

**Original citation:**

Paterson, Michael S. and Zwick, U. (1991) Shrinkage of de Morgan formulae under restriction. University of Warwick. Department of Computer Science. (Department of Computer Science Research Report). (Unpublished) CS-RR-171

**Permanent WRAP url:**

<http://wrap.warwick.ac.uk/60867>

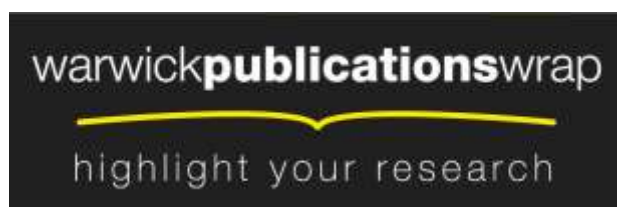
**Copyright and reuse:**

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**A note on versions:**

The version presented in WRAP is the published version or, version of record, and may be cited as it appears here. For more information, please contact the WRAP Team at: [publications@warwick.ac.uk](mailto:publications@warwick.ac.uk)



<http://wrap.warwick.ac.uk/>

# \_\_\_\_Research report 171\_\_\_\_\_

## SHRINKAGE OF DE MORGAN FORMULAE UNDER RESTRICTION

MICHAEL S PATERSON, URI ZWICK  
(RR171)

### Abstract

It is shown that a random restriction leaving only a fraction  $\epsilon$  of the input variables unassigned reduces the expected de Morgan formula size of the induced function by at least a factor of  $\epsilon^{\frac{5-\sqrt{5}}{2}} \simeq \epsilon^{1.63}$ . (A de Morgan formula is a formula over the basis  $\{\wedge, \vee, \neg\}$ .)

This improves a long-standing result of  $\epsilon^{1.5}$  by Subbotovskaya and a recent improvement to  $\epsilon^{\frac{21-\sqrt{73}}{8}} \simeq \epsilon^{1.55}$  by Nisan and Impagliazzo.

The new exponent yields an increased lower bound of  $\Omega(n^{\frac{7-\sqrt{2}}{2}-o(1)})$  for the de Morgan formula size of a function defined by Andreev. This is the largest lower bound known for a function in NP.

1. The first part of the paper is devoted to a general discussion of the problem of the existence of solutions of the system of equations

$$\begin{aligned} \frac{dx}{dt} &= f(x, y, z, t) \\ \frac{dy}{dt} &= g(x, y, z, t) \\ \frac{dz}{dt} &= h(x, y, z, t) \end{aligned}$$

where  $f, g, h$  are continuous functions of  $x, y, z, t$  and

the initial conditions are given by the system of equations

$$\begin{aligned} x(0) &= x_0 \\ y(0) &= y_0 \\ z(0) &= z_0 \end{aligned}$$

where  $x_0, y_0, z_0$  are constants. The existence of solutions of this system of equations is guaranteed by the theorem of Peano.

The second part of the paper is devoted to a study of the stability of the solutions of the system of equations. It is shown that the solutions are stable if the functions  $f, g, h$  satisfy certain conditions.

# Shrinkage of de Morgan formulae under restriction \*

Michael S. Paterson <sup>†</sup>      Uri Zwick <sup>‡</sup>

January 17, 1991

## Abstract

It is shown that a random restriction leaving only a fraction  $\epsilon$  of the input variables unassigned reduces the expected de Morgan formula size of the induced function by at least a factor of  $\epsilon^{\frac{8-\sqrt{3}}{2}} \simeq \epsilon^{1.63}$ . (A de Morgan formula is a formula over the basis  $\{\wedge, \vee, \neg\}$ .)

This improves a long-standing result of  $\epsilon^{1.5}$  by Subbotovskaya and a recent improvement to  $\epsilon^{\frac{21-\sqrt{73}}{8}} \simeq \epsilon^{1.55}$  by Nisan and Impagliazzo.

The new exponent yields an increased lower bound of  $\Omega(n^{\frac{7-\sqrt{3}}{2}-o(1)})$  for the de Morgan formula size of a function defined by Andreev. This is the largest lower bound known for a function in NP.

## 1 Introduction

In [6], Subbotovskaya introduced the random restriction method and used it to derive an  $\Omega(n^{1.5})$  lower bound on the de Morgan formula size of the parity function. Khrapchenko [3],[4] (see also [2],[7],[8]) then used a different method to improve the lower bound for the parity function to a tight  $\Omega(n^2)$  bound. Quite a few years then passed before Andreev [1] observed that the random restriction method used by Subbotovskaya could be used to derive an  $\Omega(n^{2.5-o(1)})$  lower bound on the de Morgan formula size of a rather natural function in P. The function he used could be easily shown to have formulae of size  $O(n^3)$ .

The following basic observation is used in Subbotovskaya's proof that a random restriction leaving only a fraction  $\epsilon$  of the variables of a function unassigned reduces the expected formula size of the induced function by a factor of at least  $\epsilon^{1.5}$ . If a leaf in a de Morgan formula is set once to 0 and once to 1, then under one of these assignments the size of the formula decreases by at least one and under the other by at least two, giving a total of at least 3 and an average of at least 1.5.

---

\*This research was partially supported by the ESPRIT II BRA Programme of the EC under contract # 3075 (ALCOM).

<sup>†</sup>Department of Computer Science, University of Warwick, Coventry, CV4 7AL, England.

<sup>‡</sup>Mathematics Institute, University of Warwick, Coventry, CV4 7AL, England.



Any formula in which all the leaves are arranged in pairs, or *twigs* as we shall call them, and in which the gates in the lowest two levels alternate, shows that the preceding argument in its present form cannot be strengthened.

Nisan and Impagliazzo recently found a neat way of improving Subbotovskaya's result. They argue that even if some formula is in a form in which only small savings can be expected, subsequent reductions are likely to transform its structure into one in which greater savings are expected. In order to capture this property they assign to every formula a *weight* that takes into account not only its size but also the likelihood of further size reductions in the subsequent random assignments. A formula with a greater potential for savings will get a smaller weight, reflecting the fact that if this formula is obtained progress has been made, even if only a small reduction of size was obtained.

A constant  $\gamma$  will be called a *shrinkage exponent* if a reduction by a factor of at least  $\epsilon^\gamma$  in the formula size is expected when a random restriction that leaves only a fraction  $\epsilon$  of the inputs unassigned is applied.

In the sequel we follow Nisan's and Impagliazzo's approach. We use however a different weight function which we consider to be more natural. The new weight function enables us to improve upon Nisan's and Impagliazzo's shrinkage exponent.

Nisan and Impagliazzo define the *neighbour tree* of a leaf in a formula as the other subtree of this leaf's parent. A leaf is called a *victim* if it is contained in a neighbour tree that contains at least two leaves. They then define the weight of a function to be its formula size minus a small credit for every victim. Our weight function simply adds a penalty for every twig contained in the formula.

The main result of this note is Theorem 2.1 that improves an analogous result in Nisan's and Impagliazzo's work. The improved shrinkage exponent and the improved lower bound for Andreev's function then follow using the standard arguments that can be found in Nisan's and Impagliazzo's paper [5], Andreev's paper [1] or Dunne's book [2]. For a description of Andreev's function the reader is also referred to these sources.

In proving Theorem 2.1 we use a top-down approach which conceptually simplifies the proof. This is another difference between our work and that of Nisan and Impagliazzo. We feel that our approach is more likely to produce further improvements.

## 2 An improved shrinkage exponent

A de Morgan formula is a binary tree in which each leaf is labeled by a literal from the set  $\{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$  and each internal node functions as either an AND ( $\wedge$ ) or an OR ( $\vee$ ) gate. The *size* of a formula  $T$  is defined to be the number of leaves in it and it is denoted by  $L(T)$ . A *twig* is subtree with just two leaves. The number of twigs contained in a formula  $T$  is denoted by  $\tau(T)$ . The *weight* of a formula  $T$  is defined to be  $\omega(T) = L(T) + \alpha \cdot \tau(T)$  where  $\alpha$  is a parameter. We will take  $\alpha = \sqrt{3} - 1$ .

Every formula  $T$  computes in a natural way a function, denoted by  $f(T)$ . For a function  $f$  we denote by  $L(f)$  the minimal size of a formula computing  $f$  and by

$\omega(f)$  the minimal weight of a formula computing  $f$ . If  $f$  is a function, we denote by  $f_{x=0}, f_{x=1}$  the functions obtained from  $f$  by assigning to  $x$  the value 0 or 1 respectively. For a function  $f$ , we define  $\sigma_{x=c}(f) = \omega(f) - \omega(f_{x=c})$  for  $c = 0, 1$ ,  $\sigma_x(f) = \sigma_{x=0}(f) + \sigma_{x=1}(f)$  and  $\sigma(f) = \sum_{x \in f} \sigma_x(f)$ , where the summation is over all the variables on which  $f$  depends.

**Theorem 2.1** *For any function  $f$  with  $\omega(f) > 1$  we have*

$$\frac{\sigma(f)}{\omega(f)} \geq 2\gamma \quad \text{where} \quad \gamma = \frac{3+2\alpha}{2+\alpha} = \frac{5-\sqrt{3}}{2}.$$

An immediate corollary is

**Corollary 2.2** *For any function  $f = f(x_1, \dots, x_n)$  with  $\omega(f) > 1$  we have*

$$E[\omega(f')] \leq \left(1 - \frac{\gamma}{n}\right) \omega(f)$$

where  $E$  denotes expectation and  $f'$  is the random function  $f_{x_i=c}$  obtained by choosing  $i \in \{1, 2, \dots, n\}$  and  $c \in \{0, 1\}$  with equal probabilities.

Using induction, some analysis and the fact that for any function  $f$  we have  $\omega(f)/(1 + \frac{\alpha}{2}) \leq L(f) \leq \omega(f)$ , we get

**Corollary 2.3** *If  $f = f(x_1, \dots, x_n)$  and  $f'$  is obtained by randomly choosing  $n - m$  variables out of  $x_1, \dots, x_n$  and randomly assigning them values, then*

$$E[L(f')] \leq c \left(\frac{m}{n}\right)^\gamma L(f)$$

where  $c$  is a fixed constant and  $\gamma = \frac{3+2\alpha}{2+\alpha} = \frac{5-\sqrt{3}}{2} \simeq 1.63397$ .

This improves the analogous result in [5] with  $\gamma = \frac{21-\sqrt{73}}{8} \simeq 1.557$ .

Using the last corollary in the standard way, the  $\Omega(n^{2.5-o(1)})$  lower bound on the formula size of the Andreev function, derived by Andreev in [1] (see also [2]) and later improved by Nisan and Impagliazzo in [5] to  $\Omega(n^{2.557})$  can now be further improved to  $\Omega(n^{2.633})$ :

**Corollary 2.4** *The formula size of the Andreev function satisfies*

$$L(A_n) = \Omega(n^{\frac{7-\sqrt{3}}{2}-o(1)}).$$

### 3 Preliminaries

Before proceeding it is useful to extend the definition of formula and allow leaves to be labeled by constants. (Note that, although we have not said so explicitly, we already consider a tree with one leaf, labeled by a constant, to be a formula with zero size and weight.)



The size of a formula containing constant leaves is defined to be the number of non-constant leaves in it. The weight of such a formula is obtained by adding the constant  $\alpha$  for every twig, even if its two leaves, or one of them, are labeled by constants.

We first show that this extension does not change the weights that were assigned to functions.

**Lemma 3.1** *For any formula  $T$  computing a non-constant function, there exists a formula  $T'$  with no constant leaves that computes the same function and for which  $\omega(T') \leq \omega(T)$ .*

**Proof:** Let  $\ell$  be a leaf in  $T$  labeled by a constant  $c$ . Let  $m$  be the parent of  $\ell$  and let  $S$  be the neighbour tree of  $\ell$ .

If the function computed at  $m$  is constant, then remove  $\ell$  and  $S$  from  $T$  making  $m$  a leaf labeled by the appropriate constant. This action did not increase the number of twigs (at most one new twig, involving  $m$ , was formed, but at least one twig was also removed) and clearly did not increase the number of non-constant leaves in  $T$ . The weight therefore could have only decreased and the new tree still computes the same function.

If the function computed at  $m$  is not constant, then it is equal to the function computed by  $S$ . We can therefore replace the subtree of  $\ell, m$  and  $S$  in  $T$  by  $S$  alone. Again the function computed did not change and its weight did not increase.

By repeating the above process we eventually end up with a formula with the required properties. □

The following lemma is now trivial.

**Lemma 3.2** *Let  $T$  be a formula and let  $f'$  be a function computed by  $T$  when  $k$  of its non-constant leaves are assigned constant values. Then,  $\omega(f') \leq \omega(T) - k$ .*

This lemma will be used in the next section to show that if  $f$  depends on  $x_1, \dots, x_k$  and  $f' = f_{x_1=c_1, \dots, x_n=c_n}$  then  $\omega(f') \leq \omega(f) - k$ .

It is interesting to note at this stage that Lemmas 3.1 and 3.2 are not valid for the weight function of Nisan and Impagliazzo. These properties make our weight function easier to work with.

Next note that if  $f = f_1 \vee f_2$  (or  $f = f_1 \wedge f_2$ ) where  $\omega(f) = \omega(f_1) + \omega(f_2)$  and  $f_1|_{x=c}$  and  $f_2|_{x=c}$  are not both literals then  $\sigma_{x=c}(f) \geq \sigma_{x=c}(f_1) + \sigma_{x=c}(f_2)$ , i.e., the savings behave super-additively. The hard cases in the proof of Theorem 3.1 will be those in which  $f_1$  and  $f_2$  reduce to literals under certain assignments. We therefore need some understanding of the circumstances under which this can happen.

Let  $f$  be a function. We write  $x \Rightarrow y$  where  $x$  and  $y$  are literals if  $f_{x=1} = y$ , or in other words if  $f = xy \vee \bar{x}g$  for some function  $g$ . As an example note that if  $f_{x=0} = y$  then this is expressed by writing  $\bar{x} \Rightarrow y$ . The collection of all the relations  $x \Rightarrow y$  valid for the function  $f$  will be called the *solo structure* of  $f$ . The next lemma classifies the possible solo structures that functions can have.

**Lemma 3.3** *If  $f$  has a non-empty solo structure then it has of one of the following forms :*

1.  $\{x \Rightarrow y, y \Rightarrow x, \bar{x} \Rightarrow \bar{y}, \bar{y} \Rightarrow \bar{x}\}$  and  $f = xy \vee \bar{x}\bar{y}$ .
2.  $\{x \Rightarrow y, \bar{x} \Rightarrow z\}$  and  $f = xy \vee \bar{x}z$ .
3.  $\{x \Rightarrow y, y \Rightarrow x\}$  and  $f = xy \vee \bar{x}\bar{y}g$ .
4.  $\{x \Rightarrow y, \bar{y} \Rightarrow \bar{x}\}$  and  $f = xy \vee \bar{x}\bar{y} \vee \bar{x}yg$ .
5.  $\{x_1 \Rightarrow y, \dots, x_k \Rightarrow y\}$  for some  $k \geq 1$  and  $f = (x_1 \vee \dots \vee x_k)y \vee \bar{x}_1 \dots \bar{x}_k g$ .

*In cases (3) and (4) the function  $g$  does not depend on  $x$  or  $y$ , in case (3)  $g \neq 1$ , in case (4)  $g \neq 0$ , and in case (5) the function  $g$  does not depend on  $x_1, \dots, x_k$  and if  $k = 1$  then  $g \neq 0$ .*

**Proof :** Let  $x \Rightarrow y$  be in the solo structure of  $f$ . If  $x \Rightarrow y$  is the only relation in the solo structure of  $f$  then  $f$  is of the form (5) with  $k = 1$ . Suppose therefore that  $x' \Rightarrow y'$  is also in the solo structure of  $f$ .

If  $x' = x$  then  $y = y'$  and therefore  $x' \Rightarrow y'$  is not a new relation. If  $x' = \bar{x}$  then either  $y' = \bar{y}$  in which case  $f$  is of the form (1), or  $y' = z$  in which case  $f$  is of the form (2). The case  $x \Rightarrow y$  and  $\bar{x} \Rightarrow y$  is impossible since then  $f = y$ .

If  $x' = y$  then  $f = xy \vee \bar{x}g = yy' \vee \bar{y}h$  and  $f_{x=1, y=1} = y'_{x=1} = 1$  and therefore  $y' = x$ . It follows that  $f = xy \vee \bar{x}\bar{y}g$  and therefore  $f$  is either of the form (1) if  $g = 1$  or of the form (3) otherwise.

If  $x' = \bar{y}$  then similarly we get that  $y' = \bar{x}$  and  $f$  is either of the form (1) or (4).

If  $x' \neq x, \bar{x}, y, \bar{y}$  then by assigning  $x \rightarrow 1, x' \rightarrow 1$  simultaneously we get that  $y = y'$ . The function is therefore of the form (5) with  $k \geq 2$ . □

As an example note that  $xy$  is of the form (3) with  $g = 0$ ,  $x \oplus y = x\bar{y} \vee \bar{y}x = x\bar{y} \vee \bar{x}\bar{y}$  is of the form (1) and  $x \vee y = \bar{x}\bar{y} \vee \bar{x}\bar{y} \vee \bar{x}y$  is of the form (4) with  $g = 1$ .

## 4 The Proof

**Proof (of Theorem 3.1) :**

The proof is by induction on the weight of  $f$ . As the basis of the induction we verify the theorem directly for all functions with size at most 4. These turn out to be exactly the functions with weight at most  $4 + 2\alpha \simeq 5.46$  or equivalently with weight less than  $5 + \alpha \simeq 5.73$ . The details may be found in Table 4.1. As can be seen from the table, the values of  $\alpha$  and  $\gamma$  were chosen to satisfy  $2\gamma = \frac{6+4\alpha}{2+\alpha} = \frac{10+3\alpha}{3+\alpha}$ .

Suppose now that  $L(f) > 4$  (and therefore  $\omega(f) > 5.7$ ) and that the theorem holds for all the non-trivial functions with less weight than  $f$ . (A function is said to be trivial if it is a constant or a literal.) By definition, there exist two functions  $f_1, f_2$  such that  $f = f_1 \vee f_2$  or  $f = f_1 \wedge f_2$ , and  $\omega(f) = \omega(f_1) + \omega(f_2)$ . Without loss of generality we will assume that  $f = f_1 \vee f_2$ , the other case being dual.



| <u>weight</u> | <u>size</u> | <u>function</u>          | $\sigma(f)/\omega(f)$                    |
|---------------|-------------|--------------------------|--|
| $2 + \alpha$  | 2           | $x \vee y$               | $\frac{6+4\alpha}{2+\alpha} = 2\gamma$   |
| $3 + \alpha$  | 3           | $x \vee yz$              | $\frac{10+3\alpha}{3+\alpha} = 2\gamma$  |
| $3 + \alpha$  | 3           | $x \vee y \vee z$        | $\frac{12+3\alpha}{3+\alpha} > 2\gamma$  |
| $4 + \alpha$  | 4           | $x \vee yzw$             | $\frac{17+4\alpha}{4+\alpha} > 2\gamma$  |
| $4 + \alpha$  | 4           | $x \vee y(z \vee w)$     | $\frac{15+2\alpha}{4+\alpha} > 2\gamma$  |
| $4 + \alpha$  | 4           | $x \vee y \vee zw$       | $\frac{16+2\alpha}{4+\alpha} > 2\gamma$  |
| $4 + \alpha$  | 4           | $x \vee y \vee z \vee w$ | $\frac{20+4\alpha}{4+\alpha} > 2\gamma$  |
| $4 + 2\alpha$ | 4           | $xy \vee zw$             | $\frac{12+8\alpha}{4+2\alpha} = 2\gamma$ |
| $4 + 2\alpha$ | 4           | $xy \vee \bar{x}\bar{y}$ | $\frac{12+8\alpha}{4+2\alpha} = 2\gamma$ |
| $4 + 2\alpha$ | 4           | $xy \vee \bar{x}z$       | $\frac{14+8\alpha}{4+2\alpha} > 2\gamma$ |

Table 4.1: Saving factors for all functions with  $L(f) \leq 4$ .

If one of  $f_1$  or  $f_2$  is trivial then the required inequality for  $f$  follows from the induction hypothesis and from Lemma 4.1 below.

**Lemma 4.1** *If  $f = x \vee g$  then  $\sigma(f)/\omega(f) \geq 2\gamma$ .*

**Proof :** Let  $k$  be the number of different assignments under which  $g$  reduces to a literal. Note that since  $\omega(f) = 1 + \omega(g)$ , the function  $g$  is independent of  $x$ .

If  $k \leq 4$  then using the induction hypothesis for  $g$  we get that

$$\begin{aligned}
\sigma(x \vee g) &\geq 1 + \omega(f) + \sigma(g) - 4\alpha \\
&\geq 1 + \omega(f) + 2\gamma(\omega(f) - 1) - 4\alpha \\
&\geq (1 + 2\gamma)\omega(f) - (4\alpha + 2\gamma - 1) \\
&\geq 2\gamma \cdot \omega(f)
\end{aligned}$$

since  $\omega(f) > 5.7 > (4\alpha + 2\gamma - 1) \simeq 5.2$ .

If  $k \geq 5$  then

$$\begin{aligned}
\sigma(x \vee g) &\geq 1 + \omega(f) + 5(\omega(f) - 2 - \alpha) \\
&= 6\omega(f) - (9 + 5\alpha) \\
&\geq 2\gamma \cdot \omega(f)
\end{aligned}$$

since  $\omega(f) > 5.7 > (9 + 5\alpha)/(6 - 2\gamma) \simeq 4.7$ .

Thus, the proof of the lemma is complete.  $\square$

Otherwise, both  $f_1$  and  $f_2$  are non-trivial and the required inequality for  $f$  follows from the induction hypothesis and from Lemma 3.2 below.

**Lemma 4.2** If  $f = f_1 \vee f_2$  then  $\sigma(f)/\omega(f) \geq 2\gamma$ .

**Proof:** Since  $\omega(f) = \omega(f_1) + \omega(f_2)$ , it is enough to prove that  $\sigma(f) \geq \sigma(f_1) + \sigma(f_2)$ . Let  $k$  be the number of different assignments under which both  $f_1$  and  $f_2$  simplify to literals based on different variables. It is clear that  $\sigma(f) \geq \sigma(f_1) + \sigma(f_2) - k\alpha$ . This inequality does not take into account extra savings that arise in cases where  $\omega(f_{x=c}) < \omega(f_{1|x=c}) + \omega(f_{2|x=c})$ . We will show that these extra savings total to more than  $k\alpha$ .

The proof now breaks into three cases according to the value of  $k$ .

**Case 1.**  $k = 0$

There is nothing to prove here.

**Case 2.**  $1 \leq k \leq 2$

Let  $x \rightarrow 1$  be one of special assignments. Then  $f_1 = xy \vee \bar{x}g$  and  $f_2 = xz \vee \bar{x}h$  for some functions  $g, h$  independent of  $x$  where the literals  $x, y, z$  are all of distinct variables. Without loss of generality we may assume that  $x, y, z$  are variables.

Consider the effect on  $f$  of setting  $y$  or  $z$  to 1 :

$$\begin{aligned} f &= (xy \vee \bar{x}g) \vee (xz \vee \bar{x}h) \\ f_{y=1} &= (x \vee g_{y=1}) \vee (xz \vee \bar{x}h_{y=1}) \\ f_{z=1} &= (xy \vee \bar{x}g_{z=1}) \vee (x \vee h_{z=1}) \end{aligned}$$

Since  $f_{1|y=1} = x \vee g_{y=1}$  we have  $f_{y=1} = f_{1|y=1} \vee f_{2|y=1, x=0}$ . If  $h_{y=1} \neq z$  then  $f_{2|y=1} = xz \vee \bar{x}h_{y=1}$  depends on  $x$  and therefore any formula for  $f_{2|y=1}$  contains at least one occurrence of  $x$ . Lemma 3.1 therefore implies that  $\omega(f_{2|y=1, x=0}) \leq \omega(f_{2|y=1}) - 1$ , giving us an extra saving of at least 1 when  $y \rightarrow 1$ .

Similarly, if  $g_{z=1} \neq y$ , we get again an extra saving of at least 1 when  $z \rightarrow 1$ .

If both  $h_{y=1} \neq z$  and  $g_{z=1} \neq y$  hold then these two extra savings add up to give an extra credit of  $2 > 2\alpha$ , which is enough if  $k \leq 2$ .

Suppose now that  $h_{y=1} = z$  but  $g_{z=1} \neq y$ , the case where  $h_{y=1} \neq z$  but  $g_{z=1} = y$  being similar. We then have  $f_2 = (x \vee y)z \vee \bar{x}\bar{y}h$  for some  $h$ .

Consider again the effect of setting  $z$  to 1

$$\begin{aligned} f &= (xy \vee \bar{x}g) \vee ((x \vee y)z \vee \bar{x}\bar{y}h) \\ f_{z=1} &= (xy \vee \bar{x}g_{z=1}) \vee (x \vee y \vee h_{z=1}) \end{aligned}$$

Using the same argument as before we get that  $f_{z=1} = f_{1|x=1, y=0, z=0} \vee f_{2|x=1}$ . The function  $f_{1|x=1} = xy \vee \bar{x}g_{z=1}$  depends both on  $y$  (as seen with  $x \rightarrow 1$ ) and on  $x$  (since we assumed that  $g_{z=1} \neq y$ ). Using Lemma 3.1 we get extra savings of at least 2 from this case alone. Again this is enough if  $k \leq 2$ .

The last case to consider here is when both  $h_{y=1} = z$  and  $g_{z=1} = y$ . We then have  $f_1 = (x \vee z)y \vee \bar{x}\bar{z}g$  and  $f_2 = (x \vee y)z \vee \bar{x}\bar{y}h$ , for some  $g$  and  $h$ . The effect of setting  $y$  or  $z$  to 1 is now

$$\begin{aligned} f &= ((x \vee z)y \vee \bar{x}\bar{z}g) \vee ((x \vee y)z \vee \bar{x}\bar{y}h) \\ f_{y=1} &= x \vee z \vee g_{y=1} \vee z \\ f_{z=1} &= y \vee (x \vee y \vee h_{z=1}) \end{aligned}$$



We get an extra saving of 1 when  $y$  is set to 1, and an additional such saving when  $z$  is set to 1. The total of 2 is again enough for  $k \leq 2$ .

**Case 3.**  $k \geq 3$ .

The functions  $f_1$  and  $f_2$  both reduce to literals for at least three different assignments. According to Lemma 3.3,  $f_1$  and  $f_2$  are either both of the form  $xy \vee \bar{x}\bar{y}$  (or one of its duals), in which case the condition  $\omega(f) = \omega(f_1) + \omega(f_2)$  is clearly violated, or else all  $k$  assignments involve literals  $x_1, \dots, x_k$  of different variables. In this case,  $f_1 = Xy \vee \bar{X}g$  and  $f_2 = Xz \vee \bar{X}h$ , where  $X = x_1 \vee \dots \vee x_k$ , the functions  $g$  and  $h$  do not depend on  $x_1, \dots, x_k$  and  $y, z$  are literals of distinct variables, distinct also from the variables of  $X$ . Again, without loss of generality, we may assume that  $x_1, \dots, x_k, y, z$  are (distinct) variables.

If  $h_{y=1} \neq z$  then as in the previous case we get an extra saving of at least  $k$  when  $y$  is set to 1 since all the appearances of  $x_1, \dots, x_k$  in  $f_2|_{y=0}$  could be taken to be 0. A similar situation arises if  $g_{z=1} \neq y$ .

The only case left to consider is therefore when  $f_1 = (X \vee z)y \vee \bar{X}\bar{z}g$  and  $f_2 = (X \vee y)z \vee \bar{X}\bar{y}h$  where  $g, h$  are some functions. In this case we prove directly that  $\omega(f)/\omega(f) \geq 2\gamma$ . Note that  $\sigma_{x_i=1}(f) = \omega(f) - (2 + \alpha)$  for  $1 \leq i \leq k$ ,  $\sigma_{y=1}(f) \geq 1 + (\omega(f_2) - 1) = \omega(f_2)$  and  $\sigma_{y=1}(f) \geq 1 + (\omega(f_2) - 1) = \omega(f_1)$ . The first relation holds because  $f$  reduces to a twig when each one of the  $x_i$ 's is set to 1. The second relation (and similarly the third) holds because  $\sigma_{y=1}(f_1) \geq 1$  and because  $f_2$  is reduced to a literal when  $y$  is set to 1 and therefore  $\sigma_{y=1}(f_2) = \omega(f_2) - 1$ . Lemma 3.1 implies that  $\sum_v \sigma_{v=0}(f) \geq L(f) \geq \omega(f)/(1 + \alpha/2)$ . Putting all this together and using the fact that  $k \geq 3$  we get that

$$\sigma(f) \geq (4 + \frac{1}{1 + \alpha/2})\omega(f) - 3(2 + \alpha) \geq 2\gamma \cdot \omega(f)$$

since

$$\omega(f) > 5.7 > \frac{3(2 + \alpha)^2}{4} \simeq 5.6.$$

This completes the proof of the lemma. □

The proof of Theorem 2.1 is now complete. □

## 5 Possible improvements

A natural extension of our proof method would be to define new weight functions giving credits or debits for other small trees contained in the formula. Such an approach may yield an increasing sequence of shrinkage exponents.

The parity function is an example proving that 2 is an upper bound for shrinkage exponents. The gap between this and our lower bound presents a challenge.



## References

- [1] A.E. Andreev, *On a method for obtaining more than quadratic effective lower bounds for the complexity of  $\pi$ -schemes*. Vestnik Moskov. Univ. Mat. 42(1)(1987) 70-73 (in Russian). English translation in Moscow Univ. Math. Bull. 42(1)(1987) 63-66.
- [2] P.E. Dunne, *The complexity of Boolean networks*. Academic Press, 1988.
- [3] V.M. Khrapchenko, *Complexity of the realization of a linear function in the class of  $\pi$ -circuits*. Mat. Zametki 9 (1971) 35-40 (in Russian). English translation in Math. Notes Acad. Sciences USSR 9 (1971) 21-23.
- [4] V.M. Khrapchenko, *A method of determining lower bounds for the complexity of  $\Pi$ -schemes*. Mat. Zametki 10 (1972) 83-92 (in Russian). English translation in Math. Notes Acad. Sciences USSR 10 (1971) 474-479.
- [5] N. Nisan, R. Impagliazzo, *Improved lower bounds for formulae size*. Preprint.
- [6] B. A. Subbotovskaya, *Realizations of linear functions by formulas using  $+$ ,  $*$ ,  $-$* , Doklady Akademii Nauk SSSR 136 (1961) 553-555 (in Russian). English translation in Soviet Mathematics Doklady 2 (1961) 110-112.
- [7] I. Wegener, *The complexity of Boolean functions*. Wiley-Teubner Series in Computer Science, 1987.
- [8] U. Zwick, *An extension of Khrapchenko's theorem*. Information Processing Letters (to appear).

