Journal of Computer and System Sciences 96 (2018) 107–119

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

On the limits of gate elimination ☆

Alexander Golovnev [a,*], Edward A. Hirsch [b,c], Alexander Knop [b], Alexander S. Kulikov [b]

[a] New York University, United States of America
[b] St. Petersburg Department of Steklov Institute of Mathematics of the Russian Academy of Sciences, Russian Federation
[c] St. Petersburg State University, Russian Federation

ARTICLE INFO

Article history:
Received 27 April 2017
Received in revised form 1 January 2018
Accepted 30 April 2018
Available online 26 May 2018

Keywords:
Circuit complexity
Lower bounds
Gate elimination

ABSTRACT

Although a simple counting argument shows the existence of Boolean functions of exponential circuit complexity, proving superlinear circuit lower bounds for *explicit* functions seems to be out of reach of the current techniques. There has been a (very slow) progress in proving linear lower bounds with the latest record of $3\frac{1}{86}n - o(n)$. All known lower bounds are based on the so-called gate elimination technique. A typical gate elimination argument shows that it is possible to eliminate several gates from a circuit by making one or several substitutions to the input variables and repeats this inductively. In this paper we prove that this method cannot achieve linear bounds of cn beyond a certain constant c, where c depends only on the number of substitutions made at a single step of the induction.

1. Introduction

One of the most important and at the same time most difficult questions in theoretical computer science is proving circuit lower bounds. A binary Boolean circuit is a directed acyclic graph with nodes of in-degree either 0 or 2. Nodes of in-degree 0 are called inputs and are labeled by variables $x_1, \ldots, x_n$. Nodes of in-degree 2 are called gates and are labeled by binary Boolean functions. Some k nodes are additionally labeled as the outputs of the circuit. The output gate compute a Boolean function $\{0,1\}^n \to \{0,1\}^k$ in a natural way. The size of a circuit C is defined as the number of gates in C and is denoted by $\texttt{gates}(C)$. By $\texttt{inputs}(C)$ we denote the number of inputs of C. A circuit complexity measure μ is a function assigning each circuit a non-negative real number. In particular, $\texttt{gates}$ and $\texttt{inputs}$ are circuit complexity measures.

By $B_{n,k}$ we denote the set of all Boolean functions $f : \{0,1\}^n \to \{0,1\}^k$. B_n is a shorthand for $B_{n,1}$. For a circuit complexity measure μ and a function $f \in B_n$, by $\mu(f)$ we denote the minimum value of $\mu(C)$ over all circuits C computing f. For example, $\texttt{gates}(f)$ is the miniumum size of a circuit computing f.

By comparing the number of small size circuits with the total number 2^{2^n} of Boolean functions of n variables, one concludes that almost all such functions have circuit size at least $\Omega(\frac{2^n}{n})$. This was shown by Shannon in 1949 [36]. However we still do not have an example of a function from NP that requires circuits of superlinear size. The currently strongest known lower bound is $(3 + \frac{1}{86})n - o(n)$ [14].

☆ A preliminary version of this paper [16] appeared in the proceedings of the 41st International Symposium on Foundations of Computer Science (MFCS 2016).

* Corresponding author.
E-mail address: alexgolovnev@gmail.com (A. Golovnev).

https://doi.org/10.1016/j.jcss.2018.04.005
0022-0000/

The lack of strong lower bounds is a consequence of the lack of methods for proving lower bounds for general circuits. Practically, the only known method for proving lower bounds is the gate elimination method. We illustrate this method with a simple example. Consider the function $\mathrm{MOD}^n_{3,r} \in B_n$ which outputs 1 if and only if the sum of n input bits is congruent to r modulo 3. One can prove that $\mathtt{gates}(\mathrm{MOD}^n_{3,r}) \geq 2n-4$ for any $r \in \{0,1,2\}$ by induction on n. The base case $n \leq 2$ clearly holds. Assume that $n \geq 3$ and consider an optimal circuit C computing $\mathrm{MOD}^n_{3,r}$ and its topologically first (with respect to some topological ordering) gate G. This gate is fed by two different variables x_i and x_j (if they were the same variable, the circuit would not be optimal). A crucial observation is that it cannot be the case that the out-degrees of both x_i and x_j are equal to 1. Indeed, in this case the whole circuit would depend on x_i and x_j through the gate G only. In particular, the four ways of fixing the values of x_i and x_j would give at most two different subfunctions (corresponding to $G=0$ and $G=1$), while $\mathrm{MOD}^n_{3,r}$ has three such different subfunctions: $\mathrm{MOD}^{n-2}_{3,0}$, $\mathrm{MOD}^{n-2}_{3,1}$, and $\mathrm{MOD}^{n-2}_{3,2}$. Assume, without loss of generality, that x_i has out-degree at least 2. We then substitute $x_i \leftarrow 0$, eliminate the gates fed by x_i from the circuit and proceed by induction. The eliminated gates are those fed by x_i. After the substitution, each such gate computes either a constant or a unary function of the other input of the gate, so can be eliminated. The resulting function computes $\mathrm{MOD}^{n-1}_{3,r}$. Thus we get by induction: $\mathtt{gates}(\mathrm{MOD}^n_{3,r}) \geq \mathtt{gates}(\mathrm{MOD}^{n-1}_{3,r}) + 2 \geq (2(n-1)-4)+2 = 2n-4$. This proof was given by Schnorr in 1984 [35]. In fact, it works for a wider class of functions $Q^n_{2,3}$ containing functions that have at least three different subfunctions with respect to any two variables.

This example reveals the main idea of the gate elimination process: a lower bound is proven inductively by finding at each step an appropriate substitution that eliminates many gates from the given circuit. At the same time, using just bit-fixing substitutions is not enough for proving even stronger than $2n$ lower bounds: the class $Q^n_{2,3}$ contains, in particular, a function THR^n_2 that outputs 1 iff $\sum_{i=1}^n x_i \geq 2$ whose circuit complexity is known to be at most $2n + o(n)$ [13] (see also Theorem 2.3 in [41]). For this reason, known proofs of stronger lower bounds use various additional tricks.

- One can use amortized analysis of the number of eliminated gates. For example, one can show that at each step one can either find a substitution that eliminates 3 gates *or* a pair of consecutive substitutions, the first one eliminating 2 gates and the next one eliminating 4 gates.
- They also substitute variables not just by constants but by affine functions, quadratic functions, and even arbitrary functions of other variables.
- In order to amortize for steps that eliminate too few gates, they also use more intricate complexity measures that combine the number of gates with the number of variables or other quantities.

We give an overview of known lower bounds and used tricks in Section 2.

One can guess that the gate elimination method changes only the top of a circuit in few places and thus cannot eliminate many gates. In general, this intuition fails (it is easy to present examples where a single substitution greatly simplifies a function, in particular, every substitution to a function of the highest possible complexity $2^n/n$ (see Theorem 2.1 and below in [41]) lowers the complexity of this function almost twice as for a function of $n-1$ variables it cannot exceed $2^{n-1}/(n-1) + o(2^{n-1}/(n-1))$. However, in this paper we manage to make this intuition work: we design circuits such that no substitution (and even a constant number of substitutions) can reduce the size of the circuit by more than a constant number of gates. This, in turn, implies that one cannot prove a superlinear lower bound this way. For recently popular measures that combine the number of gates with the number of inputs we prove a stronger result: One cannot prove lower bounds beyond cn for a certain specific constant c; this constant may depend on the number m of consecutive substitutions made in one step of the induction, but does not depend on the substitutions themselves (in all modern proofs $m=1$ or 2).

The gate elimination method has also been recently used for proving average-case circuit lower bounds and upper bounds for Circuit #SAT [8,18]. The limitation result of this work also applies to this line of research, implying that gate elimination cannot lead to strong improvements on the currently known results.

The paper is organized as follows. In Section 2 we list known proofs based on gate elimination, we discuss their differences and limits. Section 3 presents several examples that lead us to the main questions of this work. This section contains main results of the paper: provable limits of the gate elimination method for various complexity measures. Section 4 contains a brief overview of the known barriers for proving circuit lower bounds. Finally, Section 5 concludes the work with open questions.

2. Known lower bounds proofs

Improving Schnorr's $2n$ lower bound proof mentioned above is already a non-trivial task. It can be the case that all variables in the given circuit feed two parity gates. In this case, substituting any variable by any constant eliminates just two gates from this circuit. In 1977, Stockmeyer [37] used the following clever trick to prove a $2.5n - \Theta(1)$ lower bound for many symmetric functions including all MOD^n_m functions for constant $m \geq 3$. The idea is to eliminate five gates by *two* consecutive substitutions. This time, instead of substituting $x_i \leftarrow c$ where $c \in \{0,1\}$ we substitute $x_i \leftarrow f, x_j \leftarrow f \oplus 1$ where f is an *arbitrary function* that does not depend on x_i and x_j. One should be careful with such substitutions as they potentially might produce a subfunction outside of the class of functions for which we are currently proving a lower bound by induction. At the same time, one can see that, for example, $\mathrm{MOD}^n_{3,0}$ function turns into $\mathrm{MOD}^{n-2}_{3,2}$ function under the