$. Note that $f(0x) = f(0y) = f(1x) = 0$ and $f(1y) = 1$ for every $x \in g^{-1}(0)$ and every $y \in g^{-1}(1)$. Define q as

$$\begin{aligned} q_{1y} &= \left(\frac{1}{z^2+1}, \frac{z^2 p_y(1)}{z^2+1}, \ldots, \frac{z^2 p_y(n-1)}{z^2+1} \right), && \text{for } y \in g^{-1}(1), \\ q_{0x} = q_{1x} &= (0, p_x(1), \ldots, p_x(n-1)), && \text{for } x \in g^{-1}(0), \\ q_{0y} &= (1, 0, \ldots, 0), && \text{for } y \in g^{-1}(1). \end{aligned}$$

It is easy to check that

$$\max_{\substack{u,v \\ f(u) \neq f(v)}} \frac{1}{\sum_{i:u_i \neq v_i} \sqrt{q_u(i) q_v(i)}} \leq \sqrt{z^2+1}.$$

For example, if $(u, v) = (1y, 0x)$ where $y \in g^{-1}(1)$ and $x \in g^{-1}(0)$, then

$$\frac{1}{\sum_{i:u_i \neq v_i} \sqrt{q_u(i) q_v(i)}} = \frac{\sqrt{z^2+1}}{z} \frac{1}{\sum_{i:x_i \neq y_i} \sqrt{p_x(i) p_y(i)}} \leq \sqrt{x^2+1}$$

since $\mathsf{SumPI}(g) = z$. The other cases are similar and omitted.

(ii) $\mathsf{SumPI}^2(f) \geq \mathsf{SumPI}^2(g) + 1$.

To prove the other direction, we use the semidefinite version of the definition of $\mathsf{SumPI}(f)$ described in Theorem 1.

Let (Δ_g, Z_g) be a solution of (2) for the function g, and let μ_g be the minimal value of the objective function, i.e., $\mu_g = \mathrm{tr}(\Delta_g)$. Let F_g and $(D_g)_i$ be matrices defined in Theorem 1. Without loss of generality, we can assume that all diagonal elements of Δ_g is non-negative and $\sum_{x \in g^{-1}(0)} \Delta_g[x] = \sum_{y \in g^{-1}(1)} \Delta_g[y] = \mu_g/2$ (whose proof is omitted in this version due to the space constraint.) Since $\mathsf{SumPI}(g) = 1/\mu_g$, all we have to show is the minimal solution of SDP (2) for the function f is at most $\mu_g/\sqrt{\mu_g+1}$.

In order to show this, we give two matrices Δ_f and Z_f that satisfy all conditions in (2) and $\mathrm{tr}(\Delta_f) = \mu_g/\sqrt{\mu_g+1}$. Define a $2^n \times 2^n$ diagonal matrix Δ_f as:

$$\begin{aligned} \Delta_f[0x] &= 0, && \text{for } x \in g^{-1}(0), \\ \Delta_f[0y] &= \frac{\mu_g^2 \Delta_g[y]}{(\mu_g^2+1)^{3/2}}, && \text{for } y \in g^{-1}(1), \\ \Delta_f[1x] &= \frac{\Delta_g[x]}{(\mu_g^2+1)^{3/2}}, && \text{for } x \in g^{-1}(0), \\ \Delta_f[1y] &= \frac{\Delta_g[y]}{(\mu_g^2+1)^{1/2}}, && \text{for } y \in g^{-1}(1). \end{aligned}$$

We also define a $2^n \times 2^n$ matrix Z_f as:

$$Z_f = \begin{pmatrix} 0 & Z_f^1 \\ Z_f^1 & \frac{1}{\mu_g^2+1} Z_g \end{pmatrix},$$

where Z_f^1 is a $2^{n-1} \times 2^{n-1}$ diagonal matrix such that

$$\begin{aligned} Z_f^1[x,x] &= 0 && \text{for } x \in g^{-1}(0), \\ Z_f^1[y,y] &= \frac{\mu_g}{\mu_g^2+1}\Delta_g[y] && \text{for } y \in g^{-1}(1). \end{aligned}$$

Then the value of the objective function, which is equal to $\mathrm{tr}(\Delta_f)$, is given by

$$\begin{aligned} \mathrm{tr}(\Delta_f) &= \sum_{x \in g^{-1}(0)} \frac{\Delta_g[x]}{(\mu_g^2+1)^{3/2}} \\ &\quad + \sum_{y \in g^{-1}(1)} \Delta_g[y] \left(\frac{\mu_g^2}{(\mu_g^2+1)^{3/2}} + \frac{1}{(\mu_g^2+1)^{1/2}} \right) \\ &= \frac{\mu_g}{2} \left(\frac{\mu_g^2+1}{(\mu_g^2+1)^{3/2}} + \frac{1}{(\mu_g^2+1)^{1/2}} \right) = \frac{\mu_g}{(\mu_g^2+1)^{1/2}}. \end{aligned}$$

The condition $Z_f \geq 0$ is trivially hold. We have

$$\begin{aligned} Z_f \cdot F_f &= \frac{1}{\mu_g^2+1}(Z_g \cdot F_g) + 2 \sum_{y \in g^{-1}(1)} Z_f[0y, 1y] \\ &= \frac{1}{\mu_g^2+1} + 2\frac{\mu_g}{\mu_g^2+1} \sum_{y \in g^{-1}(1)} \Delta_g[y] = \frac{1}{\mu_g^2+1} + \frac{\mu_g^2}{\mu_g^2+1} = 1. \end{aligned}$$

The second last equality follows from $\sum_{y \in g^{-1}(1)} \Delta_g[y] = \mu_g/2$.

We now show that $\Delta_f - Z_f \circ (D_f)_i$ is positive semidefinite for every $i = 1, \ldots, n$, where $(D_f)_i$ is a $2^n \times 2^n$ binary matrix whose $[x,y]$ entry is 1 iff $x_i \neq y_i$ and $f(x) \neq f(y)$. We divide this into two cases. Case i) $i = 1$.

The matrix $(D_f)_1$ is of the form

$$(D_f)_1 = \begin{pmatrix} \mathbf{0} & A \\ A & \mathbf{0} \end{pmatrix},$$

where $\mathbf{0}$ is a $2^{n-1} \times 2^{n-1}$ matrix whose elements are all 0, and A is a $2^{n-1} \times 2^{n-1}$ matrix satisfying that $A[y,y] = 1$ for every $y \in g^{-1}(1)$ (since $1 = f(1y) \neq (0y) = 0$). Let S be a submatrix of $\Delta_f - Z_f \circ (D_f)_i$ consisting of the rows and columns indexed by $0y$ and $1y$ for $y \in g^{-1}(1)$. Then S is a $2|g^{-1}(1)| \times 2|g^{-1}(1)|$ matrix given by

$$\begin{aligned} S[0y,0y] &= \frac{\mu_g^2 \Delta_g[y]}{(\mu_g^2+1)^{3/2}}, \\ S[0y,1y] &= S[1y,0y] = -\frac{\mu_g \Delta_g[y]}{\mu_g^2+1}, \\ S[1y,1y] &= \frac{\Delta_g[y]}{(\mu_g^2+1)^{1/2}}. \end{aligned}$$

Since $(\Delta_f - Z_f \circ (D_f)_i)[vx, vx] \geq 0$ for every $v \in \{0,1\}$ and $x \in g^{-1}(0)$, it is sufficient to show that S is positive semidefinite, i.e., for every $w = (w_{0y}, w_{1y})_{y \in g^{-1}(1)} \in \mathbf{R}^{2|g^{-1}(1)|}$, $w^T S w \geq 0$. This can be verified as follows:

$$\begin{aligned} &\sum_y \frac{\mu_g^2 \Delta_g[y]}{(\mu_g^2+1)^{3/2}} w_{0y}^2 + \sum_y \frac{\Delta_g[y]}{(\mu_g^2+1)^{1/2}} w_{1y}^2 - 2\sum_y \frac{\mu_g \Delta_g[y]}{(\mu_g^2+1)} w_{0y} w_{1y} \\ &= \sum_y \left(\sqrt{\frac{\mu^2 \Delta_g[y]}{(\mu_g^2+1)^{3/2}}} w_{0y} - \sqrt{\frac{\Delta_g[y]}{(\mu_g^2+1)^{1/2}}} w_{1y} \right)^2 \geq 0. \end{aligned}$$

Case ii) $i = 2, \ldots, n$.

We show the case $i = 2$. The other cases are analogous to this case. The matrix $(D_f)_2$ is of the form

$$(D_f)_2 \;=\; \begin{pmatrix} 0 & B \\ B & (D_g)_1 \end{pmatrix},$$

where B is a $2^{n-1} \times 2^{n-1}$ matrix whose diagonal elements are all 0. Note that all diagonal elements of $(D_g)_1$ are 0. Hence $\Delta_f - Z \circ (D_f)_2$ has non-zero elements only at $[0u, 0u]$ for $u \in g^{-1}(1)$ and at $[1u, 1v]$ for $u, v \in \{0,1\}^{n-1}$. Since $\Delta_f[0u]$ is non-negative for every u, $\Delta_f - Z \circ (D_f)_2$ is positive semidefinite if $\Delta_f^1 - Z \circ (D_g)_1$ is positive semidefinite where Δ_f^1 denotes a submatrix of Δ_f consisting of rows and columns indexed by $1u$ for $u \in \{0,1\}^{n-1}$. Let $w = (w_u)_{u \in \{0,1\}^{n-1}} \in \mathbf{R}^{2^{n-1}}$. Then $w^T(\Delta_f^1 - Z \circ (D_g)_1)w$ is given by

$$\sum_{x \in g^{-1}(0)} \frac{\Delta_g[x]}{(\mu_g^2+1)^{3/2}} w_x^2 \;+\; \sum_{y \in g^{-1}(1)} \frac{\Delta_g[y]}{(\mu_g^2+1)^{1/2}} w_y^2 - 2 \sum_{\substack{x \in g^{-1}(0) \\ y \in g^{-1}(1)}} \frac{(Z_g \circ (D_g)_1)[x,y]}{\mu_g^2+1} w_x w_y. \qquad (3)$$

Let

$$\frac{1}{(\mu_g^2+1)^{3/2}} w_x^2 = \tilde{w_x}^2, \quad \text{and} \quad \frac{1}{(\mu_g^2+1)^{1/2}} w_y^2 = \tilde{w_y}^2,$$

or equivalently,

$$w_x = \sqrt{(\mu_g^2+1)^{3/2}}\tilde{w_x}, \quad \text{and} \quad w_y = \sqrt{(\mu_g^2+1)^{1/2}}\tilde{w_y}.$$

Putting these into Eq. (3) gives

$$\sum_{x \in g^{-1}(0)} \Delta_g[x]\tilde{w_x}^2 + \sum_{y \in g^{-1}(1)} \Delta_g[y]\tilde{w_y}^2 - 2 \sum_{\substack{x \in g^{-1}(0) \\ y \in g^{-1}(1)}} (Z_g \circ (D_g)_1)[x,y]\tilde{w_x}\tilde{w_y}.$$

This is always non-negative since $\Delta_g - Z_g \circ (D_g)_1$ is positive semidefinite. This completes the proof of Theorem 3. ◁

An analogous proof can be carried for the functions $x \wedge \overline{g}$, $x \vee g$ and $x \vee \overline{g}$. Since $L(f) = L(g) + 1$, we have:

Corollary 1 *Suppose that g is a tight function and x is a variable not appearing in g. Then all of $x \wedge g$, $x \wedge \overline{g}$, $x \vee g$ and $x \vee \overline{g}$ are tight.* ◁

Combining Corollary 1 with the fact that every function in B_2 is tight, we find that $x_1x_2x_3$, $(x_1 \oplus x_2)x_3$ and $x_1x_2 \vee x_3$ in F_3 are also tight. Interestingly, there are only one more tight function in F_3.

Theorem 4 *The function $x_1x_2 \vee \overline{x_1}x_3 \in B_3$ is tight.*

Proof. Let $f = x_1x_2 \vee \overline{x_1}x_3$. It is obvious that $L(f) = 4$. By Theorem 1, it is sufficient to give a feasible solution (Δ, Z) of SDP (2) such that $\mathrm{tr}(\Delta) = 1/2$.

Define a diagonal matrix Δ such that $\Delta[010] = \Delta[110] = \Delta[001] = \Delta[101] = 1/8$ and all other elements are 0, and define Z such that $Z[x,y] = Z[y,x] = 1/8$ if

$$(x,y) \;\in\; \{(010,110), (001,101), (101,110), (010,001)\},$$

and all other elements are 0. A simple calculation shows that these matrices satisfy all conditions in Eq. (2). ◁

Note that every other function in F_3 is not tight. Here's the table of $L(f)$ and $\mathsf{SumPI}^2(f)$ for $f \in F_3$ with $\mathsf{SumPI}^2(f) \neq L(f)$.

f	$L(f)$	$\mathsf{SumPI}^2(f)$
Maj_3	5	4
Parity_3	10	9
$\mathsf{Eq}_{3,\{2\}}$	8	7
$\mathsf{Eq}_{3,\{0,3\}}$	6	4.5
$x_1x_2x_3 \vee \overline{x_1}(\overline{x_2} \vee \overline{x_3})$	6	$3+2\sqrt{2}$
$(x_1x_2 \vee x_3)(\overline{x_2} \vee \overline{x_3})$	5	$3+\sqrt{3}$

Theorem 5 *A function f on 3 variables is tight iff f is an NPN-equivalent to a function in $T_3 = F_2 \cup \{x_1x_2x_3, (x_1 \oplus x_2)x_3, x_1x_2 \vee x_3, x_1x_2 \vee \overline{x}_1x_3\}$.* ◁

Note that the number of tight classes in B_3 is 8 and the total number of tight functions in B_3 is 134. Now we proceed to consider base functions on 4 variables. Let T_4 be a set of representatives of tight functions on 4 variables. The following corollary is straightforward from Theorem 2.

Corollary 2 *Let g and h be two tight functions. Then $g \otimes h$ is tight.* ◁

Obviously, T_4 contains T_3. By applying Corollary 1 to each tight function on 3 variables, we find that the following 7 classes are also in T_4:

$$x_1x_2x_3x_4, x_1x_2x_3 \vee x_4, (x_1 \oplus x_2)x_3x_4, ((x_1 \oplus x_2)x_3) \vee x_4,$$
$$(x_1x_2 \vee x_3)x_4, (x_1x_2 \vee x_3) \vee x_4, (x_1x_2 \vee \overline{x_1}x_3)x_4.$$

Note that the function $(x_1x_2 \vee \overline{x_1}x_3) \vee x_4$ is NPN-equivalent to the last function in the above. In addition, by applying Corollary 2 to two tight functions on 2 variables, we find that the following 4 classes are also in T_4:

$$x_1x_2 \oplus x_3x_4, x_1x_2 \vee x_3x_4, (x_1 \oplus x_2)(x_3 \oplus x_4), \mathsf{Parity}_4$$

Again, we have only one more:

Theorem 6 *The function $(x_1 \vee x_2)(x_3 \oplus x_4) \in B_4$ is tight.*

Proof. Let $f = (x_1 \vee x_2)(x_3 \oplus x_4)$. It is easy to check that $L(f) = 6$. As for the proof of Theorem 5, we give a feasible solution (Δ, Z) for SDP (2) such that $\mathrm{tr}(\Delta) = 1/\sqrt{6}$. Let Δ be a diagonal matrix such that

$$\Delta[0100] = \Delta[0111] = \Delta[1000] = \Delta[1011] = \Delta[0001] = \Delta[0010] = \frac{1}{12\sqrt{6}},$$
$$\Delta[1010] = \Delta[0110] = \Delta[1001] = \Delta[0101] = \frac{1}{8\sqrt{6}},$$

and all other elements are 0. Let Z be a matrix such that $Z[x,y] = Z[y,x] = 1/24$ if

$$\begin{aligned}(x,y) \in \{&(0100,0101),(0100,0110),(0101,0111),(0110,0111),\\ &(1000,1001),(1000,1010),(1001,1011),(1010,1011),\\ &(0001,0101),(0001,1001),(0010,0110),(0010,1010)\},\end{aligned}$$

and all other elements are 0. An easy but tedious calculation shows that (Δ, Z) satisfies all conditions in Eq. (2). ◁

In fact, we computed the values of $\mathsf{SumPI}(f)$ for all functions on 4 variables by using a SDP solver program. The results confirm that there are 20 tight classes (out of 222) in B_4, all of them have been addressed. The total number of tight functions in B_4 is 2144 (out of 2^{16}).

By applying Corollaries 1 and 2 recursively, we can easily obtain a set of tight functions on $n \geq 5$ variables. We do not know whether there is a tight function that cannot be generated by applying these corollaries at the time of writing.

In conclusion, our results can be restated as follows:

Theorem 7 *Let $\mathcal{G}$ be a class of Boolean formulas recursively defined as follows:*

1. *(Basis) $\{x_1, x_1\overline{x_2} \vee \overline{x_1}x_2, x_1x_2 \vee x_1\overline{x_3}, (x_1 \vee x_2)((x_3\overline{x_4}) \vee (\overline{x_3}x_4))\} \subseteq \mathcal{G}$.*
2. *(NPN-equivalence) if $F(x_1, \ldots, x_n) \in \mathcal{G}$, then*
   (a) *(Input Negation) a formula obtained from F by replacing all occurrences of x_i by $\overline{x_i}$ for some $i \in [n]$ is in $\mathcal{G}$,*
   (b) *(Input Permutation) $F(x_{\rho(1)}, \ldots, x_{\rho(n)})$ for some permutation ρ on $[n]$ is in $\mathcal{G}$,*
   (c) *(Output Negation) a formula obtained from F by interchanging $\vee$ and $\wedge$ and replacing all occurrences of x_i by $\overline{x_i}$ for every $i \in [n]$ is in $\mathcal{G}$.*

3. *(Extension) if $F \in \mathcal{G}$ and x is a variable not appearing in F, then $x \vee F, x \wedge F \in \mathcal{G}$.*

4. *(Composition) if $F(x_1, \ldots, x_n), G(x_1, \ldots, x_m) \in \mathcal{G}$, then*

$$F(G(x_1, \ldots, x_m), G(x_{m+1}, \ldots, x_{2m}), \ldots, G(x_{(n-1)m+1}, \ldots, x_{nm})) \in \mathcal{G}.$$

Then every formula in $\mathcal{G}$ is optimal. ◁

4 Discussions

Many open interesting problems remain to be resolved. To extend the class $\mathcal{G}$ in Theorem 7 is apparently one of the most interesting future works. For three-variable functions, we have shown that 8 classes are good and one class is bad, and there remains 5 unknown classes. We have not known whether there is a good function f such that $\mathsf{SumPI}^2(f) \neq L(f)$.

The iterated function of Parity_3, which was discussed in Section 3, shows a considerable gap between $L(f^d)$ and $L(f)^d$. We have

$$\frac{L(\mathsf{Parity}_3)^d}{L((\mathsf{Parity}_3)^d)} = \frac{L(\mathsf{Parity}_3)^d}{L(\mathsf{Parity}_{3^d})} = \Omega\left(\frac{n^{\log_3 10}}{n^2}\right) = \Omega(n^{0.095}),$$

where $n = 3^d$ is the total number of input variables. The problem to find a base function that admits a larger gap also seems to be interesting.

By writing a simple computer program, we can check that $L(f) \leq n^2$ for every function f on $n \leq 4$ variables except for Parity_3 and $\neg\mathsf{Parity}_3$ (in fact $L(f) \leq 8$ for every f on 3 variables except for $f \in \{\mathsf{Parity}_3, \neg\mathsf{Parity}_3\}$, and $L(f) \leq 16$ for every f on 4 variables). This implies that, for every f on $n \leq 4$ variables, the formula complexity of the iterated function of f is at most quadratic in the total number of input variables. So, for example, if we want to obtain a super-quadratic lower bound for composite functions, we need a base function depending on at least 5 variables.

Acknowledgements

The author would like to thank Eiji Takimoto and Hideaki Fukuhara for enjoyable discussions. This work was partially supported by Grant-in-Aid for Scientific Research on Priority Areas "New Horizons in Computing" from MEXT of Japan.

References

[1] A. Ambainis, Polynomial Degree vs. Quantum Query Complexity, *Proc. 44th FOCS*, 230–239, 2003.

[2] H. Barnum, M. Saks, and M. Szegedy, Quantum Decision Trees and Semidefinite Programming, *Proc. 18th CCC*, 179–193, 2003.

[3] J. Håstad, The Shrinkage Exponent of de Morgan Formulas is 2, *SIAM J. Comput.*, 27(1), 48-64, 1998.

[4] P. Høyer and R. Špalek, Tight Adversary Bounds for Composite Functions, quant-ph/0509067, 2005.

[5] K. Iwama and H. Morizumi, An Explicit Lower Bound of $5n-o(n)$ for Boolean Circuits, *Proc. 27th MFCS*, 353–364, 2002.

[6] V.M. Khrapchenko, Complexity of the Realization of a Linear Function in the Case of Π-Circuits, *Math. Notes Acad. Sci.*, 9, 21–23, 1971.

[7] S. Laplante, T. Lee and M. Szegedy, The Quantum Adversary Method and Classical Formula Lower Bounds, *Proc. 20th CCC*, 76–90, 2005.

[8] S. Laplante and F. Magniez, Lower Bounds for Randomized and Quantum Query Complexity using Kolmogorov Arguments, *Proc. 19th CCC*, 294–304, 2004

[9] R. Špalek and M. Szegedy, All Quantum Adversary Methods are Equivalent, *Proc. 32nd ICALP*, 1299–1311, 2005.

[10] S. Zhang, On the Power of Ambainis's Lower Bounds, *Proc. 31st ICALP*, 1238–1250, 2004.