Which bases admit non-trivial shrinkage of formulae? *

Hana Chockler † *Uri Zwick* ‡

June 15, 2000
Revised: May 23, 2001

Abstract

We show that the shrinkage exponent, under random restrictions, of formulae over a finite complete basis B of Boolean functions is strictly greater than 1 if and only if all the functions in B are *unate*, i.e., monotone increasing or monotone decreasing in each one of their variables. As a consequence, we get non-linear lower bounds on the formula complexity of the parity function over any basis composed only of unate functions.

1 Introduction

Let B be a finite collection of Boolean functions. We say that the *shrinkage exponent* of B-formulae, i.e., formulae over the basis B, is $\Gamma = \Gamma_B$, if and only if B-formulae shrink by an expected factor of at least ϵ^Γ when hit by a random restriction that leaves only a fraction ϵ of the variables unassigned, and if Γ is the largest constant for which this statement holds.

Shrinkage of de Morgan formulae, i.e., Boolean formulae over the basis $\{\vee, \wedge, \neg\}$, under random restrictions is a well studied subject. Subbotovskaya [Sub61] was the first to observe that every de Morgan formula shrinks by an expected factor of at least $\epsilon^{1.5}$, when hit by a random assignment leaving only a fraction ϵ of its variables unassigned, and hence $\Gamma \geq 1.5$. Slightly better bounds on Γ were then obtained by Nisan and Impagliazzo [NI93] and by Paterson and Zwick [PZ93b], before Håstad [Hås98] obtained the tight result $\Gamma = 2 - o(1)$.

Andreev [And87b] obtained an $\Omega(n^{\Gamma+1-o(1)})$ lower bound on the de Morgan formula size of a simple function. Håstad's result implies, therefore, an $\Omega(n^{3-o(1)})$ lower bound on the de Morgan formula size of the Andreev function. This is the largest formula size lower bound known, over a complete basis, for any explicit sequence of P or even NP functions.

Let B_2 be the basis that contains all Boolean functions of two variables. It is easy to see that $\Gamma_{B_2} = 1$. This follows from the fact that the formula $x_1 \oplus x_2 \oplus \ldots \oplus x_n$, for example, shrinks only by a factor of ϵ when hit by a random assignment leaving only a fraction ϵ of its variables unassigned. The same holds, of course, for any basis that contains the XOR gate itself, or an XOR gate in disguise.

---

*Preliminary versions of some of the results reported in this paper appeared in the M.Sc. thesis of the first author [Kva96], and, under a slightly different title, in the proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA'01), Washington D.C., 2001, pages 702–708.

†School of Computer Science and Engineering, The Hebrew University, Givat Ram, Jerusalem 91904, Israel, E-mail: `hanac@cs.huji.ac.il`.

‡School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail address: `zwick@post.tau.ac.il`.

Given a finite *complete* basis B, how can we tell whether $\Gamma_B > 1$, i.e., whether B-formula exhibit non-trivial shrinkage under restriction? We provide here the following very simple answer: $\Gamma_B > 1$ if and only if all the functions of the basis B are *unate*, i.e., monotone increasing or decreasing in each one of their variables.

The AND and OR functions $x \vee y$ and $x \wedge y$ are unate as they are monotone increasing in their two variables. The negation function $\neg x$ is unate as it is monotone decreasing. Thus, de Morgan formulae should shrink, as indeed we know to be the case. The 3-bit majority function $m_3(x, y, z)$ is unate as it is increasing in its three variables. Thus, formulae over the basis $\{m_3, \vee, \wedge, \neg\}$ should also shrink. On the other hand, the XOR function $x \oplus y$, or the symmetric function $s_{12}(x, y, z)$ that outputs 1 if and only if $x + y + z$ is 1 or 2, or the selection function $sel(x, y, z) = \bar{x}y \vee xz$, are not unate. So, according to our characterization, formulae over bases that contain these functions do not necessarily exhibit not trivial shrinkage. (Note that $sel(x, y, z)$ is monotone increasing in y and z but is not monotone in x.)

In the process of obtaining the above shrinkage characterization we also obtain the following result which is perhaps of independent interest: The formula complexity of the *parity* function $P_n(x_1, x_2, \ldots, x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ over the basis B is *non-linear* if and only if all the function of the basis B are unate.

To obtain the these results we revive the *weighting* technique of [NI93] (see also [PZ93b]).

The rest of the paper is organized as follows. In the next section, we formally define the notions used in the above discussion. In Section 3 we describe the construction of linear size formulae for the parity function over bases that contain a non-unate function. In Section 4 we show that the shrinkage exponents of unate bases are all strictly greater than 1. In Section 5 we look more closely at the basis U_3 composed of all the unate functions of at most three variables. We end in Section 6 with some open problems and some concluding remarks.

2 Definitions

We begin by formally defining the notion of monotonicity and unateness used above.

Definition 2.1 (Monotonicity and Unateness) *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function of n Boolean variables. We say that f is* monotone increasing *in its i-th variable, if and only if*

$$f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \leq f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) ,$$

for every $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \in \{0,1\}$. We say that f is monotone decreasing *in its i-th variable, if and only if*

$$f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \geq f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) ,$$

for every $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \in \{0,1\}$. We say that f is a unate *function if it is monotone increasing or monotone decreasing in each one of its variables. We say that f is* monotone *if it is monotone increasing in each one of its variables.*

Note that if $f(x_1, \ldots, x_n)$ is a unate function, then there are constants $a_1, \ldots, a_n \in \{0,1\}$ such that $f(x_1 \oplus a_1, \ldots, x_n \oplus a_n)$ is a monotone function.

We next define formally the notion of *shrinkage exponents.*

Definition 2.2 (Shrinkage exponent) *Let B be a basis. If f is a Boolean function that can be computed over B, we let $L_B(f)$, the* formula complexity *of f over B, be the size of the smallest B-formula that computes f. The size of a formula is defined to be the number of leaves in it. For every $\epsilon > 0$, we* *let f^ϵ be the random restriction of f obtained by independently assigning each variable the value 0 with probability $\frac{1-\epsilon}{2}$, the value 1 with probability $\frac{1-\epsilon}{2}$, and leaving it unassigned with probability ϵ. The shrinkage exponent Γ_B of B-formulae is defined to be the least upper bound of all constants γ such that there exists a constant c such that for every function f and every $\epsilon > 0$ we have*

$$E[L_B(f^\epsilon)] \;\leq\; c(\epsilon^\gamma L_B(f) + 1) \; .$$

Let $P_n(x_1, x_2, \ldots, x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ be the n-bit parity function. Andreev [And87a] defines a simple explicit function $A_n(x_1, x_2, \ldots, x_n)$. Most of the interest in shrinkage exponents stems from the following two results (cf. [Sub61],[And87a],[NI93],[PZ93b],[Hås98]):

Theorem 2.3 $L_B(P_n) = \Omega(n^{\Gamma_B})$.

Theorem 2.4 $L_B(A_n) = \Omega(n^{\Gamma_B + 1 - o(1)})$.

3 Parity formulae over non-unate bases

Let B be a finite *complete* basis, i.e., a basis using which all Boolean functions could be implemented. Suppose that $f \in B$, where $f : \{0,1\}^k \to \{0,1\}$, and that f is not unate. Then, there exists $1 \leq i \leq k$ such that f is not monotone in its i-th variable. This means that there are constants $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k \in \{0,1\}$ such that

$$f(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_k) \;=\; x \; ,$$

and constants $b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_k \in \{0,1\}$ such that

$$f(b_1, \ldots, b_{i-1}, x, b_{i+1}, \ldots, b_k) \;=\; \bar{x} \; .$$

Let $c_j = a_j \oplus b_j$, for $1 \leq j \leq k$ and $j \neq i$. Then,

$$f((c_1 \wedge y) \oplus a_1, \ldots, (c_{i-1} \wedge y) \oplus a_{i-1}, x, (c_{i+1} \wedge y) \oplus a_{i+1}, \ldots, (c_k \wedge y) \oplus a_k) \;=\; x \oplus y \; .$$

In other words, after a suitable reordering of the variables of f, we get that

$$f(0, \ldots, 0, 1, \ldots, 1, y, \ldots, y, \bar{y}, \ldots, \bar{y}, x) \;=\; x \oplus y \; .$$

We have obtained a formula for $x \oplus y$ in which x appears only *once*. Let $f^*(0, 1, y, \bar{y}, x)$ be a shorthand for $f(0, \ldots, 0, 1, \ldots, 1, y, \ldots, y, \bar{y}, \ldots, \bar{y}, x)$. As the basis B is complete, there are constant size B-formulae that compute the constants 0 and 1 and the negation function. We get therefore the following linear size B-formula for the parity of n bits:

$$x_1 \oplus \ldots \oplus x_n \;=\; f^*(0, 1, x_1, \bar{x}_1, f^*(0, 1, x_2, \bar{x}_2, f^*(0, 1, x_3, \bar{x}_3, f^*(\ldots \; f^*(0, 1, x_{n-1}, \bar{x}_{n-1}, x_n)) \ldots) \; .$$

The number of appearances of each variable x_i in this formula is at most $(k-1)\ell$, where ℓ is the (minimal) number of appearances of the variable y in a B-formula computing $\bar{y}$. As ℓ is finite, we get that the size of the above formula is indeed $O(n)$. As a result we get the following theorem:

Theorem 3.1 *Let B be a finite complete basis that contains a non-unate function. Then, there are B-formulae of size $O(n)$ that compute the parity function $x_1 \oplus x_2 \oplus \cdots \oplus x_n$.*

As any restriction of a parity function is a parity function or its negation, we also obtain the following:

Theorem 3.2 *Let B be a finite complete basis that contains a non-unate function. Then, $\Gamma_B = 1$.*

The above theorem does not hold if the basis B is not complete. Consider the, somewhat contrived, basis $\{sel(x,y,z)\}$, where $sel(x,y,z) = \bar{x}y \vee xz$ is the $2 \to 1$ selector. This basis is clearly not complete, as $sel(0,0,0) = 0$ and $sel(1,1,1) = 1$ (see Post [Pos41] for a characterization of complete bases). The reader may verify, as an exercise, that formulae over this basis do shrink, although $sel(x,y,z)$ is not monotone increasing or decreasing in x. Formulae over the complete basis $\{sel(x,y,z), \neg\}$ do not necessarily shrink, however, as $sel(x,y,\bar{y}) = x \oplus y$.

It is worthwhile noting that linear size parity formulae constructed above also have linear *depth*. It may be interesting to characterize the bases that admit linear size, logarithmic depth parity formulae. (Note that while any polynomial size formula can be converted into an equivalent formula of polynomial size *and* logarithmic depth, it may not be possible to convert a formula of linear size into an equivalent formula of linear size *and* logarithmic depth.)

4 Shrinkage of formulae over unate bases

For any $k \geq 2$, let U_k be the basis composed of all the unate functions of at most k variables. Clearly if B is a basis composed of unate functions, each depending on at most k variables, then $B \subseteq U_k$. To prove our results, it is therefore enough to show that $\Gamma_{U_k} > 1$, for every $k \geq 2$.

Let T be an U_k-formula. We assume that all the gates of T, i.e., the internal nodes of T, have at least two inputs, and that the leaves of T are labelled by literals, i.e., variables or their negation. The *size* $L(T)$ of T is defined to be the number of leaves in T. The *weight* $W(T)$ of the formula T is defined to be $W(T) = L(T) + \sum_{r=2}^{k} \alpha_r \cdot G_r(T)$, where $G_r(T)$ is the number of gates in T with r inputs, and α_r, for $2 \leq r \leq k$, is an increasing sequence of positive weights, to be chosen later. Let $L_r(T)$ be the number of leaves in T that enter gates with r inputs. For brevity, we sometimes omit T from the expressions $L(T), W(T), G_r(T)$ and $L_r(T)$ when the formula T is clear from the context. Let T a formula in the variables $x_1, x_2, \ldots, x_n$. The *savings* obtained by assigning each variable each Boolean value is defined as:

$$S(T) \;=\; \sum_{i=1}^{n} \sum_{b=0}^{1} \left[W(T) - W(T_{x_i \leftarrow b}) \right] ,$$

where $T_{x_i \leftarrow b}$ is the formula obtained from T by assigning the variable x_i the value b and performing some simple simplifications (to be described in more detail below). The following simple lemma is based on Subbotovskaya's ideas [Sub61] and is implicit in [NI93] and [PZ93b] (cf. Theorems 2.1 and 2.5 of [PZ93b]).

Lemma 4.1 *If $W(T)$ is a weight function on B-formulae such that $W(T) = \Theta(L(T))$, and if $S(T)/W(T) \geq 2\Lambda$ for every B-formula T such that $L(T) > 1$, then $\Gamma_B \geq \Lambda$.*

In the sequel, we assume that each U_k-formula T that we deal with satisfies the following three conditions: (i) If a binary gate of T is fed by a leaf labelled by x or $\bar{x}$ and by a subformula T_0, then the variable x does not appear in T_0. (ii) A ternary gate is never fed by two leaves labelled by literals of the same variable. (iii) An r-ary gate, where $r \geq 4$, is not fed by more than two leaves labelled by literals of the same variable. If T does not satisfy one of these conditions than it is easy to obtain an equivalent formula of smaller weight. For example, assume that a ternary gate is fed by both x and $\bar{x}$. Let $f(x_1, x_2, x_3) \in U_3$ be the function computed by this gate. We claim that $f(x, \bar{x}, y)$ is a unate function so that the ternary gate in question can be replaced by a binary gate, thus reducing the weight of the formula. (Note that $\alpha_{r-1} < \alpha_r + 1$, for $3 \leq r \leq k$, as $\alpha_{r-1} \leq \alpha_r$.) Why is $f(x, \bar{x}, y)$ unate? It is a function of two variables and it is monotone (increasing or decreasing) in y. The only non-unate two variable functions, $x \oplus y$ and

$x \oplus y \oplus 1$, are non-monotone in both variables. Thus $f(x, \bar{x}, y)$ must be unate. Similarly, if a gate is fed by two leaves labelled x, or by two leaves labelled $\bar{x}$, then that gate can be replaced by a smaller gate.

We now describe the process used to obtain $T_{x_i \leftarrow b}$. We assume that T satisfies the three conditions of the previous paragraph. Otherwise, we simplify T before assigning x_i the value b. We assign the value b to all the leaves labelled x_i and the value $\bar{b}$ to all the leaves labelled $\bar{x}_i$. If a gate is fed by a leaf or leaves labelled by constants, then this gate is replaced by a smaller equivalent gate without constant inputs. If the output of the gate is constant, then this gate is replaced by a leaf labelled by this constant. This process is repeated until all leaves are labelled by literals and all gates compute non-constant functions, though for our purposes, it would be enough to consider the first iteration of this process.

Recall that α_r, for $2 \le r \le k$, is the weight attached to each r-ary gate. For a reason that will become clear shortly, we let

$$\begin{aligned}
\beta_2 &= 3 + 2\alpha_2 \,, \\
\beta_3 &= 2 + 2(\alpha_3 - \alpha_2) \,, \\
\beta_r &= \min\{\, 2(1 + \alpha_r - \alpha_{r-1}) \,,\, 2 + \alpha_r - \alpha_{r-2} \,\} \quad , \quad \text{for} \quad 4 \le r \le k \,.
\end{aligned}$$

We now obtain a lower bound for the savings $S(T)$ obtained by assigning each variable that appears in T each Boolean value:

Lemma 4.2 *Let T be an U_k-formula. Let L_r, for $2 \le r \le k$, be the number of leaves of T entering r-ary gates. Then, $S(T) \ge \sum_{r=2}^{k} \beta_r \cdot L_r$.*

Proof: Consider the effect of assigning the values 0 and 1 to a single leaf of T labelled by the literal x. If x enters a binary gate, then as in Subbotovskaya's original argument, one of the assignments removes at least two leaves from the formula, while the other assignment removes at least one leaf from the formula. In each case, we have also removed at least one binary gate. The total reduction in the weight is therefore at least $\beta_2 = 3 + 2\alpha_2$. Suppose now that x enters an r-ary gate, where $r > 2$. Under each assignment, the number of leaves in the formula is reduced by one, the number of r-ary gates is reduced by one, and the number of $(r-1)$-ary gates is increased by one. The total reduction in the weight is therefore $2(1 + \alpha_r - \alpha_{r-1})$. What happens if we now assign the values 0 and 1 to *all* the leaves labelled by x and $\bar{x}$? All the savings add up, except for cases in which two leaves labelled by x or $\bar{x}$ feed the same r-ary gate, where $r \ge 4$. The total reduction obtained when assigning the values 0 and 1 to two such leaves is $4 + 2(\alpha_r - \alpha_{r-2})$, and the reduction per leaf is now $2 + \alpha_r - \alpha_{r-2}$. The reduction per each of the L_r leaves that enter r-ary gates is therefore at least $\beta_r = \min\{\, 2(1 + \alpha_r - \alpha_{r-1}) \,,\, 2 + \alpha_r - \alpha_{r-2} \,\}$. Summing this up, we get the claim of the lemma. □

We next define

$$\begin{aligned}
\gamma_r &= 1 + \frac{\alpha_r}{r} + \frac{\alpha_k}{r(k-1)} \quad , \quad \text{for} \quad 2 \le r < k \,, \\
\gamma_k &= 1 + \frac{\alpha_k}{k-1} \,.
\end{aligned}$$

We now claim:

Lemma 4.3 *Assume that $\frac{\alpha_k}{k-1} \ge \frac{\alpha_r}{r-1}$ for every $2 \le r \le k$. Let $\Lambda = \frac{1}{2}\min_{r=2}^{k} \beta_r/\gamma_r$. Then, for every U_k-formula T with $L(T) > 1$, we have $S(T)/W(T) \ge 2\Lambda$.*

Proof: We know, from Lemma 4.2, that $S(T) \ge \sum_{r=2}^{k} \beta_r \cdot L_r$. We now show that $W(T) \le \sum_{r=2}^{k} \gamma_r \cdot L_r$. The claim of the lemma would then follow as

$$\frac{S(T)}{W(T)} \ge \frac{\sum_{r=2}^{k} \beta_r \cdot L_r}{\sum_{r=2}^{k} \gamma_r \cdot L_r} \ge \min_{2 \le r \le k} \frac{\beta_r}{\gamma_r} = 2\Lambda \,.$$

Our goal, therefore, is to obtain an upper bound on $W = W(T)$, given that L_r of the leaves of T enter r-ary gates. Let G_r be the number of r-ary gates in T, for $2 \leq r \leq k$. Recall that $L = \sum_{r=2}^{k} L_r = \sum_{r=2}^{k}(r-1)G_r + 1$, and that $W = L + \sum_{r=2}^{k} \alpha_r \cdot G_r$. As there are L_r leaves that enter r-ary gates, we get that $G_r \geq \frac{1}{r}L_r$. We can thus bound W in the following way:

$$W \leq L + \left[\begin{array}{ll} \max & \sum_{r=2}^{k} \alpha_r \cdot G_r \\ \text{s.t.} & \sum_{r=2}^{k} (r-1)G_r = L-1 \\ & G_r \geq L_r/r \;\;,\;\; 2 \leq r \leq k \end{array} \right] \;,$$

where within the brackets, the G_r's are considered to be the variables of a linear program, and not necessarily the number of gates in a concrete formula. The condition $\frac{\alpha_k}{k-1} \geq \frac{\alpha_r}{r-1}$, for $2 \leq r \leq k$, ensures that the optimal value of the linear program is attained when $G_r = L_r/r$, for $2 \leq r < k$, and $G_k = \frac{1}{k-1}\left(L - \sum_{r=2}^{k-1}(r-1)G_r - 1\right)$. In other words, we would have only the minimum required number of r-ary gates, for $2 \leq r < k$, and as many k-ary gates as possible. Thus

$$G_k = \frac{1}{k-1}\left(L - \sum_{r=2}^{k-1}(r-1)G_r - 1\right) < \frac{1}{k-1}\left(L - \sum_{r=2}^{k-1}\frac{r-1}{r}L_r\right) = \frac{1}{k-1}\left(\sum_{r=2}^{k-1}\frac{L_r}{r} + L_k\right) \;,$$

and as a consequence

$$\begin{aligned} W \;&=\; L + \sum_{r=2}^{k}\alpha_r G_r \;\leq\; L + \sum_{r=2}^{k-1}\frac{\alpha_r}{r}L_r + \frac{\alpha_k}{k-1}\left(\sum_{r=2}^{k-1}\frac{L_r}{r} + L_k\right) \\ &=\; \sum_{r=2}^{k-1}\left(1 + \frac{\alpha_r}{r} + \frac{\alpha_k}{r(k-1)}\right)\cdot L_r + \left(1 + \frac{\alpha_k}{k-1}\right)\cdot L_k \;=\; \sum_{r=2}^{k}\gamma_r \cdot L_r \;. \end{aligned}$$

This completes the proof of the Lemma. □

Theorem 4.4 *The shrinkage exponent Γ_k of U_k-formulae satisfies $\Gamma_k \geq 1 + \frac{1}{3k-4}$.*

Proof: Let $\alpha_r = r-2$, for $2 \leq r \leq k$. Clearly, $\frac{\alpha_r}{r-1}$ is an increasing sequence so the condition $\frac{\alpha_k}{k-1} \geq \frac{\alpha_r}{r-1}$, for $2 \leq r \leq k$, is satisfied. With these values of α_r we get $\beta_2 = 3$ and $\beta_r = 4$, for $3 \leq r \leq k$. Next, we get that $\gamma_r = 1 + \frac{r-2}{r} + \frac{k-2}{r(k-1)}$, for $2 \leq r < k$, and $\gamma_k = 1 + \frac{k-2}{k-1}$. It is not difficult to check that the minimum of β_r/γ_r is attained by $\beta_2/\gamma_2 = 2(1 + \frac{1}{3k-4})$. The theorem thus follows from Lemma 4.1 and Lemma 4.3. □

The choice $\alpha_r = r-2$ is perhaps the simplest choice of the weights. It is not the optimal choice however. As $L = \sum_{r=2}^{k}(r-1)G_r + 1$ is a linear combination of G_r, for $2 \leq r \leq k$, we can always take $\alpha_2 = 0$. When $k = 3$, the optimal choice of α_3 is $\alpha_3 = \sqrt{3} - 1$ which gives $\Gamma_3 \geq 3 - \sqrt{3} \simeq 1.26795$. When $k = 4$, an optimal choices for α_3 and α_4 are $\alpha_3 = \sqrt{7} - 2$ and $\alpha_4 = \sqrt{7} - 1$ which give $\Gamma_4 \geq \frac{1}{2}(5 - \sqrt{7}) \simeq 1.17712$. (The value of α_3 may be changed slightly without affecting the bound obtained on Γ_4.) There is no reason to believe, as we in fact show in the next section, that these bounds are tight.

As an immediate consequence, we get non-linear lower bounds on the complexity of the parity function over the basis U_k, for any $k \geq 2$.

Corollary 4.5 $L_{U_k}(x_1 \oplus x_2 \oplus \cdots \oplus x_n) = \Omega(n^{1+\frac{1}{3k-4}})$.

5 The basis U_3

In this section we consider more closely the basis U_3 of all unate Boolean functions of at most 3 variables. Let $m_3(x_1, x_2, x_3)$ be the 3-bit majority function. It is not difficult to check that all functions of U_3 may be obtained from one of the functions x_1, $x_1 \vee x_2$, $x_1 \wedge x_2$, $x_1 \vee x_2 \vee x_3$, $x_1 \vee (x_2 \wedge x_3)$, $x_1 \wedge x_2 \wedge x_3$, $x_1 \wedge (x_2 \vee x_3)$ and $m_3(x_1, x_2, x_3)$ by permuting and possibly negating some of the arguments. We can therefore transform any U_3-formula into an equivalent formula in which all gates are labeled by one of the functions $x_1 \vee x_2$, $x_1 \wedge x_2$ and $m_3(x_1, x_2, x_3)$ and all the leaves are labelled by literals.

We have seen in the previous section that Γ_3, the shrinkage exponent of U_3-formulae, satisfies $\Gamma_3 \geq 3-\sqrt{3} \simeq 1.26795$. Here we obtain a slightly better lower bound.

Theorem 5.1 *The shrinkage exponent Γ_3 of U_3-formula satisfies $\Gamma_3 \geq \frac{4}{3}$.*

Proof: We again use the weight function $W(T) = L(T) + \alpha G_3(T)$, where α is a parameter to be chosen later. For brevity, we let $M(T) = G_3(T)$. If we consider the arguments given in the previous section for the case $k = 3$, we see that the leaves of an U_3-formula are divided into two categories, those that feed binary gates, and those that feed ternary gates. We use here the more refined division of the leaves into the following three classes:

1. Leaves that feed a binary gate that feeds a binary gate.
2. Leaves that feed a binary gate that feeds a ternary gate.
3. Leaves that feed a ternary gate.

If the root of the formula is a binary gate, then leaves that feed the root of the formula, if there are any, are placed in the first class. We let ℓ_1, ℓ_2 and ℓ_3, respectively, be the number of leaves of these three classes.

Consider now the effect of assigning the values 0 and 1 to a leaf of the first class. As in Subbotovskaya's argument, the total savings is 3. More savings are obtained when the values 0 and 1 are assigned to a leaf of the second class. For at least one of the assigned values the binary gate fed by the leaf now outputs a constant value and the majority gate fed by this blocked binary gate can now be replaced by a binary gate. The total savings is, therefore, at least $3+\alpha$. When the values 0 and 1 are assigned to a leaf of the third class, the total savings is easily seen to be $2+2\alpha$ as in each case at least one leaf and one majority gate may be removed from the formula.

But, do all these savings add up when all the leaves corresponding to the same variables are simultaneously assigned the 0 and 1 values? Not exactly. For example, the formula may contain a subformula of the form $m_3(x \vee \phi, \bar{x} \wedge \psi, \mu)$, where ϕ, ψ and μ are some subformulae. In this case, when we assign 0 to x and 1 to $\bar{x}$ we can only remove two leaves, and when we assign 1 to x and 0 to $\bar{x}$ we may only be able to remove four leaves (when ϕ and ψ are also leaves) and one majority gate. Thus the total savings per leaf is now only $3+\frac{\alpha}{2}$, and not $3+\alpha$ as before. We claim, however, that this is the only case we have to worry about. To see this, we note that we can use the following simple identities to simplify U_3-formulae:

$$
\begin{aligned}
m_3(x, x, \phi) &= x \ , \\
m_3(x, \bar{x}, \phi) &= \phi \ , \\
m_3(x, x \vee \phi, \psi) &= x \vee (\phi \wedge \psi) \ , \\
m_3(x, \bar{x} \vee \phi, \psi) &= (x \wedge \phi) \vee \psi \ , \\
m_3(x \vee \phi, x \vee \psi, \mu) &= x \vee m_3(\phi, \psi, \mu) \ , \\
m_3(x \vee \phi, \bar{x} \vee \psi, \mu) &= ((x \vee \phi) \wedge (\bar{x} \vee \psi)) \vee \mu \ .
\end{aligned}
$$

Also note, that as $\overline{x \vee y} = \bar{x} \wedge \bar{y}$, $\overline{x \wedge y} = \bar{x} \vee \bar{y}$ (de Morgan rules), and as $\overline{m_3(x,y,z)} = m_3(\bar{x},\bar{y},\bar{z})$, all these inequalities remain valid when all the $\vee$'s are replaced by $\wedge$'s, and vice versa. Using these rules it is also easy to see that every subformula of the form $m_3(x^a \circ \phi, x^b \bullet \psi, x^c \diamond \mu)$, where x^a, x^b and x^c stand for either x or $\bar{x}$, and $\circ, \bullet$ and $\diamond$ stand for $\vee$ or $\wedge$, can be simplified. This, and the fact that if $x \circ \phi$ is a subformula of T then we may assume that x and $\bar{x}$ do not occur in ϕ, ensure that the only case in which savings do not add up is indeed the case considered above. We thus have:

$$S(T) \;\geq\; 3\ell_1 + (3 + \frac{\alpha}{2})\ell_2 + (2 + 2\alpha)\ell_3 \;.$$

We now obtain an upper bound on $W(T)$ in terms of ℓ_1, ℓ_2 and ℓ_3. Let B be the number of binary gates in the formula T. As in the previous section, we clearly have $B \geq \frac{\ell_1+\ell_2}{2}$, as $\ell_1 + \ell_2$ is the number of leaves that feed binary gates. The improved bound on Γ_3 is obtained by noticing that each leaf of the first class is not only a child of a binary gate but also a *grandchild* of a binary gate. Thus, $B \geq \frac{3\ell_1-2}{4} + \frac{\ell_2}{2}$. (The -2 term takes into account the fact that at most two leaves of the first class do not have a grandparent.) Recall that M is the number of ternary gates in T and that $L = 2M + B + 1$. Thus,

$$M \;=\; \frac{1}{2}(L - B - 1) \;\leq\; \frac{1}{2}(\ell_1 + \ell_2 + \ell_3 - \frac{3\ell_1 - 2}{4} - \frac{\ell_2}{2} - 1) \;<\; \frac{\ell_1}{8} + \frac{\ell_2}{4} + \frac{\ell_3}{2} \;,$$

$$W(T) = L + \alpha M < (1 + \frac{\alpha}{8})\ell_1 + (1 + \frac{\alpha}{4})\ell_2 + (1 + \frac{\alpha}{2})\ell_3 \;,$$

and

$$\frac{S(T)}{W(T)} \;\geq\; \min\{\, \frac{3}{1+\frac{\alpha}{8}} \,,\, \frac{3+\frac{\alpha}{2}}{1+\frac{\alpha}{4}} \,,\, \frac{2+2\alpha}{1+\frac{\alpha}{2}} \,\} \;.$$

It is easy to see that the optimal choice of α here is $\alpha = 1$. With this choice, we get that whenever $L(T) > 1$ we have $S(T)/W(T) \geq \frac{8}{3}$. By Lemma 4.1 we get, therefore, that $\Gamma_3 \geq \frac{4}{3}$. □

Corollary 5.2 $L_{U_3}(x_1 \oplus x_2 \oplus \cdots \oplus x_n) = \Omega(n^{4/3})$.

We next obtain an upper bound on $L_{U_3}(x_1 \oplus x_2 \oplus \cdots \oplus x_n)$ and therefore also an upper bound on Γ_3.

Lemma 5.3 $L_{U_3}(x_1 \oplus x_2 \oplus \cdots \oplus x_n) = O(n^{1.74})$.

Proof: The 3-bit parity function can be computed using the following U_3-formula of size 7:

$$x_1 \oplus x_2 \oplus x_3 \;=\; m_3(\,x_1\,,\, m_3(\bar{x}_1, x_2, x_3)\,,\, m_3(\bar{x}_1, \bar{x}_2, \bar{x}_3)\,) \;.$$

To verify the correctness of this formula, note that when $x_2 = x_3$, the formula simplifies to x_1, while if $x_2 \neq x_3$, it simplifies to $\bar{x}_1$, as required. Let $P(n)$ be the formula complexity, over U_3 of the n-bit parity function. Using the above formula, we get that $P(n) \leq 7P(\frac{n}{3})$ and therefore $P(n) = O(n^{\log_3 7})$. Note that $\log_3 7 \simeq 1.7712$ so this is a bit worse than the result claimed. We can get a slightly better bound using a simple idea used in [PZ93a]. The size of the 3-bit parity formula is 7, but each of the variables x_2 and x_3 appears in this formula only twice. For any n_1, n_2 and n_3, we therefore get $P(n_1 + n_2 + n_3) \leq 2P(n_1) + 2P(n_2) + 3P(n_3)$. In particular, for any $0 < \alpha < 1$ we get that $P(n) \leq 4P(\alpha n) + 3P((1 - 2\alpha)n)$. It is easy to check that this implies that $P(n) = O(n^\beta)$, where β satisfies $4\alpha^\beta + 3(1 - 2\alpha)^\beta = 1$. Choosing $\alpha \simeq 0.388$, we get that $\beta < 1.74$, as required. □

Corollary 5.4 $\frac{4}{3} \leq \Gamma_3 < 1.74$.

We end this section by mentioning that Gupta and Mahajan [GM97] used the amplification technique of Valiant [Val84] to obtain $\{m_3\}$-formulae of size $O(n^{4.30})$ for the n-bit majority function. Rokhlina [Rok70] (see also [Kva96]) obtains an $\Omega(n^{1.58})$ lower bound on the size of such formulae.

6 Concluding remarks

The main result of this paper is a simple characterization of complete bases with shrinkage exponent equal to 1. These are exactly the complete bases that contain at least one non-unate function.

To prove this result, we obtained an explicit lower bound, strictly larger than 1, on the shrinkage exponent of any basis composed solely of unate functions.

The bounds we obtained are probably far from being optimal. Obtaining tighter bounds on the shrinkage exponents Γ_k remains an open problem. It would be interesting to see whether Håstad's technique [Hås98] could be used to obtain such improved bounds. It may be especially interesting to narrow the gap between the lower and upper bounds given for Γ_3.

Perhaps the most interesting open problem regarding shrinkage exponents is the determination of the shrinkage exponent of *monotone de Morgan* formulae. It is conjectured in [PZ93b] that it is $\log_{\sqrt{5}-1} 2 \simeq 3.27$. This would imply an $\Omega(n^{3.27})$ lower bound on the size of monotone de Morgan formulae for the n-bit majority function.

Acknowledgement

The authors would like to thank two anonymous referees for their valuable comments on a previous version of the paper.

References

[And87a] A.E. Andreev. A method for obtaining efficient lower bounds for monotone complexity. *Algebra and Logica*, 26:1–18, 1987.

[And87b] A.E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Vestnik Moskov. Univ. Mat.*, 42, 1987. (In Russian.) English translation in *Moscow Univ. Math. Bull.* 42:63–66, 1987.

[GM97] A. Gupta and S. Mahajan. Using amplification to compute majority with small majority gates. *Computational Complexity*, 6:46–63, 1997.

[Hås98] J. Håstad. The shrinkage exponent of de Morgan formulas is 2. *SIAM Journal on Computing*, 27:48–64, 1998.

[Kva96] H. Kvartin. Formulae with 3-bit majority gates. Master's thesis, Department of Computer Science, Tel Aviv University, 1996.

[NI93] N. Nisan and R. Impagliazzo. The effect of random restrictions on formulae size. *Random Structures & Algorithms*, 4:121–133, 1993.

[Pos41] E.L. Post. The two-valued iterative systems of mathematical logic. *Annals of Math. Studies*, 5, 1941. Princeton University Press, N.J.

[PZ93a] M. Paterson and U. Zwick. Shallow circuits and concise formulae for multiple addition and multiplication. *Computational Complexity*, 3:262–291, 1993.

[PZ93b] M. Paterson and U. Zwick. Shrinkage of de Morgan formulae under restriction. *Random Structures & Algorithms*, 4:135–150, 1993.

[Rok70] M.M. Rokhlina. On formulae for improving reliability. *Problemy Kibernetiki*, pages 295–301, 1970. (In Russian.).

[Sub61] B.A. Subbotovskaya. Realizations of linear functions by formulas using $+, *, -$. *Doklady Akademii Nauk SSSR*, 136:553–555, 1961. (In Russian.) English translation in *Soviet Mathematics Doklady*, 2:110–112, 1961.

[Val84] L.G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5:363–366, 1984.