Lower Bounds for Deterministic and Nondeterministic Branching Programs

Alexander A. Razborov
Steklov Mathematical Institute
Vavilova 42, 117966, GSP–1, Moscow, USSR

Abstract

We survey lower bounds established for the complexity of computing explicitly given Boolean functions by switching-and-rectifier networks, branching programs and switching networks. We first consider the unrestricted case and then proceed to various restricted models. Among these are monotone networks, bounded-width devices*, oblivious devices and read-k times only devices.

1 Introduction

The main goal of the Boolean complexity theory is to prove lower bounds on the complexity of computing "explicitly given" Boolean functions in interesting computational models. By "explicitly given" researchers usually mean "belonging to the class NP". This is a very plausible interpretation since on the one hand this class contains the overwhelming majority of interesting Boolean functions and on the other hand it is small enough to prevent us from the necessity to take into account counting arguments. To illustrate the second point, let me remind the reader that already the class Δ_2^p, next after NP in the complexity hierarchy, contains languages whose circuit size is $\Omega(n^k)$ for any fixed $k > 0$ [22]. The proof of this fact essentially uses counting

*we will be using the word "device" to denote one of the two words "program" and "network" arguments. However, proving even an $\omega(n)$ lower bound for functions from the class NP is one of the major open problems of the Complexity Theory.

The interest paid by theoreticians to this problem is mainly caused by the fact that circuit computations in a nice way capture time in consecutive and parallel Turing computations. The model, next in importance after general Boolean circuits, should, of course, capture space limitations.

It is interesting to note that such models had been discovered well before the general complexity theory of Turing computations was developed. I am talking of switching networks and their modifications such as switching-and-rectifier networks. They already appeared in pioneering works of Shannon [36, 37] and were extensively studied in the Russian literature (see e.g. the book [17]). Some remarkable results of that period are included into this survey.

The way switching networks capture deterministic space is somewhat artificial. The branching program model which does the same in a more natural way was introduced by Masek in [28]. The attempt to incorporate nondeterminism in this model bacisallly leads to switching-and-rectifier networks so in most cases we do not need to distinguish between the latter model and nondeterministic branching programs.

The success in proving lower bounds for these three models is slightly better than for general circuits. The best known bound for NP-functions (which actually holds for a function from P) was proved by Nečiporuk in 1966 [29]. This bound is of order $\Omega\left(\frac{n^2}{(\log n)^2}\right)$ for switching networks, which immediately implies the same bound for branching programs, and (observed in [32]) of order $\Omega\left(\frac{n^{3/2}}{\log n}\right)$ for switching-and-rectifier networks. It seems however that we are still far away from proving superpolynomial lower bounds for NP-functions in either of these three models.

We can obtain even better bounds if we place various additional restrictions onto our models. The hope is that proving lower bounds in these restricted models might help to accumulate machinery for attacking the general case. It is also important that quite often proving lower bounds for restricted models involves proving lower bounds for various communication models (see e.g. [5, 12]). The latter bounds probably are of independent interest.

This paper is an attempt to survey lower bounds on the complexity of NP-functions proved for both unrestricted and restricted versions of switching-and-rectifier networks, branching programs and switching networks. I have also included a few related upper bounds and simulations. To keep the size of the presentation within reasonable limits I am bound to skip very interesting trade-off results proved for multi-output functions and confine myself to the single-output case.

The paper is organized as follows. In the section 2 I will give necessary definitions, indicate relationships between our basic models and also to other models and discuss possible modifications of these models. Section 3 contains results known for the unrestricted case. One of these results [34] uses a purely combinatorial characterization of the switching-and-rectifier network size. This characterization is given in Appendix A. Section 4 is devoted to lower bounds known for monotone networks. In section 5 we consider bounded-width branching programs and switching-and-rectifier networks. I have included here the outstanding result of Barrington [7] characterizing the power of such devices and a number of lower bounds for them. Section 6 contains results concerning oblivious programs and networks and section 7 contains results concerning read-k times only devices. I have included into the latter section results from the quite new paper [12]; more technical details connected with those results are given in Appendix B. The paper is concluded with several open problems.

2 Preliminaries

In this section we will give definitions of basic computational models considered in the paper, indicate relationships between them and also to other models, and discuss possible modifications of basic models.

A *switching-and-rectifier network* is a tuple $< G, s, t, \mu >$ where G is a directed graph (W, E) with two distinguished vertices s, t and μ is a *labelling function* which associates with *some* edges $e \in E$ their *labels* $\mu(e)$ either of the form "$x_i = 0$" or of the form "$x_i = 1$" where x_i is a variable ($1 \leq i \leq n$). Edges which do not receive any label are called *free*. The network $< G, s, t, \mu >$ computes the Boolean function $f : \{0,1\}^n \to \{0,1\}$ defined as follows: for each $u \in \{0,1\}^n$ we let $f(u) = 1$ iff there exists at least one (directed) s-t path (called an *accepting* path for u) such that all labels along this path are consistent with u. The *size* of the network $< G, s, t, \mu >$ is the total number of *labelled* edges in G. We denote the minimal possible size of a switching-and-rectifier network computing a Boolean function f by $RS(f)$.

The notion of a *switching network* is defined in the same way. The only difference is that now we consider undirected graphs G and undirected accepting paths. Note that the use of free edges does not help in switching networks since we can always contract such edges. So in what follows we assume that all edges in switching networks are labeled. Denote the corresponding size measure by $S(f)$.

It is convenient for our purposes to view branching programs as a special case of switching-and-rectifier networks. Namely, a switching-and-rectifier network $< G, s, t, \mu >$ is called a *branching program* if G is acyclic, s is a source node[†], t is a sink node, the outdegree of each non-sink node is exactly 2 and the two outedges are labeled "$x_i = 0$" and "$x_i = 1$" for some variable x_i associated with the node. The idea is that in this case we keep constructing the accepting path for a Boolean input *deterministically* since at each non-sink node we have exactly one possibility to proceed prescribed by our input. t plays the role of *accepting* node, all other sink nodes are *rejecting*. The measure corresponding to the branching program size is denoted by $BP(f)$.

We will call RS, S, BP *basic* complexity measures.

Some relations between basic measures, the circuit size $C(f)$ and the formula size $L(f)$ over $\{\neg, \wedge, \vee\}$ are summarized in the following chain of inequalities (we will sometimes use $F \preceq G$ to denote $F = O(G)$):

$$C^{1/3}(f) \preceq RS(f) \preceq S(f) \leq BP(f) \preceq L(f). \tag{1}$$

Proofs are easy (and can be found, for example, in [32]). The only known upper bound for BP, S, RS in terms of the circuit size follows from (1) and the bound $L(f) \leq \exp\left(O\left(\frac{C(f)}{\log C(f)}\right)\right)$ [31]. It is also known that

$$L(f) \leq RS(f)^{O(\log RS(f))}. \tag{2}$$

(1) implies that (2) is also true for BP and S instead of L.

We now turn to connections with Turing complexity. Branching programs, as was intended in [28], capture deterministic space. That is,

[†]We prefer to use the word "node" instead of "vertex" while talking of branching programs

$$LOGSPACE/poly = BP(poly) \quad (3)$$

where $LOGSPACE/poly$ is the nonuniform analogue of $LOGSPACE$ and $BP(poly)$ is the class of all languages having polynomial branching program size. It was not the original intention for switching-and-rectifier networks but it turned out nevertheless that they capture in the same fashion nondeterministic space. That is,

$$NLOGSPACE/poly = RS(poly).$$

How about switching networks? It follows from (1) and (3) that $LOGSPACE/poly \subseteq S(poly)$. On the other hand, "UNDIRECTED GRAPH ACCESSIBILITY" is doable in random logarithmic space [3] which implies the converse inclusion. Hence

$$LOGSPACE/poly = S(poly). \quad (4)$$

(3) and (4) imply

$$BP(f) \leq S(f)^{O(1)}. \quad (5)$$

It is worth noting that no constructive proof of (5) is known.

It follows from the discussion above that Boolean simulations (1) actually reflect class inclusions

$$P \supseteq NLOGSPACE \supseteq LOGSPACE \supseteq NC^1.$$

It is also worth noting that the famous result of Szelepsényi [40] and Immerman [20] translates to the following very interesting simulation:

$$RS(\neg f) \leq RS(f)^{O(1)}. \quad (6)$$

At the end of this section we discuss possible modifications of our basic measures.

Usually the size of branching programs is measured as the total number of *nodes*. However this changes the complexity only by the factor 2 and this is unessential for purposes of this survey.

I do not know of any reasonable modification of switching networks.

There are several ways to restrict power of switching-and-rectifier networks. The most restrictive version is what might be called *nondeterministic branching programs*. Namely, we allow in branching programs *guessing nodes* that is nodes with outdegree 2, both outedges being free. The *size* is measured as the number of nodes. Let $NBP(f)$ be the corresponding measure.

It is easy to see that

$$RS(f) \preceq NBP(f) \leq RS(f)^{O(1)}. \qquad (7)$$

Moreover, most lower bounds known on NBP hold also for RS. That is why I prefer to take RS as a basic measure. One more reason for that is contained in Appendix A.

3 Lower Bounds on Basic Measures

In this section we survey superlinear lower bounds on our basic measures RS, S, BP established for NP-functions.

First (and the best known at present) such bound goes back to Nečiporuk [29]. Let $f(X_1, \ldots, X_m)$ be a Boolean function with the set of variables partitioned into m groups. Let $s_i(f)$ be the total number of different subfunctions in X_i which can be obtained from f by fixing all variables not in X_i and let

$$N(f) \rightleftharpoons \sum_{i=1}^{m} \frac{\log s_i(f)}{1 + \log\log s_i(f)}.$$

Theorem 1 ([29]) $S(f) \geq \Omega(N(f))$ *provided f essentially depends on all its variables.*

Corollary 1 $BP(f) \geq \Omega(N(f))$ (*again, provided f essentially depends on all its variables*).

Theorem 1 allows us to get $\Omega\left(\frac{n^2}{(\log n)^2}\right)$ lower bounds on the switching network size and the branching program size for functions from P. Nećhiporuk himself [29] gave an example of such functions. The simplest example of functions for which the bound of Theorem 1 is optimal was given by P.Beame and S.Cook (unpublished):

Theorem 2 (P.Beame, S.Cook) *Let* $m \rightleftharpoons \frac{n}{2\log_2 n}$, $|X_1| = \ldots = |X_m| = 2\log_2 n$ *and*
$f_n(X_1, \ldots, X_m) = 1$ *iff there are distinct* i, j *such that* $X_i = X_j$ *(as binary words). Then for each* i, $s_i(f) \geq \exp(\Omega(n))$ *and hence*

$$N(f_n) \geq \Omega\left(\frac{n^2}{(\log n)^2}\right).$$

Pudlák [32] observed that the same method which was used by Nečiporuk for proving Theorem 1, gives for switching-and-rectifier networks the following:

Theorem 3 ([29, 32]) $RS(f) \geq \Omega(N'(f))$ *where*

$$N' \rightleftharpoons \sum_{i=1}^{m} \sqrt{\log s_i(f)}$$

and f *essentially depends on all its variables.*

For example, for the function f_n from the previous theorem we have the bound

$$RS(f_n) \geq \Omega\left(\frac{n^{3/2}}{\log n}\right).$$

These results are currently the best known lower bounds for each of the three basic measures.

The Nečiporuk technique does not apply to symmetric Boolean functions. A series of lower bounds were proved for such functions using Ramsey-like arguments. For simplicity we formulate them in terms of MAJ_n, the MAJORITY function in n variables.

Pudlák [33] proved that $BP(MAJ_n) \geq \Omega(n \log n \log n / \log n \log n \log n)$. This result was improved by Babai, Pudlák, Rödl and Szemeredi [6]:

Theorem 4 ([6]) $BP(MAJ_n) \geq \Omega(n \log n / \log n \log n)$.

Grinchuk [18] proved that $S(MAJ_n) \geq \Omega(n \log^{**} n)$. Razborov [34] gave a slightly better bound which holds for general switching-and-rectifier networks:

Theorem 5 ([34]) $RS(MAJ_n) \geq \Omega(n \cdot \log\log\log^* n)$.

The following upper bound of Lupanov [26] should be compared with theorems 4,5:

Theorem 6 ([26]) *For any sequence* (f_n) *of symmetric functions the following two bounds are true:*

a) $BP(f_n) \leq O\left(\frac{n^2}{\log n}\right)$,

b) $RS(f_n) \leq O\left(n^{3/2}\right)$.

Actually, papers [18, 34] give more. Namely, it is possible to completely characterize symmetric Boolean functions having linear size in each of basic models and moreover this characterization is the same for all three models. Let us call a sequence (f_n) of symmetric Boolean functions *almost periodic* [18] if there exists $T > 0$ (not depending on n!) such that $f_n(x_1, \ldots, x_n) = f_n(y_1, \ldots, y_n)$ whenever $T \leq \#x, \#y \leq n - T$ and $|\#x - \#y|$ is divisable by T ($\#x$ is the number of ones in x).

The following theorem for measures BP and S was proved in [18]. J.Håstad (unpublished) observed that the proof of theorem 5 also allows to extend this to switching-and-rectifier networks:

Theorem 7 ([18], [34], J.Håstad) *For any sequence* (f_n) *of symmetric Boolean functions the following are equivalent:*

- (f_n) *is almost periodic,*
- $B(f_n) \leq O(n)$ *for some basic measure* B*,*
- $B(f_n) \leq O(n)$ *for any basic measure* B*.*

As a by-product, in [34] a purely combinatorial characterization of $RS(f)$ for *any* f was obtained. Since it is probably of independent interest, we give it in Appendix A.

There is just one more technique for proving superlinear lower bounds on a basic measure in the unrestricted case. We will return to it in section 6.

4 Monotone Networks

It is not quite clear how to define monotone branching programs. But this is obvious for switching networks and switching-and-rectifier networks. We merely forbid labels of the form "$x_i = 0$". These modified measures are denoted by $S_+(f)$, $RS_+(f)$ where f is a *monotone* Boolean function.

In the outstanding paper [27] Markov determined *exactly* the switching-and-rectifier network size of all threshold functions:

Theorem 8 ([27]) *For each* $0 \leq k \leq n$*,*

$$RS_+(T_n^k) = k(n-k+1)$$

where T_n^k *outputs* 1 *at* x *iff* $\#x \geq k$*.*

This result, along with theorem 6 shows that nonmonotone labels "$x_i = 0$" can help in computing monotone symmetric functions.

Relations (1) and (2) hold also for the monotone case. Hence a number of superpolynomial lower bounds on monotone circuit and formula size proved in the last several years are also applicable to the measures S_+, RS_+. However these bounds do not take into account any specific aspect of the "switching" measures. So I do not discuss them here. An extended survey on these bounds can be found, for example, in [9].

5 Bounded-Width Devices

A switching-and-rectifier network (and, in particular, a branching program) is *leveled* iff its vertices can be assigned with nonnegative integers $l(v)$ such that $l(s) = 0$ and if $< v, v' >$ is an edge then $l(v') = l(v) + 1$. Sets $L_l \rightleftharpoons \{v \,|\, l(v) = l\}$ are called *levels* of the network. $l(t)$ is its *length*‡ and the size of the largest level is the *width*. It is easy to see that any switching-and-rectifier network or branching program can be leveled such that both length and width will not exceed, up to a constant factor, the size of the original device. In this and the next section we will be considering only leveled devices.

‡note that all vertices at levels higher than $l(t)$ can be safely removed from the device

Let $w > 0$ be a fixed integer. Denote by $RS_w(f)$ $[BP_w(f)]$ the minimal possible length of a leveled switching-and-rectifier network [branching program respectively] which computes f and has width $\leq w$.

We start this section with a couple of lower bounds which were proved for width 2 only. Borodin, Dolev, Fich and Paul [11] proved that $BP_2(MAJ_n) \geq \Omega\left(\frac{n^2}{\log n}\right)$. Yao [44] announced the following superpolynomial bound:

Theorem 9 ([44]) $BP_2(MAJ_n) \geq n^{\omega(1)}$.

Before proceeding to results which hold for *arbitrary* fixed width, let me mention some useful simulations. First, we have

$$BP_{(2^w)}(f) \leq RS_w(f). \tag{8}$$

This is proved in essentially the same way as the simulation of nondeterministic finite automata by deterministic ones (see e.g. [19]). Another easy simulation is

$$L(f) \leq BP_w(f)^{O(1)}$$

for any *fixed* w. The following very surprising result of Barrington [7] says that if $w \geq 5$ then the reverse simulation also takes place:

Theorem 10 *For any fixed $w \geq 5$, $BP_w(f) \leq L(f)^{O(1)}$.*

This should be compared with theorem 9 which in particular implies that the analogue of theorem 10 for $w = 2$ is false.

How efficient is the simulation in theorem 10? Define *Barrington's constants* $\mathcal{B}_w$ by

$$\mathcal{B}_w \rightleftharpoons \overline{\lim_f} \frac{\log_2 BP_w(f)}{D(f)}$$

where $D(f)$ is the depth of f over the basis $\{\neg, \wedge, \vee\}$ (let me remind the reader that $D(f) = \theta(\log L(f))$ [39]). The original proof of the Barrington's theorem gives $\mathcal{B}_5 \leq 2$. Cai and Lipton [13] improved this to $\mathcal{B}_5 \leq 1.811\ldots$ Cleve [15] established the following asymptotic upper bound on Barrington's constants:

Theorem 11 ([15]) $\lim_{w\to\infty} \mathcal{B}_w = 1$.

Now we survey lower bounds known on $BP_w(f)$ for *arbitrary* (but fixed) width w. By (8), they are automatically extended to the nondeterministic case and we formulate them in this more general form.

First, all lower bounds from the section 3 hold also for the bounded-width case. However it was noted in [6] that Nećhiporuk's method gives a slightly better bound for bounded width:

Theorem 12 ([29, 6]) *For any fixed w and any $f(X_1, \ldots, X_m)$ essentially depending on all its variables,*

$$RS_w(f) \geq \Omega(N''(f))$$

where

$$N''(f(X_1, \ldots, X_m)) \rightleftharpoons \sum_{i=1}^{m} \log s_i(f).$$

This gives the bound $BP_w(f_n) \geq \Omega\left(\frac{n^2}{\log n}\right)$ for f_n from the theorem 2 and this is the best bound on BP_w known for $w \geq 3$.

A series of papers was devoted to symmetric functions. As in the section 3, we try to formulate corresponding results in terms of the MAJORITY function.

Chandra, Furst and Lipton [14] showed that $RS_w(MAJ_n) \geq \Omega(n \cdot W(n))$ where $W(n)$ is the inverse of van der Waerden function. Ajtai, Babai, Hajnal, Komlos, Pudlák, Rödl, Szemeredi and Turán [1] established the bound $RS_w(f_n) \geq \Omega(n \log n / \log\log n)$ for some symmetric f_n. The following bound was independently proved by Alon, Maass [4] and Babai, Pudlák, Rödl and Szemeredi [6]:

Theorem 13 ([4, 6]) *$RS_w(MAJ_n) \geq \Omega(n \log n)$.*

An interesting general algebraic technique for obtaining superlinear lower bounds on $RS_w(f_n)$, f_n symmetric, was developed by Barrington and Straubing [8]. It allows to obtain bounds of order $\Omega(n \log\log n)$ only but can be applied to a wider class of symmetric functions.

6 Oblivious Devices

A leveled switching-and-rectifier network or branching program is called *oblivious* if all edges are labeled and moreover all edges connecting ith and $(i+1)$st levels are labeled by the same variable depending only on i. In this section we mention two lower bounds which establish trade-offs between length and width of oblivious branching programs. Again, the machinery is easily extended to the nondeterministic case and we state the results in this more general way.

Alon and Maass [4] proved the following:

Theorem 14 ([4]) *Any oblivious switching-and-rectifier network of width w computing*
MAJ_n requires length $\Omega(n \log n / \log w)$.

Note that theorem 14 implies theorem 13 since each bounded-width network can be made oblivious with at most constant blow-up in size.

Another bound for oblivious programs was proved by Babai, Nisan and Szegedy in [5]:

Theorem 15 ([5]) *For some explicit polynomially computable function f_n, any oblivious switching-and-rectifier network of width* $\exp(n^\epsilon)$ ($\epsilon > 0$ *is sufficiently small constant*) *computing f_n has length $\Omega(n(\log n)^2)$.*

It is easy to see that any *directed* switching-and-rectifier network (that is a switching-and-rectifier network whose underlying graph is acyclic) of size s can be simulated by an oblivious switching-and-rectifier network of length $\leq s$ and width $O(s)$. Hence theorem 15 implies the following bound:

Corollary 2 ([5]) *$DS(f_n) \geq \Omega(n(\log n)^2)$ where f_n is the function from theorem* 15 *and DS is the complexity measure corresponding to directed networks.*

This, surely, implies the same bound for the branching program size.

7 Read-k Times Devices

In this section we will be considering arbitrary devices, not necessarily leveled.

Read-k times programs and networks capture space limitations in computations on so-called *eraser Turing machines* which erase each input cell after a fixed number k of readings. A branching program is *read-k times* if each variable occurs at most k times along each *consistent* path going from the origin s.

What about the nondeterministic case, it is clear how to give the right definition (that is the one which captures space for nondeterministic eraser Turing machines) in terms of nondeterministic branching programs introduced in section 2. It is just the same as in the deterministic case. The right definition in terms of switching-and-rectifier networks must satisfy the analogue of (7) for read-k times devices. A short thinking leads to the following definition. A switching-and-rectifier network is *read-k times only* if each variable occurs at most k times along each almost consistent path going from the origin s where a path is called *almost consistent* iff the path obtained from it by removing the last edge is consistent.

We denote corresponding complexity measures by $BP^k(f)$ and $RS^k(f)$ respectively.

A series of lower bounds was proved for read-once only branching programs. Masek [28] and Wegener [42, Chapter 14.4] proved the following tight lower bound on the complexity of MAJORITY:

Theorem 16 ([28],[42, Chapter 14.4]) $BP^1(MAJ_n) \geq \Omega(n^2)$.

This bound should be compared with Theorem 6.

Let CLIQUE_m [$\mathrm{CLIQUE\text{-}ONLY}_m$] respectively be the Boolean function in $\frac{m(m-1)}{2}$ variables which outputs 1 at an input $(x_{11}, \ldots, x_{m-1,m})$ iff the corresponding graph contains an $[m/2]$- clique [is exactly an $[m/2]$-clique respectively]. Wegener [43] and Žák [45] independently established exponential bounds

$$BP^1(\mathrm{CLIQUE}_m) \geq \exp(\Omega(m))$$

and

$$BP^1(\mathrm{CLIQUE\text{-}ONLY}_m) \geq \exp(\Omega(m)) \qquad (9)$$

respectively. It is easy to see that $BP^2(\text{CLIQUE-ONLY}_m) \leq m^{O(1)}$. Hence (9) gives the exponential separation between BP^1 and BP^2. Dunne [16], Jukna [21] and Krause [23] applied the Wegener's method to several important functions.

Ajtai, Babai, Hajnal, Komlos, Pudlák, Rödl, Szemeredi and Turán [1] proved the truly exponential lower bound for the function $\oplus\text{CLIQUE}_{m,3}$ which outputs parity of the number of triangles in a graph:

$$BP^1(\oplus\text{CLIQUE}_{m,3}) \geq \exp(\Omega(m^2)). \tag{10}$$

An easier proof of the same bound for another function was given by Kriegel and Waack [25]. Simon and Szegedy [38] presented a technique which allowed them to view many of previous results in a natural general way. Borodin, Razborov and Smolensky [12] extended bounds of Wegener and Žák to the nondeterministic case (which is nontrivial!):

Theorem 17 ([12]) a) $RS^1(CLIQUE_m) \geq \exp(\Omega(m))$,

b) $RS^1(CLIQUE\text{-}ONLY_m) \geq \exp(\Omega(m))$.

It is worth noting that $RS^1(\neg\text{CLIQUE-ONLY}_m) \leq O(m^4)$ [12]. Hence the nonuniform analogue of $NLOGSPACE = co - NLOGSPACE$ (6) fails for read-once networks.

It was also proved in [12] that for some explicit polynomially computable function f_n the truly exponential bound

$$RS^1(f_n) \geq \exp(\Omega(n)) \tag{11}$$

holds.

No lower bounds on BP^k, RS^k which would not follow from results of the section 2 are known for a fixed $k > 1$. Krause [24] proved exponential bounds for *oblivious* read-k times branching programs with some additional restrictions. Theorem 15 of Babai, Nisan and Szegedy implies exponential lower bounds for *arbitrary oblivious* read-k times switching-and-rectifier networks. A further progress in this direction was made by Borodin, Razborov and Smolensky in [12]. Let us call a switching-and-rectifier network *syntactic read-k times* if each variable occurs at most k times along *each* path going from s. Denote by $\overline{RS^k}$ the corresponding measure. Clearly, every syntactic read-k times switching-and-rectifier network is merely read-k times which implies $\overline{RS^k}(f) \geq RS^k(f)$.

Theorem 18 ([12]) *For the same function f_n as in* (11) *and for each fixed k,*

$$\overline{RS^k}(f_n) \geq \exp(\Omega(n)).$$

Note that every read-once only switching-and-rectifier network is also syntactic read-once only switching-and-rectifier network. Hence theorem 18 implies (11). Note also that any oblivious read-k times switching-and-rectifier network is also syntactic read-k times. Hence the bound from theorem 18 holds also for arbitrary oblivious read-k times switching-and-rectifier networks.

A few technical details concerning proofs of theorems 17,18 are given in Appendix B.

8 Open Problems

The obvious open problem is to prove superpolynomial lower bounds on (at least) BP_5 for NP-functions. Below I present a list of another problems which might be easier to solve.

1. Does there exist a sequence f_n for which $NBP(f_n) \geq \omega(RS(f_n))$? (compare with (7)) A natural way to attack this problem is to show that (directed or undirected) GRAPH ACCESSIBILITY PROBLEM can not be computed by linear width linear length oblivious switching-and-rectifier networks.

2. Prove better lower bounds on $RS(f_n)$ for some polynomially computable f_n. Is it true, for example, that $RS(f_n) \geq n^{2-o(1)}$ where f_n is the function from theorem 2?

3. Prove better lower bounds on $RS(MAJ_n)$. Is it true that $RS(MAJ_n) \geq \Omega(n \cdot \log \ldots \log n)$ for a *finite* iteration of logarithms? Is it true that $RS(MAJ_n) \geq n^{1+\Omega(1)}$?

4. Separate monotone analogues of $LOGSPACE$ and $NLOGSPACE$. Equivalently, prove that $S_+(DGAP_m) \geq m^{\omega(1)}$ where $DGAP_m$ is the DIRECTED GRAPH ACCESSIBILITY PROBLEM.

5. Separate monotone analogues of $NLOGSPACE$ and P. By the monotone analogue of (2), this would follow from the separation

$$L_+(f_n) \geq C_+(f_n)^{\omega(\log(C_+(f_n)))}$$

for some f_n. However existence of such f_n does not follow from known results.

6. (M.Sipser) Is it true that for any fixed w, $RS_{+,w}(MAJ_n) \geq n^{\omega(1)}$? Since $L_+(MAJ_n) \leq n^{O(1)}$ [2, 41], that would imply that the monotone analogue of theorem 10 does not take place.

7. Is it possible to extend theorem 9 to the nondeterministic case? That is, is it true that $RS_2(MAJ_n) \geq n^{\omega(1)}$?

8. Is it true that $BP_3(MAJ_n) \geq n^{\omega(1)}$?

9. [15] Is it possible to extend theorem 11 to the case of unbalanced formulas? That is, is it true that $\lim_{w\to\infty} \mathcal{B}'_w = 1$ where

$$\mathcal{B}'_w \rightleftharpoons \overline{\lim_f} \frac{\log_2 BP_w(f)}{\log_2 L(f)}?$$

10. [13] Is it true that $\mathcal{B}_w > 1$ for each fixed w?

11. The function f_n in (11) is somewhat artificial. Is it true that

$$RS^1\left(\oplus \mathrm{CLIQUE}_{m,3}\right) \geq \exp(\Omega(m^2))?$$

12. Prove superpolynomial lower bounds on RS^k for arbitrary k without any additional restrictions.

13. (for motivation see Appendix B) Give an explicit example of $n \times n$ matrices A_n over $\mathbf{F}_2$ such that for each fixed $\epsilon > 0$ the rank of any $(\epsilon n) \times (\epsilon n)$ submatrix of A_n is $\Omega(n)$.

9 Acknowledgements

I am indebted to Allan Borodin for useful remarks on an earlier version of this paper.

Appendix

A

In this appendix we give a combinatorial characterization of $RS(f)$ for any function f [34]. This characterization is analogous to a similar characterization for the circuit size from [35].

Let $U \rightleftharpoons f^{-1}(0)$, $V \rightleftharpoons f^{-1}(1)$. Denote by $\mathcal{F}$ the set of all nontrivial monotone functionals $F : \mathcal{P}(U) \longrightarrow \{0,1\}$. Given $1 \leq i \leq n, \epsilon \in \{0,1\}, A \subseteq U$, set

$$\delta_{i,\epsilon}(A) \rightleftharpoons \left\{ (F,V) \in \mathcal{F} \times V \,\middle|\, v^i = \epsilon, F(A) = 1, F\left(A \cap X_i^\epsilon\right) = 0 \right\}.$$

Let

$$\Delta \rightleftharpoons \{ \delta_{i,\epsilon}(A) \,|\, 1 \leq i \leq n, \epsilon \in \{0,1\}, A \subseteq U \}.$$

Theorem 19 ([34]) *For each function f, $RS(f)$ equals to the minimal possible cardinality of a $\Delta_0 \subseteq \Delta$ such that $\bigcup \Delta_0 = \mathcal{F} \times V$.*

This theorem gives an extra evidence as to why RS is the most natural measure among several candidates polynomially equivalent to each other.

B

In this appendix we give a few technical details from the paper [12].

The proof of theorems 17,18 is based upon the following reduction which further simplifies and generalizes methods from [38]:

Theorem 20 ([12]) *Let f be a Boolean function in n variables; let k, a be positive integers. Let $T = \left(2\overline{RS^k}(f)\right)^{2ka}$. Then f can be represented in the form*

$$f = \bigvee_{i=1}^{T} \bigwedge_{j=1}^{ka} f_{ij}(X_{ij}) \tag{12}$$

where f_{ij} is a function depending only on variables from $X_{ij} \subseteq \{x_1, \ldots, x_n\}$, $|X_{ij}| \leq \lceil n/a \rceil$ and for any $1 \leq i \leq T$, each variable belongs to at most k of the sets $\{X_{i1}, \ldots, X_{i,ka}\}$.

In the partial case $k = 1$, $a = 2$ (and it is the case which is needed for the proof of theorem 17) theorem 20 becomes the following:

Corollary 3 *Let f be a Boolean function in n variables, n even. Then f can be represented in the form*

$$f = \bigvee_{i=1}^{T} [f_{i1}(X_{i1}) \wedge f_{i2}(X_{i2})] \tag{13}$$

where $\{X_{i1}, X_{i2}\}$ is a partition of $\{x_1, \ldots, x_n\}$ into two groups of equal size and $T \leq RS^1(f)^{O(1)}$.

Note that if the partition $\{X_{i1}, X_{i2}\}$ in (13) is independent of i, then $\log_2$ of the minimal possible T equals to the nondeterministic communication complexity in the model which allows an arbitrary partition of inputs [30].

For proving lower bounds on T in (13) we use in [12] a plain combinatorial machinery. Namely, we call any function of the form $f_{i1}(X_{i1}) \wedge f_{i2}(X_{i2})$ an *elementary rectangle* and prove that each elementary rectangle which is a subset of CLIQUE (that is may appear in the right-hand side of (13)) contains only a small fractions of ones of the CLIQUE-ONLY function.

For proving lower bounds on T when $k > 1$ we need an example of matrices over $\mathbf{F}_2$ all whose minors are of a high rank. Unfortunately we do not know such matrices (see open question 13 above). What we can prove is the following:

Theorem 21 ([12]) *If A is a submatrix with area s of a $n \times n$ Generalized Fourier Transform matrix over an arbitrary field then* $\mathrm{rank}(A) \geq \frac{s}{2n \ln\left(\frac{2n}{\sqrt{s}}\right)}$.

This theorem allows us to prove lower bounds first for the case of R-way switching-and-rectifier networks [10] when $R \geq 3$ is a power of a prime. Then they are easily extended to the Boolean case.

References

1 M. Ajtai, L. Babai, P. Hajnal, J. Komlos, P. Pudlák, V. Rödl, E. Szemeredi, and Gy. Turán. Two lower bounds for branching programs. In *Proceedings of the* 18*st ACM STOC*, pages 30–38, 1986.

2 M. Ajtai, J. Komlós, and E. Szemeredi. An $O(n \log n)$ sorting network. In *Proceedings of the* 15*st ACM STOC*, pages 1–9, 1983.

3 R. Aleluinas, R. M. Karp, R. J. Lipton, R. J. Lovász, and C. Rackoff. Random walks, universal sequences and the complexity of maze problems. In *Proceedings of the* 20*th IEEE Symposium on Foundations of Computer Science*, pages 218–223, 1979.

4 N. Alon and W. Maass. Meanders and their applications in lower bounds arguments. *Journal of Computer and System Sciences*, 37:118–129, 1988.

5 L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols and logspace-hard pseudorandom sequences. In *Proceedings of the* 21*st ACM STOC*, pages 1–11, 1989.

6 L. Babai, P. Pudlák, V. Rödl, and E. Szemeredi. Lower bounds in complexity of symmetric Boolean functions. 1988. To appear in *Theoretical Computer Science*.

7 D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In *Proceedings of the* 18*st ACM STOC*, pages 1–5, 1986.

8 D. A. Barrington and H. Straubing. Superlinear lower bounds for bounded-width branching programs. 1990. Preprint.

9 R. B. Boppana and M. Sipser. The complexity of finite functions. In Jan Van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol. A* (*Algorithms and Complexity*), chapter 14, pages 757–804, Elsevier Science Publishers B.V. and The MIT Press, 1990.

10 A. Borodin and S. Cook. A time-space tradeoff for sorting on a general sequential model of computation. *SIAM Journal on Computing*, 11(2):287–297, 1982.

11 A. Borodin, D. Dolev, F. Fich, and W. Paul. Bounds for width 2 branching programs. In *Proceedings of the* 15*st ACM STOC*, pages 87–93, 1983.

12 A. Borodin, A. Razborov, and R. Smolensky. On lower bounds for read-k times branching programs. May 1991. Submitted to *Computational Complexity.*

13 Jin-yi Cai and R. J. Lipton. Subquadratic simulations of circuits by branching programs. In *Proceedings of the* 30*st IEEE FOCS*, pages 568–573, 1989.

14 A. Chandra, M. Furst, and R. Lipton. Multiparty protocols. In *Proceedings of the* 15*th ACM STOC*, pages 94–99, 1983.

15 R. Cleve. Towards optimal simulations of formulas by bounded-width programs. In *Proceedings of the* 22*th ACM STOC*, pages 271–277, 1990.

16 P. E. Dunne. Lower bounds on the complexity of one-time-only branching programs. In *Proceedings of the FCT, Lecture Notes in Computer Science,* 199, pages 90–99, Springer-Verlag, New York/Berlin, 1985.

17 M. A. Gavrilov. *The theory of relay and switching circuits.* Nauka, 1950. In Russian.

18 M. I. Grinchuk. *On the switching network size of symmetric Boolean functions* (in Russian). Master's thesis, Moscow State University, Moscow, 1989.

19 J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, Reading, Massachussetts, 1979.

20 N. Immerman. Nondeterministic space is closed under complementation. In *Proceedings of the* 3*rd Structure in Complexity Theory Annual Conference*, pages 112–115, 1988.

21 S. Jukna. Lower bounds on the complexity of local circuits. In *Proceedings of the MFCT'*86*, Lecture Notes in Computer Science,* 233, pages 440–448, Springer-Verlag, New York/Berlin, 1986.

22 R. Kannan. A circuit size lower bound. In *Proceedings of the* 22*th IEEE Symposium on Foundations of Computer Science*, pages 304–309, 1981.

23 M. Krause. Exponential lower bounds on the complexity of local and real-time branching programs. 1986. To appear in *J. Inform. Process. Cybern.* (EIK).

24 M. Krause. Lower bounds for depth-restricted branching programs. *Infomation and Computation*, 91(1):1–14, 1991.

25 K. Kriegel and S. Waack. Lower bounds on the complexity of real-time branching programs. In *Proceedings of the FCT, Lecture Notes in Computer Science,* 278, pages 90–99, Springer-Verlag, New York/Berlin, 1987.

26 O. B. Lupanov. On computing symmetric functions of the propositional calculus by switching networks (in Russian). In *Problems of Cybernetics, vol.* 15, pages 85–100, Nauka, 1965.

27 A. A. Markov. On minimal switching-and-rectifier networks for monotone symmetric functions (in Russian). In *Problems of Cybernetics, vol.* 8, pages 117–121, Nauka, 1962.

28 W. Masek. *A fast algorithm for the string edidting problem and decision graph complexity.* Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1976.

29 E. I. Nečiporuk. On a Boolean function. *Doklady of the Academy of Sciences of the USSR*, 169(4):765–766 (in Russian), 1966. English translation in *Soviet Mathematics Doklady* 7:4, pages 999-1000.

30 C. H. Papadimitriou and M. Sipser. Communication complexity. In *Proceedings of the* 14*th ACM STOC*, pages 196–200, 1982.

31 M. S. Paterson and L. G. Valiant. Circuit size is nonlinear in depth. *Theoretical Computer Science*, 2:397–400, 1976.

32 P. Pudlák. The hierarchy of Boolean circuits. *Computers and Artificial Intelligence*, 6(5):449–468, 1987.

33 P. Pudlák. A lower bound on complexity of branching programs. In *Proceedings of the* 11*th MFCT, Lecture Notes in Computer Science,* 176, pages 480–489, Springer-Verlag, New York/Berlin, 1984.

34 A. A. Razborov. Lower bounds on the size of switching-and-rectifier networks for symmetric Boolean functions. *Mathematical Notes of the Academy of Sciences of the USSR*, 48(6):79–91, 1990.

35 A. A. Razborov. On the method of approximation. In *Proceedings of the* 21*st ACM Symposium on Theory of Computing*, pages 167–176, 1989.

36 C. Shannon. A symbolic analysis of relay and switching networks. *Transactions of American Institute of Electrical Engeneers*, 57:713–723, 1938.

37 C. Shannon. The synthesis of two-terminal switching circuits. *Bell Systems Technical Journal*, 28(1):59–98, 1949.

38 J. Simon and M. Szegedy. Lower bound techniques for read only once branching programs. November 1990. Preprint.

39 P. M. Spira. On time-hardware complexity tradeoffs for Boolean functions. In *Proceedings of the* 4*th Hawaii Symposium on System Sciences*, pages 525–527, Western Periodicals Company, North Hollywood, 1971.

40 R. Szelepcsényi. The method of forcing for nondeterministic automata. *Bulletin of the EATCS*, 33:96–99, 1987.

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

42 I. Wegener. *The complexity of Boolean functions.* Wiley-Teubner, 1987.

43 I. Wegener. On the complexity of branching programs and decision trees for clique functions. *Assoc. Comput. Math.*, 35:461–471, 1988.

44 A. Yao. Lower bounds by probabilistic arguments. In *Proceedings of the* 24*th IEEE FOCS*, pages 420–428, 1983.

45 S. Žák. An exponential lower bound for one-time-only branching programs. In *Proceedings of the* 11*th MFCT, Lecture Notes in Computer Science,* 176, pages 562–566, Springer-Verlag, New York/Berlin, 1984.