References

[1] A. M. Frieze. On the Lagarias-Odlyzko Algorithm for the Subset Sum Problem. *Siam J. Comp* (1986) vol. 15, no. 2, 536-539.

[2] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Proceedings of IEEE symposium on the foundations of Computer Science.* (1983) 1-10.

[3] A. K. Lenstra, H. W. Lenstra and L. Lovàsz. Factoring polynomials with rational coefficients. *Math. Annalen* 261 (1982) 515-534.

[4] J. E. Mazo and A. M. Odlyzko. Lattice Points in High-dimensional Spheres. *AT&T Technical Memorandum*

FCT'91

Optimal versus Stable in Boolean Formulae

Stasys JUKNA
Institute of Mathematics
Lithuanian Academy of Sciences
SU-232600 Vilnius, Akademijos 4

Abstract. We present some initial ideas of a new lower bound technique which captures, in a strong way, the structure of optimal circuits. The key observation is that optimal circuits are unstable not only w.r.t. deletion of gates but also w.r.t. renaming gates. Such an unstability allows one to extract some useful information about the inner structure of optimal circuits. We demonstrate the technique by new exponential lower bounds on the size of null-chain-free formulae over the basis $\{\wedge, \vee, \neg\}$ approximating subsets of the Boolean $n-$cube, and in particular, for formulae computing "semi-slice" functions, i.e. functions f such that, for some $k < l \leq m$,

$$f = f \wedge T_k^n \wedge \neg T_l^n \vee T_m^n,$$

where T_k^n is the $k-$th threshold function.

Introduction

One reason for the failure of existing methods to derive superpolynomial lower bounds on circuit size lies in the fact that these methods have made little use of the structure of optimal circuits. An information about the form of minimal circuits appears to be very difficult to obtain. The only information about an optimal circuit is that it computes a given function f and no circuit of smaller size computes f. In particular, any deletion of a gate from an optimal circuit spoils the function computed by the whole circuit. Even this simple observation has led to some new information which has been successfully employed by Krichevskii [6] in his proof of non-linear lower bound for the complexity of computing the threshold function in the class of formulae over $\{\&, \vee, \neg\}$. The observation has also been utilized in a powerful technique for monotone circuits called "replacement rules", which was independently proposed by Paterson [9] and Melhorn and Galil [7], and generalized by Dunne [3].

It appears, however, (see Theorem 1 bellow) that optimal circuits (over arbitrary bases) are unstable not only with respect to deletion of gates but also in a much more stronger sense: any replacement of a gate by a different one spoils the function computed by the whole circuit (the size remains unchanged). This result supports an intuitive idea that in optimal circuits the necessity of each of its gates must be "strongly motivated" by some input vectors detecting all its possible faults. The properties of these motivation vectors may be used to extract some useful information about the behavior or structure of optimal circuits. The point is that although the properties of motivating vectors

depend on the circuits hardware (which is a "black box" for us), these vectors also depends heavily on combinatorial properties of a Boolean function computed. In case when such a dependence can be captured, one obtains an additional information about minimal circuits which, in its turn, can be used in proving lower bounds on their size.

We apply this idea to prove new exponential lower bounds on the size of formulae over the basis $\{\wedge, \vee, \neg\}$ with restricted usage of $\neg$-gates computing so-called semi-slice functions. The restriction is that formulae has no short null-chains, i.e. chains with zero conductivity. The class of slice-like Boolean functions is one which is promising for the proof of nonlinear lower bounds for unrestricted circuits. In spite of the fact that (for other functions) exponential lower bounds for null-chain-free formulae are already known, we hope that proof techniques based on unstability of optimal circuits may be fruitful for more general circuits.

The paper is organized as follows. In section 1 we prove (Theorem 1) that optimal circuits (over arbitrary bases) are unstable in a very strong sense: any replacement of a gate by another one spoils the function computed by the whole circuit. The general idea of how this phenomenon can be used in proving lower bounds is also described. In section 2 we consider formulae over the basis $\{\wedge, \vee, \neg\}$ without short null-chains. We show that the unstability of optimal formulae implies that some of their subformulae have very special structure. This is used to derive a general lower bound on the size of such formulae approximating subsets of the Boolean n-th cube (Theorem 2). In section 3 this bound is applied to obtain new lower bounds for the complexity of semislice functions.

1. Optimal versus Stable: General Observations

Throughout the paper $\mathcal{B}_n$ denotes the set of all Boolean functions $f : \{0,1\}^n \longrightarrow \{0,1\}$. Given a Boolean circuit S over some basis $\mathcal{B} \subseteq \mathcal{B}_m (m \geq 1)$ and its gate e, let $\hat{e} \in \mathcal{B} \cup \{x_1, \ldots, x_n\}$ denote its label and $f_e \in \mathcal{B}_n$ the function computed at the gate e in S. The *size* of S is the number of inner gates in S.

Convention: To simplify notations we will denote by S a circuit as well as the function computed at its output gate; the meaning will be clear from the context.

For a gate e of S and a function $h \in \mathcal{B}_m$, let $S_{e \to h}$ denote the circuit obtained from S as follows: replace the label of e by h and remove all the gates which become redundant in the resulting circuit.

Given a class of Boolean functions $\mathcal{F} \subseteq \mathcal{B}_n$, we say that a circuit S *represents* $\mathcal{F}$ if S computes a function from $\mathcal{F}$, i.e. if $S \in \mathcal{F}$. Thus, for example, S computes f iff S represents the class $\mathcal{F} = \{f\}$. A circuit S is *optimal* for $\mathcal{F}$ if S represents $\mathcal{F}$, and no circuit of smaller size does it. Thus, optimal circuits are *unstable* with respect to deletion of gates. It appears however that optimal circuits are unstable in a stronger sense: we cannot replace even the labels of gates; the size of the resulting circuit remains the same but, nevertheless, the function computed differs from the original one.

Definition: A function $h \in \mathcal{B}_m$ is called to be a *strong neighbor* of a function $g \in \mathcal{B}_m$ if either $h \leq \neg g$ or $h \geq \neg g$ (or both). h is a *neighbor* of g if either it is a strong neighbor of g or g depends on at least two variables and $g \oplus x_i \leq g \oplus h$ for some $i \in \{1, \ldots, n\}$. A neighbor of a gate is a neighbor of its label.

For example, neighbors of the disjunction $\vee$ are all the two variable Boolean functions, except $\oplus$ and the function $\vee$ itself. Constants 0 and 1 are strong neighbors of all non-constant functions.

Given an optimal circuit S representing $\mathcal{F}$, a gate e of S and its neighbor h, an *h-motivation* for e in S with respect to $\mathcal{F}$ is a subset of vectors $W \subseteq \{0,1\}^n$ motivating the necessity of this gate:

$$\forall f \in \mathcal{F} \;\; \exists v \in W \; : \qquad S_{e \to h}(v) \neq f(v).$$

Theorem 1: *Let S be an optimal circuit representing a class of Boolean functions $\mathcal{F}$ with $\mathcal{F} \cap \{0,1\} = \emptyset$, and let e be any of its gates. Let h be a strong neighbor of e or an arbitrary neighbor of e if $|\mathcal{F}| = 1$. Then the circuit $S_{e \to h}$ does not represent $\mathcal{F}$.*

Proof: Let $f_1, \ldots, f_m$ be the functions computed at the inputs of the gate e, i.e. $f_e = g(f_1, \ldots, f_m)$ where g is the label of e. Let $f_{e \to h}$ denote the function computed at the gate e in the modified circuit $S_{e \to h}$, i.e. $f_{e \to h} = h(f_1, \ldots, f_m)$. For $\delta \in \{0,1\}$, let W^δ be a minimal δ-motivation for e in S. Since S is optimal and $size(S_{e \to \delta}) \leq size(S) - 1$, both sets W^0 and W^1 are nonempty, and moreover

$$\forall f \in \mathcal{F} \; : \qquad S_{e \to \delta}(W^\delta) \neq f(W^\delta) \tag{1.1}$$

Case 1: `h is a strong neighbor of g, i.e. for some` $\delta \in \{0,1\}$

$$g \oplus \delta \leq h \oplus \delta \oplus 1 \tag{1.2}$$

Since W^δ is minimal, we have that $f_e(W^\delta) = g(f_1, \ldots, f_m)(W^\delta) = \delta \oplus 1$. On the other hand, by (1.2), $f_{e \to h}(W^\delta) = h(f_1, \ldots, f_m)(W^\delta) = \delta$. Thus, $S_{e \to h}(W^\delta) = S_{e \to \delta}(W^\delta)$, and by (1.1) we conclude that $S_{e \to h}$ does not represent the class $\mathcal{F}$.

Case 2: `|F| = 1 and h is a neighbor of g.`

Let $\mathcal{F} = \{f\}$. We have to prove that $S_{e \to h} \neq f$ if

$$g \oplus x_i \leq g \oplus h \tag{1.3}$$

for some $i \in \{1, \ldots, n\}$ Assume w.l.o.g. that $i = 1$.

Let S' denote the circuit obtained from S as the result of the following transformation: add a new edge connecting the output of the gate e_1 with the output of e, and delete the gate e.

Since S was optimal, we have that $S' \neq f$. Take a vector $v \in \{0,1\}^n$ for which $S'(v) \neq f(v)$. Then $f_e(v) = g(f_1(v), \ldots, f_m(v)) \neq f_1(v)$, and so by (1.3), $f_{e \to h}(v) = f_1(v)$. Thus,

$S_{e\to h}(v) = S'(v)$, and hence $S_{e\to h}(v) \neq f(v)$. This completes the proof of Theorem 1. □

The lower bounds problem is, given a Boolean function f, to estimate from below its complexity, i.e. the size of an optimal circuit computing f. In doing so, we suggest to use the strong unstability of optimal circuits.

To be more specific, let S be an optimal circuit representing a class $\mathcal{F}$. Then, by Theorem 1, for any gate e of S there exist nonempty subsets $W \subseteq \{0,1\}^n$ motivating its necessity in S. Our idea is to use these motivation-sets to extract some extra information about the structure of S. The point is that although these sets depend heavily on the circuits hardware (which is a "black box" for us), they also depend on the function computed. In this paper we will show that even the dimension $Dim(W) = \max\{|v| : v \in W\}$ where $|v| = v_1 + \ldots + v_n$ is the *weight* of v, can be useful for lower bounds arguments.

Definition: Given a set of functions $\mathcal{H}$, define the *degree* $deg[e : \mathcal{H}, \mathcal{F}, S]$ of a gate e in S as the minimum of $Dim(W)$ over $h \in \mathcal{H}$ and all h-motivations W of e in S w.r.t. $\mathcal{F}$. The *degree of optimality* $deg[S : \mathcal{H}, \mathcal{F}]$ of the whole circuit S is the maximum of $deg[e : \mathcal{H}, \mathcal{F}, S]$ over all gates e of S.

Thus, $0 \leq deg[S : \mathcal{H}, \mathcal{F}] \leq n$, and S has large degree of optimality iff the necessity of its gates is motivated by vectors of small weight.

The idea is, given a Boolean function f, to associate with it an appropriate class of functions $\mathcal{F}$ such that the size of optimal circuits representing $\mathcal{F}$ does not heavily exceed the size of circuits computing f (this is so, for example, if $f \in \mathcal{F}$), and to proceed in the following three steps.

Step 1: Using the (functional) properties of the class $\mathcal{F}$ and the unstability of optimal circuits representing $\mathcal{F}$, prove that at least one optimal circuit S_0 representing $\mathcal{F}$ has low degree of optimality.

Step 2: Using the information that S_0 computes a function in $\mathcal{F}$ with low degree of optimality, show that S_0 has some special structure.

Step 3: Using this extra information about the structure of S_0 estimate from below its size.

By the definition of $\mathcal{F}$, this lower bound on the size of S_0 directly yields a lower bound for the complexity of f.

2. The General Lower Bound

In this section we will consider Boolean formulae (i.e. fan-out 1 circuits) over the basis $\{\wedge, \vee\}$ with tight negations, i.e. each input gate e is labelled by a variable or its negation $\hat{e} \in \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$ and each inner gate is labelled by $\wedge$ or $\vee$. We will describe the structure of formulae in terms of their chains.

A *chain* of a formula F is a minimal set of its input gates (not variables !) which if we set to 1, F will compute the constant function $\equiv 1$. regardless of the values assigned to the other input gates. For a chain α, we define its *length* $\|\alpha\|$ as the number of variables x_i which appear in α positively, i.e. $x_i = \hat{e}$ for some gate $e \in \alpha$. Set also $K_\alpha = \bigwedge\{\hat{e} : e \in \alpha\}$.

A chain α is *null-chain* if $K_\alpha \equiv 0$. Notice that such chains and cuts do not effect the function computed by F since $F = \bigvee\{K_\alpha : \alpha \in chains(F)\}$.

A formula is called to be $\lambda - free$ $(0 \leq \lambda \leq n)$ if it has no null-chains of length $\leq \lambda$. All formulas are 0-free; n-free formulas have no null-chains and are called *null-chain-free* ones.

In order to demonstrate steps 2 and 3 of our approach (see Section 1), we will prove a general lower bound for λ-free formulae with $\lambda \approx \sqrt{n}$.

First, we need some notation concerning the structure of the binary n-th cube. An (upper) *cone* B^∇ of a subset $B \subseteq \{0,1\}^n$ is the set of all vectors v covering at least one vector in B, i.e. such that $v \geq u$ for some $u \in B$. Define $dim(B)$ $(Dim(B))$ to be the minimum (maximum) weight of a vector in B. For $i \geq 0$, let $\#_i(B)$ denote the maximum number of vectors in B that have at least i ones in common. Let $rank(B)$ be the minimum number r $(0 \leq r < n)$ such that no two vectors in B have r ones in common, i.e. $rank(B) = \min\{i : \#_i(B) = 1\}$. Let also $Rank(B)$ be the maximum number R $(0 \leq R \leq n)$ such that each binary vector of weight $\leq R$ covers at most one vector in B.

Now we turn to the main result of this section. Given a subset $B \subseteq \{0,1\}^n$, let $\mathcal{F} = \mathcal{F}(B)$ denote the set of all Boolean functions f *approximating* B in the following sense:

$$B \subseteq f^{-1}(1) \subseteq B^\nabla.$$

Theorem 2: *Let F be an optimal λ-free formula approximating a subset $B \subseteq \{0,1\}^n$ with the degree of optimality $deg[F : \{1\}, \mathcal{F}(B)] = d$. If $\lambda \geq Rank(B) + 1$ and*

$$d \leq 2dim(B) - 3rank(B) - 1 \tag{2.1}$$

then

$$size(F) \geq |B|.$$

Proof: For a subformula G of F, let $I(G) \subseteq \{1, \ldots, n\}$ denote the set of indexes of those variables x_i which appear in G positively, i.e. $I(G) = \{i : \hat{e} = x_i$ for some input gate e of $G\}$. Set $r = rank(B)$ and say that G is an *r-subformula* if $r \leq |I(G)| \leq 2r$.

Our plan is to associate with F a decreasing sequence of $t \leq size(F)$ subsets $B_0 = B \supseteq B_1 \supseteq \ldots \supseteq B_t = \emptyset$ and prove that $\max_{0 \leq j \leq t-1} |B_j \setminus B_{j+1}| \leq 1$. This gives the desired lower bound on the size of F.

The procedure is to repeatedly replace some r-subformulas of F by 0 and define B_j as the set of all vectors from B accepted by the formula obtained from F after j

replacement steps. Then $B_{j-1} \setminus B_j$ is the set of vectors from B which are "lost" at the j-th replacement step.

More precisely, we start with the formula F and repeatedly perform the following *reduction* operation: choose an r-subformula G of F and replace it by 0. Repeatedly perform such reductions until no more are possible. Since the number of gates decreases with each reduction, the reduction procedure terminates with a formula having no r-subformulas. Thus we obtain a sequence of formulas $F_0 = F \geq F_1 \geq \ldots \geq F_t$ and the corresponding sequence of their r-subformulas $G_0, G_1, \ldots, G_{t-1}$ where each F_{j+1} is F_j with G_j replaced by 0. Take $B_j = B \cap F_j^{-1}(1)$, $j = 0, 1, \ldots, t$. This sequence decreases, and moreover, $B_t = \emptyset$ because otherwise, we would have by (2.1) that F_t has at least $dim(B) \geq r$ positively appearing variables, and hence, the reduction procedure could be continued, which is impossible.

We will prove that

$$\forall j = 0, 1, \ldots, t-1 \qquad |B_j \setminus B_{j+1}| \leq 1. \tag{2.2}$$

To prove this, we will use the optimality of F in order to extract some additional information about the structure of subformulas G_j. Let G stand for any such subformula.

Claim: `For any` $i \in I(G)$, `each chain of` G `contains an input gate labelled by` x_i.

Before proving the claim, notice that it directly yields (2.2) and hence, the desired lower bound on the size of F. Indeed, by Claim, all the vectors in $B_j \setminus B_{j+1}$ must have at least $|I(G)| \geq r = rank(B)$ ones in common, and so by the definition of $rank(B)$, this set consists of no more than one vector.

Now let us turn to the proof of the claim. Denote $H = F_j$, $G = G_j$ and fix some $i \in I(G)$ and an input gate e of G with $\hat{e} = x_i$. By the definition of the degree of optimality d, the necessity of e in F must be motivated by some vector of weight $\leq d$, i.e. there exists a vector $w \in \{0,1\}^n$ with $|w| \leq d$ such that

$$F_{e \to 1}(w) = 1 \neq 0 = F(w) \tag{2.3}$$

and

$$w \not\geq u, \ \forall u \in B. \tag{2.4}$$

Thus, there exists a chain α of G and a chain β of $H_{G \to 1}$ such that

$$K_{(\alpha \setminus \{e\}) \cup \beta}(w) = 1 \neq 0 = K_{\alpha \cup \beta}(w) \tag{2.5}$$

and

$$\|\alpha \cup \beta\| \leq |w| + 1 \leq d + 1. \tag{2.6}$$

Now take an arbitrary chain γ of G. To finish the proof of Claim we have to prove that γ contains a gate labelled by x_i.

By (2.6),

$$\|\gamma \cup \beta\| \leq \|\gamma\| + \|\beta\| \leq 2r + d \leq Rank(B), \tag{2.7}$$

So, the chain $\gamma \cup \beta$ is non-null. Take a minimal vector u such that

$$K_{\gamma \cup \beta}(u) = F(u) = 1 \tag{2.8}$$

Consider the vector $v = (w_1, \ldots, w_{i-1}, 1, w_{i+1}, \ldots, w_n)$. By (2.5)

$$K_{\alpha \cup \beta}(v) = F(v) = 1 \tag{2.9}.$$

Since F approximates B, (2.8) and (2.9) yield $\{u, v\} \subseteq B^{\nabla}$. Moreover, each of these two vectors may cover only one vector in B because by (2.1), (2.6) and (2.7), their weights are $\leq d + 2r \leq Rank(B)$. Let u^* and v^* be the corresponding vectors in B covered by u and v, respectively. Then $|v^*| + |u^* \wedge v| \leq d + 1 + |v^* \wedge u^*|$. Since $|v^*| \geq dim(B)$ and $|u^* \wedge v| \geq |u^*| - \|\gamma\| \geq dim(B) - 2r$, we conclude, by (2.1), that

$$|v^* \wedge u^*| \geq 2dim(B) - 2r - d - 1 \geq 2dim(B) - Rank(B) - 1 \geq r.$$

Therefore $u^* = v^*$, i.e. both vectors u and v cover the same vector v^* in B. By (2.5), the chain β does not contain a gate labelled by x_i, whereas by (2.4) and (2.9) the i-th coordinate of v^* must be 1 since w differs from v exactly in this coordinate. But then the i-th coordinate of u must be also 1 since $u \geq v^*$. Since u is the minimal vector with $K_{\gamma \cup \beta}(u) = 1$, the chain γ must contain a gate labelled by x_i. This concludes the proof of the claim, and hence the proof of theorem 2. □

3. Lower Bounds for Semi-slice Functions

The $(k, l, m) - semislice$ $(k < l \leq m)$ of a Boolean function f is the function

$$f_{k,l,m} = f \wedge T_k^n \wedge \neg T_l^n \vee T_m^n$$

where T_k^n is the k-th threshold function which takes the value 1 iff at least k of its arguments are 1. (For the sake of uniformity we assume that $T_{n+1}^n \equiv 0$.) Hence the usual *k-slice* (see [2], [11], [12]) is the (k, l, m)-semislice with $m = l = k + 1$.

Although null-chains do not effect the function computed, it is known that their presence enables one to reduce the size strongly: there exist Boolean functions which have polynomial size circuits with null-chains, but have no polynomial size circuit without them ([4], [10]). On the other hand, Berkowitz in [2] made a striking observation which showed that for some Boolean functions, namely, for slice functions null-chains are in fact superfluous. The trick is, given a circuit computing a k-th slice function f, to replace each negated input gate $\neg x_i$ by a monotone circuit computing $T_k^{n-1}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. The resulting circuit is monotone (and hence, has no null-chains). One may easily verify that it also computes f. Moreover, its size increases by at most an additive factor $O(n(\log n)^2)$ ([11], [12]). Thus any lower bound of $\Omega(n^{1+\epsilon})$ on the null-chain-free circuit complexity of k-slice of f would imply that f had non-linear combinational complexity.

Unfortunately, known proof techniques for monotone circuits ([1], [8]) and for null-chain-free circuits ([4], [5]), although apparently quite general, do not seem to be applicable to slice functions. It is thus desirable to learn more about the null-chain-free complexity of slice-like functions.

Let us mention the present knowledge concerning this problem. Let $L_\lambda(f)$ and $(L_*(f))$ denote the minimum size of a λ-free (null-chain-free) formula over $\{\wedge, \vee, \neg\}$ computing f, and let B_f denote the set of all vectors in $f^{-1}(1)$ of minimal weight, i.e. the set of all 'lower ones' of f.

Theorem 3 ([4], [5]): *Let f be a $(k, k+1, m)$-semislice of some Boolean function and $B = B_f$. Then*

$$L_m(f) \geq \frac{|B|}{\#_s(B)^2},$$

where

$$s = \min\{\, k/3,\ (m-k-1)/2 \,\}.$$

Theorem 4 ([5], using [8]): *Let f be a (k, m, m)-semislice of a monotone Boolean function and $B = B_f$. Then for any $t, s \geq 1$ and $0 \leq \epsilon < 1$ we have*

$$L_m(f) \geq t^{-s/2} \min\left\{ \frac{|B|}{\#_{s/2}(B)},\ \theta 2^{t\epsilon' - s\log\sqrt{\epsilon}} \right\}$$

where

$$\theta = 1 - \epsilon n/m - |B|\epsilon^k.$$

Theorem 2 implies the following lower bound for other semislices.

Theorem 5: *Let f be a $(k, l, n+1)$-semislice of monotone Boolean function and $B = B_f$. If k and l satisfy $l \leq 2k - 3rank(B) + 1$ then $L_*(f) \geq |B|$.*

Proof: Let F be an optimal null-chain-free formula computing f. Then F approximates B since f is a semislice of a monotone Boolean function. Let $d = deg[F : \{1\}, \mathcal{F}(B)]$ be the corresponding degree of optimality of F. By Theorem 2 it is enough to verify that $d \leq l - 2$, i.e. that each (unnegated) gate e of F has a motivation for its necessity of weight $\leq l - 2$.

Since F is optimal, there must exist a vector $v \in \{0,1\}^n$ such that $F_{e\to 1}(v) = 1 \neq f(v)$. If e is labelled by x_i then $v_i = 0$. Put $v' = (v_1, \ldots, v_{i-1}, 1, v_{i+1}, \ldots, v_n)$. Then $F(v') = 1$ as F has no null-chains. But $|v'| \leq l - 1$ since F computes 0 for all vectors of weight $\geq l$. Hence $|v| = |v'| - 1 \leq l - 2$, and we are done. □

For the sake of illustration, let us consider the following monotone function of $n = k^2$ Boolean variables $\{\, x_{i,j} \,:\, 1 \leq i, j \leq k \,\}$ (here k is a prime power):

$$POLY(k,s) = \bigvee \{\, \bigwedge_{i=1}^{k} x_{i,\gamma(i)} \,:\, \gamma(z) \in GF(k)[z] \quad and \quad deg(\gamma) \leq s-1 \,\}.$$

Theorems 3-5 yield the following lower bounds for various semislices of this function.

Corollary: *Let f be a (k, l, m)-semislice of $POLY(k, s)$ with $s \leq k/3$. If l and m satisfy either one of the following three conditions:*

(i) $l = k + 1$ *and* $m - k = \Omega(s)$,

(ii) $l = m = \Omega(n^{1-o(1)})$,

(iii) $l \leq 2k - 3s - 2$

then

$$L_*(f) \geq n^{\Omega(s)}.$$

4. Conclusion and open problems

The main goal of this paper was to demonstrate how the unstability of optimal non-monotone circuits can be used to extract some extra information about their hardware and, in turn, to prove lower bounds on their size. Results presented reflect only an embryonic stage of the approach, and the reader will have no doubt to recognize other ways in which the idea can be developed.

We conclude with some open problems stipulated by the optimal versus stable phenomenon.

1. We have seen above that, for some functions f (e.g. for semi-slice ones), the necessity of each gate in an optimal formula computing f is motivated by sufficiently short vectors. What other properties of motivation sets can be extracted from the properties of f?

2. The original statement of lower bounds problem is to bound the *size* of (i.e. the number of gates in) a minimal circuit computing a given function (no matter of how the gates are connected by wires and how gates are labelled). So, the whole family of functions which can be computed by the same number of gates must be taken into account. This leads to the following relaxed version of lower bounds problem. Given a circuit over the basis $\{\&, \vee, \neg\}$, say that a Boolean function f is derivable from S if f is computed by some circuit obtained from S after some &-gates are replaced by $\vee$-gates and vice versa. For a family of functions $\mathcal{F}$, let $C(\mathcal{F})$ be the minimum size of a circuit from which all the functions in $\mathcal{F}$ are derivable. Are there families $\mathcal{F} \subseteq \mathcal{B}_n$ with $|\mathcal{F}| \leq n^{O(1)}$ and $C(\mathcal{F}) \geq n^{\omega(1)}$? That is, is it possible to prove superpolynomial lower bound for "almost explicit" Boolean functions ?

3. A function f is *singular* for a restricted class of circuits if its restricted circuit complexity is almost the same as the unrestricted one. By Berkowitz's observation slice functions are singular for monotone circuits. Does null-chain-free circuits have singular functions other than slice ones? What non-monotone Boolean functions are singular for

λ-free formulae? Are there (k, m, m)-semislice functions with $m \geq k + 2$ singular for λ-free formulae?

References

[1] A.E. Andreev, On a method for obtaining lower bounds for the complexity of individual monotone functions, *Doklady Akademii Nauk SSSR* **282** : 5 (1985), 1033-1037.

[2] S.J. Berkowittz, On some relations between monotone and non-monotone circuit complexity, *Technical Report*, Computer Science Department, University of Toronto, 1982.

[3] P.E. Dunne, Some results on replacement rules in monotone Boolean networks, *Technical Report No. 64*, University of Warwick, 1984.

[4] S.P. Jukna, Entropy of contact circuits and lower bounds on their complexity, *Theoretical Computer Science* **57** : 1 (1988), 113-129.

[5] S.P. Jukna, The effect of null-chains on the complexity of contact schemes, *Proc. of 7-th Conf. on FCT, Lecture Notes in Computer Science* **380** (1989), 246-256.

[6] R. E. Krichevskii, Complexity of contact circuits realizing a function of logical algebra, *Sov. Phys. Dokl.* **8** (1964), 770-772.

[7] K. Melhorn, Z. Galil, Monotone switching networks and Boolean matrix product, *Computing* 16 (1976), 99-111.

[8] A.A. Razborov, Lower bounds on the monotone complexity of some Boolean functions, *Doklady Akademii Nauk SSSR* **281** : 4 (1985), 798-801.

[9] M. S. Paterson, Complexity of monotone networks for Boolean matrix product, *Theoretical Computer Science* **1** (1975), 13-20.

[10] *É.* Tardos, The gap between monotone and non-monotone circuit complexity is exponential, *Combinatorica* **8** : 1 (1988), 141-142.

[11] L.G. Valiant, Negation is powerless for Boolean slice functions, *SIAM J. on Computing* **15** : 2 (1986), 531-535.

[12] I. Wegener, On the complexity of slice functions, *Theoretical Computer Science* **38** (1985), 55-68.

The Gauß Lattice Basis Reduction Algorithm Succeeds With Any Norm

Michael Kaib

Fachbereich Mathematik / Informatik, Universität Frankfurt
Postfach 11 19 32, 6000 Frankfurt am Main, West Germany

Abstract

We generalize Gauß' definition of lattice basis reduction to an arbitrary norm and analyse the generalized version of the Gauß lattice basis reduction algorithm. We can prove that the worst-case bound established in [5] for the number of iterations of the Gauß algorithm in the euclidean norm, which is known to be the best possible in that case, holds for any norm. We prove for any norm that the norm of two consecutive vectors in the algorithm at every but the first and the last iteration decreases at least by a factor 2. We lift this result to a bound for the number of iterations of $\log_{1+\sqrt{2}}(B/\lambda_2) + 1$, where B denotes the maximum of the norms of the two input vectors and λ_2 denotes the second succesive minimum in the given norm. Furthermore we give two algorithms for the maximum norm $\| \cdot \|_\infty$ and the sum norm $\| \cdot \|_1$ that determine the integral reduction coefficient for every iteration in the Gauß algorithm in $O(n \log n)$ arithmetic operations, where n is the dimension of the given vector space.

1 Introduction

For the euclidean norm it is well-known that the Gauß lattice basis reduction algorithm solves the problem of lattice basis reduction of a two-dimensional lattice in $\mathbb{R}^n$ in polynomial time. More precisely for some constant β the number of iterations is bounded by $\log_\beta(B/\lambda_2) + O(1)$ where B denotes the maximum of the norms of the two input vectors and λ_2 denotes the second succesive minimum (see Definition 4) of the lattice. An elementary approach proves that the quotient of the norms of two consecutive vectors is at least $\sqrt{2}$ and thus the choice $\beta = \sqrt{2}$ gives an upper bound. Familiar proofs that improve the size of β