Electronic Colloquium on Computational Complexity, Report No. 18 (1998)

Randomness and Nondeterminism are Incomparable for Read-Once Branching Programs

Martin Sauerhoff*
FB Informatik, LS II, Univ. Dortmund, 44221 Dortmund, Germany

Abstract

We extend the tools for proving lower bounds for randomized branching programs by presenting a new technique for the read-once case which is applicable to a large class of functions. This technique fills the gap between simple methods only applicable for OBDDs and the well-known "rectangle technique" of Borodin, Razborov and Smolensky which works for the quite general models of nondeterministic and randomized read-k-times branching programs, but which has the drawback that it could only be applied to very special functions so far.

By an application of the new method, we resolve the remaining open problems concerning the relations of the most important probabilistic complexity classes for read-once branchings programs. We obtain that the analogues of the classes BPP and NP for read-once branching programs are incomparable and that RP is a proper subclass of NP.

Key Words: Read-once branching program, randomization, nondeterminism, communication complexity, lower bounds.

AMS subject classification: 68Q15, 94C10

1 Introduction

Branching programs, read-k-times branching programs and OBDDs (ordered binary decision diagrams) are formally defined in Section 2. For a history of complexity theoretical results for the deterministic and nondeterministic case we refer to [7], [19], [22] and [25].

Randomized branching programs are defined in analogy to probabilistic Turing machines. In spite of the fact that the complexity theoretical results for probabilistic Turing machines are still quite unsatisfactory, there has been considerable success in the analysis of combinatorially

*This work has been supported by DFG grant We 1066/8-1.

ISSN 1433-8092

simpler computation models like, e. g., communication protocols. Since there are also several restricted branching program models for which a good collection of proof techniques is available, it is natural to ask what can be done for randomized versions of these models.

The complexity theoretical analysis of randomized variants of branching programs has been launched by Ablayev and Karpinski in 1996 [2]. Since then, we can note a remarkable progress in the understanding of the randomized variants of OBDDs and of (syntactic) read-k-times branching programs. Meanwhile, there are several upper and lower bound results for randomized OBDDs, and we know the relations between most of the important complexity classes, like RP, BPP and NP (see [1], [3], [4], [20]). Agrawal and Thierauf [5] have presented results on the satisfiability problem for randomized OBDDs.

Of course, proving lower bounds for randomized read-once or even for read-k-times branching programs has turned out to be much harder than for randomized OBDDs. In [20] a lower bound technique for randomized read-k-times branching programs based on the well-known "rectangle technique" of Borodin, Razborov and Smolensky [8] for nondeterministic read-k-times branching programs has been presented.

The first applications of this technique have yielded an exponential lower bound for randomized read-k-times branching programs (also in [20]) and the result that the analogues of the classes NP and BPP for read-once branching programs are not comparable if the error allowed for the randomized model is restricted to $1/4$ (see [21] and [22]).

Thathachar [24] has managed to separate the syntactic read-k-times hierarchy by the same proof technique. He has proved an exponential gap between the size of nondeterministic or randomized read-k-times branching programs and deterministic read-$(k+1)$-times branching programs (where $k = O(\log^{1/2} n)$) for a generalized variant of the function considered in [21]. To prove such a result had been an open problem for 14 years, stated already in the seminal work of Wegener [26] on lower bounds for read-once branching programs (see also [25]).

Borodin, Razborov and Smolensky's "rectangle technique" has thus turned out to be especially fruitful in yielding results for quite general branching program models. Nevertheless, one drawback of this technique is that it only works for very special, elaborately constructed functions. In a superficial way one can say that the idea of all these constructions is to exploit the fact that the inner-product function from communication complexity theory (or some of its generalizations) is hard to compute for all important types of communication protocols.

On the other hand, we have the nice "reduction technique" for OBDDs which allows to reduce communication complexity to OBDD size directly and which has yielded the large pool of results mentioned above (see, e. g., [20] for a description of the technique for the randomized setting). But this technique does not work even for read-once branching programs, since it relies on the fixed variable ordering of the OBDD.

The intention of the present paper is to supply a new part of the overall puzzle lying in the gap between the two mentioned techniques. We extend the "reduction technique" by ideas from the technique of Borodin, Razborov and Smolensky such that it also works for read-once branching programs. As an application, we present a class of Boolean functions with the property that a certain communication problem which is proved to be hard for one-way communication protocols can be "reduced" to each member of the class by the new technique.

As a consequence, all these functions are hard for randomized read-once branching programs with arbitrary two-sided error ε, $\varepsilon < 1/2$. The considered functions are the so-called "k-stable" functions which have already been studied in the literature on read-once branching programs for a long time (see [9], [14], [15], [25]). As concrete examples, we prove that two functions considered by Jukna, Razborov, Savický and Wegener [13] have only randomized read-once branching programs of exponential size and thus affirmatively answer their question whether these functions separate the classes BPP and NP $\cap$ coNP for read-once branching programs.

The rest of the paper is organized as follows. In Section 2, we introduce the notions which are important for the following. In Section 3 we present the lower bound technique and in Section 4 its applications.

2 Preliminaries

We briefly repeat the definitions of some of the usual types of branching programs considered in the following.

Definition 1: A *branching program (BP)* on the variable set $\{x_1, \ldots, x_n\}$ is a directed acyclic graph with one source and two sinks, the latter labelled by the constants 0 and 1. Each non-sink node is labelled by a variable x_i and has exactly two outgoing edges labelled by 0 or 1. This graph represents a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ in the following way. To compute $f(a)$ for some input $a \in \{0,1\}^n$, start at the source node. For a non-sink node labelled by x_i, check the value of this variable and follow the edge which is labelled by this value (this is called a "test of variable x_i"). Iterate this until a sink node is reached. The value of f on input a is the value of the reached sink. For a fixed input a, the sequence of nodes visited in this way is uniquely determined and is called the *computation path* for a. The *size* of a branching program G is the number of its non-sink nodes and is denoted by $|G|$.

A *read-k-times branching program* is a branching program where on each path from the source to one of the sinks, each variable is allowed to be tested at most k times. For the case $k = 1$ in this definition we use the name *read-once branching program*.

An *OBDD* (ordered binary decision diagram) is a read-once branching program with an additional ordering of the variables. On each path from the source to one of the sinks, the order of the tests of variables has to be consistent with the prescribed variable ordering.

We now give the definitions of nondeterministic and randomized variants of general branching programs. It is easy to derive appropriate variants for the restricted branching program models.

Definition 2: A *randomized branching program* G syntactically is a branching program with two disjoint sets of variables $x_1, \ldots, x_n$ and $z_1, \ldots, z_r$. We will call the latter "probabilistic" variables. By the usual semantics for deterministic branching programs defined above, G represents a function g on $n + r$ variables.

Now we introduce an additional probabilistic semantics for G. We say that G as a randomized branching program represents a function $f\colon \{0,1\}^n \to \{0,1\}$ with

- *one-sided error* at most ε, $0 \leq \varepsilon < 1$, if for all $x \in \{0,1\}^n$ it holds that

  $\Pr\{g(x,z) = 0\} = 1, \quad \text{if } f(x) = 0;$
  $\Pr\{g(x,z) = 1\} \geq 1 - \varepsilon, \quad \text{if } f(x) = 1;$

- *two-sided error* at most ε, $0 \leq \varepsilon < 1/2$, if for all $x \in \{0,1\}^n$ it holds that

  $\Pr\{g(x,z) = f(x)\} \geq 1 - \varepsilon.$

In these expressions, z is an assignment to the probabilistic variables which is chosen according to the uniform distribution from $\{0,1\}^r$.

A *randomized read-k-times BP* is a randomized branching program with the restriction that on each path from the source to a sink, each variable x_i and each variable z_i is tested at most k times. For a *randomized OBDD*, an ordering on the variables $x_1, \ldots, x_n$ and $z_1, \ldots, z_r$ is given.

Definition 3: A *nondeterministic branching program* G has the same syntax as described for randomized branching programs in the previous definition. Again, let g be the function on $n + r$ variables computed by G as a deterministic branching program. Then we say that G as a nondeterministic branching program computes a function $f\colon \{0,1\}^n \to \{0,1\}$ if for all $x \in \{0,1\}^n$

$$\begin{aligned} &\Pr\{g(x,z) = 0\} = 1, \quad \text{if } f(x) = 0; \\ &\Pr\{g(x,z) = 1\} > 0, \quad \text{if } f(x) = 1; \end{aligned}$$

where z is an assignment to the probabilistic variables chosen according to the uniform distribution from $\{0,1\}^r$.

This is equivalent to other definitions of nondeterministic branching programs, e. g., that of Borodin, Razborov and Smolensky [8] and Meinel [18]. Definitions for nondeterministic read-k-times BPs and nondeterministic OBDDs are derived from this definition in the same way as done for the randomized types above.

In analogy to the well-known complexity classes for Turing machines, let RP$_\varepsilon$-BPk be the class of sequences of functions computable by polynomial size randomized read-k-times branching programs with one-sided error at most ε, $\varepsilon < 1$. Let BPP$_\varepsilon$-BPk be the class of sequences of functions computable by polynomial size randomized read-k-times branching programs with two-sided error at most ε, $\varepsilon < 1/2$. Furthermore, let

$$\text{RP-BP}k := \bigcup_{\varepsilon \in [0,1)} \text{RP}_\varepsilon\text{-BP}k, \quad \text{and} \quad \text{BPP-BP}k := \bigcup_{\varepsilon \in [0,\frac{1}{2})} \text{BPP}_\varepsilon\text{-BP}k.$$

In these definitions, ε is a constant with respect to the input size. Analogous classes can be defined for randomized OBDDs. Finally, for each of the considered complexity classes $\mathcal{C}$ let co-$\mathcal{C}$ be the class of sequences of functions (f_n) for which $(\neg f_n) \in \mathcal{C}$.

In the following, we comment on the more or less obvious relations between the complexity classes defined above. As for Turing machines, it holds that RP-BP$k \subseteq$ NP-BPk for arbitrary $k \geq 1$. We can also adapt the well-known technique of iterating probabilistic computations (called “probability amplification”) to improve the error probability of randomized branching programs and randomized OBDDs. We obtain, e. g. , that for all constant ε and ε' with $0 < \varepsilon \leq \varepsilon' < 1$ it holds that

$$\mathrm{RP}_{\varepsilon}\text{-BP} = \mathrm{RP}_{\varepsilon'}\text{-BP} \quad \text{and} \quad \mathrm{RP}_{\varepsilon}\text{-OBDD} = \mathrm{RP}_{\varepsilon'}\text{-OBDD}\,.$$

This has been proved in [20] and independently by Agrawal and Thierauf [5]. An analogous assertion for read-once branching programs does not hold [22]. Hence, it is not obvious that RP is a subclass of BPP for read-k-times branching programs. We prove this below.

Proposition 1: *For arbitrary $k \geq 1$ it holds that* RP-BPk $\subseteq$ BPP-BPk.

Proof: Let G be a randomized read-k-times BP for a function f with one-sided error at most ε, $\varepsilon < 1$. We construct a randomized read-k-times BP G' for f with two-sided error as follows. Introduce new probabilistic variables $z_1, \dots, z_r$ which are tested in a sub-program R_δ at the top of G', where $\delta \in \{i \cdot 2^{-r} \mid 0 < i < 2^r\}$. This program has two sinks labelled by δ and $1 - \delta$ reached with the respective probabilities. It is easy to see how such a program can be constructed for arbitrary values δ of the given type. The δ-sink of the program R_δ is identified with the 1-sink of G, and the $(1 - \delta)$-sink of R_δ is identified with the source node of G.

We compute the worst-case error probability of G' as a randomized branching program for f. First, let $x \in f^{-1}(0)$. Then it holds that G' computes the correct output “0” with probability $1 - \delta$, since G has one-sided error. For $x \in f^{-1}(1)$, G' computes the correct output “1” with probability at least $\delta + (1 - \delta)(1 - \varepsilon)$.

The error of G' is minimized by choosing a δ as close as possible to $\delta_{\mathrm{opt}} := \varepsilon/(1 + \varepsilon)$. Since we can construct a program R_δ for all $\delta \in \{i \cdot 2^{-r} \mid 0 < i < 2^{-r}\}$, we can ensure that $|\delta - \delta_{\mathrm{opt}}| < 2^{-r}$. The resulting randomized read-k-times branching program G' for this value of δ has error at most $\varepsilon/(1 + \varepsilon) + 2^{-r}$. Since $\varepsilon < 1$, we can ensure that this error is still at most a constant smaller than $1/2$ by choosing r large enough. $\Box$

It has already been shown in [22] that BPP-BP1 $\not\subseteq$ NP-BP1 $\cup$ coNP-BP1 and that BPP$_\varepsilon$-BP1 $\not\supseteq$ NP-BP1 for all $\varepsilon < 1/4$. In Section 3, we complete these results by showing that BPP-BP1 $\not\supseteq$ NP-BP1 $\cap$ coNP-BP1 and RP-BP1 $\subsetneq$ NP-BP1.

We mention some of the notions from communication complexity theory which will turn up in the sequel (for an introduction to this field, see, e. g., the monographs of Hromkovič [12] or Kushilevitz and Nisan [17]).

The main object of communication complexity theory is a *communication problem* described by a function $f\colon X \times Y \to \{0, 1\}$, where X and Y are finite sets. This function has to be evaluated by two cooperating players, traditionally called Alice and Bob. Alice obtains an input $x \in X$ and Bob an input $y \in Y$, and their goal is to determine $f(x, y)$ by sending messages to each other describing their input. Each player is assumed to have unlimited computational power to compute his messages. An algorithm specifying which player is the next to communicate and determining the message which this player will send given his input and the messages exchanged so far is called a *communication protocol*. The *(deterministic) communication complexity of f* is the minimal number of bits exchanged by a communication protocol by which Alice and Bob compute $f(x, y)$ for each input $(x, y) \in X \times Y$.

Many variations of this model, among them different kinds of probabilistic communication protocols, are considered in communication complexity theory. For this paper, the following probabilistic complexity measure is important.

Definition 4: The *ε-distributional complexity of* f (f as above), denoted by $D_\varepsilon(f)$, is defined as the minimum number of bits exchanged by a deterministic protocol P for f which computes the correct output only for at least an $(1-\varepsilon)$-fraction of all inputs from $X \times Y$, i. e.,

$$\frac{1}{|X||Y|} \cdot |\{(x,y) \in X \times Y \mid P(x,y) = f(x,y)\}| \geq 1 - \varepsilon,$$

where $P(x,y)$ is the output of P for $(x,y) \in X \times Y$.

Finally, we will restrict ourselves to so-called *one-way communication protocols*. In this model, Alice sends a single message to Bob who has to output the result of the protocol, which may depend on his input and the message he has obtained. We use $D_\varepsilon^{A \to B}(f)$ to denote the ε-distributional one-way complexity of f, which is the minimum number of bits exchanged by a one-way protocol with the error-restriction of Definition 4.

We conclude this section by introducing some notation concerning assignments.

Definition 5: For a set of variables X (an arbitrary finite set), let $2^X := \{a\colon X \to \{0,1\}\}$ denote the *set of assignments to* X. Let $X_1, X_2 \subseteq X$ with $X_1 \cap X_2 = \emptyset$. For $a \in X_1$, $b \in X_2$ let $a + b$ denote the *concatenation of the assignments a and b* which is the assignment $c\colon X_1 \cup X_2 \to \{0,1\}$ with $c(x) := a(x)$ if $x \in X_1$ and $c(x) := b(x)$ if $x \in X_2$. For $S \subseteq 2^{X_1}$, $T \subseteq 2^{X_2}$, define

$$S + T := \{a + b \mid a \in 2^{X_1}, b \in 2^{X_2}\} \subseteq 2^{X_1 \cup X_2}.$$

We do not distinguish between assignments and Boolean vectors (and write, e. g., $\{0,1\}^n$ instead of 2^X with $|X| = n$) if it is clear from the context which variables are concerned or if this does not matter.

3 The Lower Bound Technique

In this section, we describe the new lower bound technique. As already said in the introduction, our approach will extend the well-known "reduction technique" for proving lower bounds on the size of various OBDD variants.

The main idea of this technique is to reduce a communication problem which is known to be hard for a certain type of communication protocols to the function to be represented by the branching program (i. e., one has to show that the communication problem can be solved by using the branching program as an "oracle"). This approach has appeared in several papers in different disguises. It is used, e. g., by Babai, Nisan, and Szegedy [6] to prove time-space tradeoffs for oblivious branching programs, by Bollig, Sauerhoff, Sieling, and Wegener [7] to prove lower bounds for deterministic k-OBDDs and k-IBDDs and finally also by Ablayev [1],[4] and the author [20] to prove lower bounds for randomized OBDDs. The respective technique for deterministic OBDDs is also described in the monograph of Kushilevitz and Nisan [17].

In the following, we formally describe the considered type of reductions.

Definition 6 (CC-BP1 Reduction): Let U, V be finite sets and let a function $f\colon U \times V \to \{0,1\}$ be given (f describes a "communication problem"). Furthermore, let X be a set of variables, (X_1, X_2) a partition of X and $g\colon 2^X \to \{0,1\}$.

We call a pair of functions $\varphi_1\colon U \to 2^{X_1}$, $\varphi_2\colon V \to 2^{X_2}$ a *CC-BP1 reduction from f to g*, if for all $(u,v) \in U \times V$ it holds that

$$f(u,v) = g(\varphi_1(u) + \varphi_2(v)).$$

Such reductions can directly yield lower bound results for OBDDs. For the convenience of the reader, we describe the well-known construction for the deterministic case here. Let G be a deterministic OBDD with variable ordering π representing a function g (g as in Definition 6). Set $X_1 := \{x_{\pi(1)}, \dots, x_{\pi(k)}\}$ and $X_2 := \{x_{\pi(k+1)}, \dots, x_{\pi(n)}\}$ for a $k \in \{1, \dots, n-1\}$ chosen appropriately. Then each computation path in the graph can be split into an "upper" and a "lower" part correspoding to an assignment from 2^{X_1} and 2^{X_2}, respectively. We describe a one-way protocol which solves the communication problem f for an input $(u,v) \in U \times V$. Both players use the graph G and "their" respective function φ_1 or φ_2 from the CC-BP1 reduction. The first player (Alice) computes $a := \varphi_1(u) \in 2^{X_1}$, follows the computation path for a in G starting at the source and sends the reached node z to the second player. The second player (Bob) computes $b := \varphi_2(v) \in 2^{X_2}$, follows the computation path for b from z to one of the sinks of G and outputs its value. Obviously, the length of the messages exchanged by this protocol is at most $\lceil \log |G| \rceil$. Hence, we can transform lower bound results for one-way protocols into lower bounds for OBDDs.

This simple approach of directly reducing communication complexity to branching program size does not work, though, for read-once branching programs, since it is not clear how the computation paths in the graph can be partitioned between the two players. But we will show that we can at least construct a reduction from some weaker kind of communication complexity measure defined below.

Definition 7: Let U, V be finite sets and let $f\colon U \times V \to \{0,1\}$ be given. For a set $S \subseteq U$, $S \neq \emptyset$, let f_S be the restriction of f to $S \times V$, i. e., $f_S\colon S \times V \to \{0,1\}$, $f_S(u,v) := f(u,v)$ for $(u,v) \in S \times V$.

For $1 \leq s \leq |U|$ and ε with $0 \leq \varepsilon < 1/2$ define the *s-restricted ε-distributional one-way communication complexity of f* as

$$D_\varepsilon^{s,A\to B}(f) := \min_{S\subseteq U, |S|=s} D_\varepsilon^{A\to B}(f_S).$$

Now we have all the required notions to state the main theorem describing our proof technique for randomized read-once branching programs.

Theorem 1: *Let G be a randomized read-once branching program representing the function $g\colon 2^X \to \{0,1\}$, $|X| = n$, with two-sided error at most ε, $0 \le \varepsilon < 1/2$. Let $k \in \{1,\dots,n-1\}$ (the "partition parameter").*

Assume that there is a function $f\colon U \times V \to \{0,1\}$ (a "communication problem"), U, V finite sets and $|U| = 2^k$, such that for an arbitrary partition (X_1, X_2) of X with $|X_1| = k$ and $|X_2| = n-k$ there is a CC-BP1 reduction (φ_1, φ_2) from f to g (which may depend on the partition X_1, X_2) with the property that $\varphi_1\colon U \to 2^{X_1}$ is one-to-one and onto.

Then it holds for arbitrary ε', $\varepsilon < \varepsilon' < 1/2$, and $s := \lceil (n|G|)^{-1}(1-\varepsilon/\varepsilon') \cdot 2^k \rceil$ that

$$D_{\varepsilon'}^{s;A\to B}(f) = 0.$$

Informally, we reduce a restricted version of the communication problem f to g, where in the restricted version some inputs for the first player are forbidden. Yet the number of allowed inputs is $\Omega(2^k/(n|G|))$ and thus still "almost" the maximal number of 2^k if $n|G|$ is at most polynomially large in k. If we can show that the communication problem only becomes trivial if "many" inputs for the first player are forbidden, than we get a good bound on the size of $|G|$.

We prepare the proof of Theorem 1 by two definitions and a lemma which will be used to describe the essential sub-structures in the considered randomized read-once branching program.

Definition 8: A *partial read-once branching program* G is a read-once branching program with up to three sinks labelled by values from $\{0,1,*\}$. Let X be the variable set of G. The graph G represents an incompletely specified function $f\colon A \to \{0,1\}$, for a set $A \subseteq 2^X$, in the following way. For all $a \in A$ the computation path for a in G reaches the sink with value $f(a)$, and for all $a \in 2^X \setminus A$ the computation path for a reaches the sink with value "$*$".

The following notion has been introduced by Ablayev [1].

Definition 9: Let G be a read-once branching program with variables from X, and let (X_1, X_2) be a partition of X. G is called *weakly-ordered with respect to* (X_1, X_2) if all computation paths in G leading from the source to a sink can be decomposed into two parts, where on the first part only variables from X_1 are tested and on the second part only variables from X_2. The set which consists of the first nodes on each computation path in G starting in the source where a variable from X_2 is tested is called the *cut* of G.

The structural lemma below is the main step in the proof of our desired overall result.

Lemma 1: *Let G be randomized read-once branching program which represents the function $f\colon 2^X \to \{0,1\}$, $|X| = n$, with two-sided error at most ε, $0 \le \varepsilon < 1/2$. Let $k \in \{1,\dots,n-1\}$ be arbitrarily chosen. Furthermore, let an arbitrary ε' with $\varepsilon < \varepsilon' < 1/2$ be given.*

Then there is a partial read-once branching program G^* such that the following holds:

(1) $|G^*| \le n|G|$ and the cut of G^* has size 1;

(2) G^* is weakly-ordered with respect to a partition (X_1, X_2) of X with $|X_1| = k$ and $|X_2| = n-k$ and represents an incompletely specified function $f'\colon A \to \{0,1\}$, $A \subseteq 2^X$, with

$$|\{x \in A \mid f'(x) \ne f(x)\}| \le \varepsilon' \cdot |A|.$$

(3) It holds that $A = A' + 2^{X_2}$, *where* $A' \subseteq 2^{X_1}$ *and*

$$|A'| \geq \frac{1}{|G^*|}\left(1 - \frac{\varepsilon}{\varepsilon'}\right) 2^k.$$

Proof: The proof is in part inspired by the ideas contained in the proof of the central "structural theorem" of Borodin, Razborov and Smolensky's "rectangle technique" (Theorem 1 in [8]).

Step 1: The first step of the proof is to get rid of the probabilistic variables of the given randomized read-once branching program G. By a simple counting argument (due to Yao [27]) one can prove that there is a *deterministic* read-once branching program G' with $|G'| \leq |G|$ which represents a function $f' \colon 2^X \to \{0, 1\}$ with

$$|\{x \in 2^X \mid f'(x) \neq f(x)\}| \leq \varepsilon \cdot 2^n. \tag{$*$}$$

Step 2: Now we convert G' into a *uniform* read-once branching program G''. A read-once branching program is called uniform if for each node v on all paths from the source to v the same set of variables is tested, and if on each path from the source to one of the sinks all variables are tested. It is easy to see that the conversion can be done such that $|G''| \leq n|G'|$ and G'' still represents f' (see, e. g., [23]).

Let $v_1, \ldots, v_w$ be the nodes in G'' which are reached by paths from the source on which exactly k variables are tested. It holds that $w \leq |G''| \leq n|G|$. Using the fact that G'' is uniform, we define for each v_i the set X_i of variables tested on each path from the source to v_i. Furthermore, define R_i as the set of assignments in 2^{X_i} for which the (partial) computation path starting at the source reaches v_i. Finally, define $A_i := R_i + 2^{X \setminus X_i}$, for $i = 1, \ldots, w$. Observe that, by this construction, the sets A_i form a partition of the set of all inputs, i. e.,

$$2^X = A_1 \cup \ldots \cup A_w, \quad A_i \cap A_j = \emptyset \text{ if } i \neq j.$$

Step 3: We are now going to "restrict" the graph G'' to one of the input sets A_i constructed above, say A_{i_0}. We want a "large" A_{i_0} which additionally has the property that the fraction of inputs from A_{i_0} for which the function f' makes an error in computing f is "not much larger" than ε.

Define the fraction of inputs from the set A_i for which the wrong output is computed as

$$\varepsilon_i := \frac{|\{x \in A_i \mid f'(x) \neq f(x)\}|}{|A_i|},$$

for $i = 1, \ldots, w$.

Let ε', $\varepsilon < \varepsilon' < 1/2$, be chosen as in the hypothesis of the lemma. By ($*$) it holds that

$$\sum_{i=1}^{w} \frac{|A_i|}{2^n} \cdot \varepsilon_i \leq \varepsilon.$$

Furthermore, the A_i form a partition of 2^X. It follows by Markov's Inequality that

$$2^{-n} \cdot |\{i \mid 1 \leq i \leq w, \varepsilon_i \geq \varepsilon'\}| \leq \frac{\varepsilon}{\varepsilon'},$$

Hence, there is at least one $i_0 \in \{1, \ldots, w\}$ such that $\varepsilon_{i_0} \leq \varepsilon'$ and $|A_{i_0}| \geq (1/w)\,(1 - \varepsilon/\varepsilon') \cdot 2^n$. To complete the proof, we set $A := A_{i_0} = A' + 2^{X \setminus X_{i_0}}$, where $A' := R_{i_0}$. Let G^* be the graph obtained from G'' in the following way. Remove all nodes and edges not lying on computation paths for inputs in A. For each node which has only one successor after this process, replace the missing edge by an edge to a new sink with label "$*$". This graph G^* obviously is a partial read-once branching program which computes f' on the inputs from A. G^* is weakly-ordered with respect to $(X_{i_0}, X \setminus X_{i_0})$ and the cut of G^* consists only of the node v_{i_0}. By our calculations above, it holds that at least an $(1 - \varepsilon')$-fraction of inputs from A are computed correctly by G^*. Furthermore, we also have shown that A' is of the required size. □

Proof of Theorem 1: We apply Lemma 1 to G. We obtain a partial read-once branching program G^* which is weakly-ordered with respect to a partition (X_1, X_2) of X, $|X_1| = k$, $|X_2| = n - k$, and which represents an incompletely specified function $f'\colon A \to \{0,1\}$, where $A = A' + 2^{X_2}$, $A' \subseteq 2^{X_1}$. Let A, A' and f' have the properties described in the lemma.

Let (φ_1, φ_2) be a CC-BP1 reduction as described in the assumption of the theorem. Define $S := \varphi_1^{-1}(A') \subseteq U$. Since φ_1 is one-to-one and onto, it holds that $|S| = |A'| \geq (n|G|)^{-1}(1 + \varepsilon/\varepsilon') \cdot 2^k$. In the same way as described for OBDDs above, we can use G^* to construct a deterministic one-way communication protocol for the restricted communication problem f_S. It correctly computes f_S on at least an $(1 - \varepsilon')$-fraction of the inputs from $S \times V$ because (φ_1, φ_2) is a CC-BP1 reduction and G^* correctly computes f_A on at least an $(1 - \varepsilon')$-fraction of A. Since the cut of G^* consist only of a single node, there is exactly one message which Alice can send. Hence, the protocol can be simplified such that it uses no communication at all, Bob can compute the output only using his part of the input (the CC-BP1 reduction ensures that this protocol fulfils the error-bound). □

4 Lower Bounds for k-Stable Functions

Now we apply the lower bound method introduced in the last section to a class of functions which has been studied in the literature of lower bounds for read-once branching programs for a long time, namely the so-called "k-stable" functions.

Definition 10: Let $k \in \{1, \ldots, n-1\}$. A function $f\colon 2^X \to \{0,1\}$, $|X| = n$, is called *k-stable* if the following holds. For an arbitrary set $X_1 \subseteq X$, $|X_1| = k$, and each variable $x \in X_1$ there is an assignment $b \in 2^{X \setminus X_1}$ such that either $f(a+b) = a(x)$ for all $a \in 2^{X_1}$ or $f(a+b) = \neg a(x)$ for all $a \in 2^{X_1}$.

Lower bounds on the size of deterministic read-once branching program for k-stable functions have been proved by several authors, e. g., Dunne [9], Jukna [14], Krause [15] and Jukna, Razborov, Savický, and Wegener [13]. We list some examples from these papers.

Examples:

(1) Define the function $\mathrm{cl}_{n,k}\colon \{0,1\}^N \to \{0,1\}$, where $N := \binom{n}{2}$ and $1 \le k \le n$, on the Boolean variables $X := (x_{i,j})_{1\le i<j\le n}$. Let $G(X)$ be the undirected graph on the nodes from $\{1,\dots,n\}$ described by X, i. e., edge $\{i,j\}$ exists in $G(X)$ iff $x_{i,j} = 1$. Let $\mathrm{cl}_{n,k}(X) = 1$ iff the graph $G(X)$ contains a k-clique.

It holds that $\mathrm{cl}_{n,k}$ is s-stable for $s := \min\{\binom{k}{2} - 1, (n-k+2)/2\}$. (This can be proved easily by using the ideas contained in the works of Jukna [14] and Wegener [26]. Jukna has proved a similar result for the directed version of the clique-function, with the adjacency matrix as the input. This function is s-stable for $s := \min\{\binom{k}{2}, n-k\} - 1$.)

(2) Define $\mathrm{PM}_n, \mathrm{DET}_n\colon \{0,1\}^{n^2} \to \{0,1\}$ on an $n \times n$-matrix of Boolean variables $X := (x_{i,j})_{1\le i,j\le n}$ by

$$\mathrm{PM}_n(X) := \left[\sum_{\pi\in S_n} x_{1,\pi(1)} \cdot \ldots \cdot x_{n,\pi(n)} > 0\right], \quad \text{and}$$

$$\mathrm{DET}_n(X) := \left[\sum_{\pi\in S_n} (-1)^{\mathrm{sgn}(\pi)} \cdot x_{1,\pi(1)} \cdot \ldots \cdot x_{n,\pi(n)} \neq 0\right],$$

where the calculations within the brackets are done in $\mathbb{R}$, S_n is the permutation group of order n and the expression "$[A]$" denotes the Boolean function which is equal to 1 iff the predicate A is true. Krause [15] has proved that PM_n and DET_n are both $(n-1)$-stable.

(3) Let $n = q^2 + q + 1$. Let $P = \{1,\dots,n\}$ be the set of "points" of a projective plane of order q and let $L_1,\dots,L_n \subseteq P$ be the "lines". (Each such line contains exactly $q+1$ points, two lines intersect in exactly one point and for each point there are exactly $q+1$ lines running through this point.) A set $A \subseteq P$ is called a *blocking set* if $A \cap L_i \neq \emptyset$ for $i = 1,\dots,n$.

Define $B_n\colon \{0,1\}^n \to \{0,1\}$ by

$$B_n(x_1,\dots,x_n) := \left(\bigwedge_{1\le i\le n} \bigvee_{j\in L_i} x_j\right) \wedge \neg T^n_{q+k+1}(x_1,\dots,x_n),$$

where $k := (q+1)/2$ if q is prime, $k := \lceil\sqrt{q}\rceil$ otherwise, and $T^n_s(x_1,\dots,x_n)$ is the threshold function with output 1 iff $x_1 + \cdots + x_n \ge s$. It holds that $B_n(x_1,\dots,x_n) = 1$ iff $\{i \mid x_i = 1\}$ is a blocking set of size at most $q+k$.

The proof of the lower bound for the deterministic read-once branching program size of B_n by Jukna, Razborov, Savický and Wegener in [13] shows that B_n is k-stable.

(4) Let $n = 2^l$, and define $m := \lfloor n/l \rfloor$. We are going to define a function on the variables $x_0,\dots,x_{n-1}$, where we imagine the first $l \cdot m$ of these variables to be arranged as an $l \times m$-matrix. For $i = 0,\dots,l-1$ let $x^i := (x_{im},\dots,x_{(i+1)m-1})$ be the ith row of this matrix.

Let $\lambda\colon \{0,1\}^m \to \{0,1\}$ be an arbitrary function with the property that any assignment of constant values to at most $k \le m-1$ variables does not make λ a constant function. Define the function $\mathrm{ADDR}(\lambda)_n\colon \{0,1\}^n \to \{0,1\}$ by

$$\mathrm{ADDR}(\lambda)_n(x_0,\ldots,x_{n-1}) := x_a, \quad a := |(\lambda(x^0),\ldots,\lambda(x^{l-1}))|_2,$$

where $|x|_2$ denotes the value of a Boolean vector x interpreted as a binary representation.

It is easy to see that $\mathrm{ADDR}(\lambda)_n$ is k-stable. As a concrete example for a function λ, we can use the function λ_{SOP} defined as follows. Chop the input vector $\{0,1\}^m$ into $s := \lfloor\sqrt{m}\rfloor$ blocks of size s each. λ is defined as the disjunction of the conjunctions of all variables in each of these blocks. Then it holds that $\mathrm{ADDR}(\lambda_{\mathrm{SOP}})_n$ is $(s-1)$-stable. (See Jukna [14] and Jukna, Razborov, Savický and Wegener [13].)

We show that all the above functions are hard for randomized read-once branching programs. We are going to reduce an arbitrary k-stable function to the following well-known communication problem. Let $U := \{0,1\}^m$ and $V := \{1,\ldots,m\}$. Define $\mathrm{INDEX}_m\colon U \times V \to \{0,1\}$ by $\mathrm{INDEX}_m(u,v) := u_v$ for $(u,v) \in U \times V$. (This function describes a sort of "storage access", the input $u \in U$ represents the "memory" and the input $v \in V$ the "address" in this memory.)

Kremer, Nisan, and Ron [16] have shown that each randomized one-way protocol which computes INDEX_m with two-sided error at most $1/8$ needs $\Omega(m)$ bits of communication. In order to be able to apply Theorem 1 from the last section, we improve this result to restricted communication complexity. We prove a lower bound for the s-restricted, ε-distributional one-way communication complexity.

Lemma 2: *For arbitrary ε with $0 \le \varepsilon < 1/2$ and $s \in \{1,\ldots,2^m\}$ it holds that*

$$D_\varepsilon^{s,A\to B}(\mathrm{INDEX}_m) \ge \log s - m \cdot \mathrm{H}(\varepsilon'),$$

for arbitrary ε' with $\varepsilon < \varepsilon' < 1/2$ and $\mathrm{H}(x) := -(x\log x + (1-x)\log(1-x))$, $x \in [0,1]$.

Proof: This is a more elaborate version of the proof of Kremer, Nisan, and Ron for the "conventional" randomized one-way model where all inputs for the first player are allowed. We also use ideas from a lower bound method for one-way protocols developped by Halstenberg and Reischuk [11].

Fix a set $S \subseteq U$, $|S| = s$. We describe the communication problem INDEX by a $2^{|U|} \times 2^{|V|}$-Boolean communication matrix where all rows in $U \backslash S$ are marked as "undefined". Call this matrix M_{INDEX} in the following.

Consider a one-way communication protocol P for INDEX which computes the correct value on at least an $(1-\varepsilon)$-fraction of $S \times V$. This protocol P can be described by its computed matrix M_P, which is a $2^{|U|} \times 2^{|V|}$-Boolean matrix with entry $M_P(u,v) = c$ if P yields the output $c \in \{0,1\}$ on $(u,v) \in S \times V$, and $M_P(u,v)$ is "undefined" for all $(u,v) \in (U\backslash S) \times V$.

The total error of P is

$$\sum_{(u,v)\in S\times V} M_{\mathrm{INDEX}}(u,v) \oplus M_P(u,v) = \sum_{u\in S} d_H(u, M_P(u)) \le \varepsilon|S \times V|,$$

where d_H denotes the Hamming-distance of two Boolean vectors and $M_P(u) \in \{0,1\}^m$ the row u of M_P.

It is easy to see that P induces a partition of $S \times V$ into disjoint sets $R_1, \ldots, R_r$, where $R_i := S_i \times V$, $S_i \subseteq S$, such that whithin each such set the rows of the computed matrix M_P are identical. Our goal is to show that, in order to compute INDEX on S within the required error-bound, there has to be a "large" number of rows in S such that the relative error within these rows is not much larger than ε, i. e., can be bounded by some $\varepsilon' > \varepsilon$. After that, we show that "many" sets R_i are needed to cover these special rows in order to fulfil the given error-bound. For the rest of the proof fix ε' somehow such that $\varepsilon < \varepsilon' < 1/2$.

In the following, we again use Markov's Inequality to get a set of rows with the properties described above. For $i = 1, \ldots, r$ define

$$\varepsilon_i := \frac{1}{|R_i|} \sum_{(u,v)\in R_i} M_{\mathrm{INDEX}}(u,v) \oplus M_P(u,v),$$

the relative error of the protocol on the set R_i. For an arbitrary ε^*, $\varepsilon < \varepsilon^* < \varepsilon'$, define $I := \{i \mid \varepsilon_i \leq \varepsilon^*\}$ and $S^* := \bigcup_{i\in I} R_i$. Because of the error-bound of P and the definition of S^* it follows that

$$\varepsilon|S| \geq \sum_{i\notin I} \varepsilon_i |S_i| \geq \varepsilon'(|S| - |S^*|),$$

hence,

$$|S^*| \geq \left(1 - \frac{\varepsilon}{\varepsilon^*}\right)|S|.$$

Now we apply the above trick for a second time. Let $i_0 \in I$. For $u \in S$ define

$$\varepsilon(u) := \frac{1}{|V|} \sum_{v\in V} M_{\mathrm{INDEX}}(u,v) \oplus M_P(u,v),$$

the relative error made in row u of the computed matrix. Let $J := \{u \in S_{i_0} \mid \varepsilon(u) \leq \varepsilon'\}$. Since $i_0 \in I$, we have

$$\varepsilon^*|S_{i_0}| \geq \sum_{u\in S_{i_0}\setminus J} \varepsilon(u) \geq \varepsilon'(|S_{i_0}| - |J|),$$

and as above,

$$|J| \geq \left(1 - \frac{\varepsilon^*}{\varepsilon'}\right)|S_{i_0}|.$$

For $i \in I$ let $J(S_i)$ be the set J obtained in the above way and define $D := \bigcup_{i\in I} J(S_i)$. We have shown that

$$|D| = \sum_{i\in I} |J(S_i)| \geq \gamma|S|$$

where

$$\gamma := \left(1 - \frac{\varepsilon^*}{\varepsilon'}\right)\left(1 - \frac{\varepsilon}{\varepsilon^*}\right) > 0.$$

Finally, we argue that already a large number r of sets R_i is required in order to compute the function INDEX exactly enough for the inputs in $D \times V$.

For d with $0 \leq d \leq m$ let $N(d)$ be the maximal number of vectors in $\{0,1\}^m$ with Hamming-distance at most d from a fixed vector $x_0 \in \{0,1\}^m$, i. e.,

$$N(d) := \max_{x_0 \in \{0,1\}^m} |\{y \in \{0,1\}^m \mid d_H(x_0, y) \leq d\}| = \sum_{k=0}^{d} \binom{m}{k}.$$

To estimate the above sum, we can use the following result from [10]: For $0 < \alpha < 1/2$ it holds that

$$\sum_{k=0}^{\lfloor \alpha m \rfloor} \binom{m}{k} = 2^{m\,\mathrm{H}(\alpha) - (1/2)\log m + O(1)}.$$

Since the relative error for each row $u \in D$ is restricted to at most ε', a fixed vector $a \in \{0,1\}^m$ can approximate only a "small" number of vectors within this error-bound, namely $N(\varepsilon' m)$. Hence, also each set R_i can cover at most this many rows in the communication matrix M_{INDEX}, since all rows of the computed matrix M_P have to be identical within R_i. We get

$$r \geq \frac{|D|}{N(\varepsilon' m)} \geq \frac{\gamma |S|}{2^{m\,\mathrm{H}(\varepsilon') - (1/2)\log m + c}},$$

for some constant c, and thus $r \geq |S| / 2^{m\,\mathrm{H}(\varepsilon')}$. □

Now we are ready to prove the main result of the paper.

Theorem 2: *Let X be a set of variables, $|X| = n$, and let $f\colon 2^X \to \{0,1\}$ be k-stable, $1 \leq k \leq n-1$. Let G be a randomized read-once branching program representing f with error at most ε, $\varepsilon < 1/2$. Then it holds for arbitrary ε' with $\varepsilon < \varepsilon' < 1/2$ that*

$$|G| = \Omega\left(2^{k(1-\mathrm{H}(\varepsilon')) - \log n}\right).$$

Proof: We apply Theorem 1.

Let (X_1, X_2) be an arbitrary partition of the variables in X with $|X_1| = k$ and $|X_2| = n-k$. We construct a CC-BP1 reduction from INDEX_k to the given function f with respect to (X_1, X_2) as follows. Fix an arbitrary one-to-one and onto function $\pi\colon \{1,\ldots,k\} \to X_1$. For $u = (u_1,\ldots,u_k) \in U = \{0,1\}^k$ define $\varphi_1(u) := a \in 2^{X_1}$ where $a(x) := u_{\pi^{-1}(x)}$ for $x \in X_1$.

The crucial part is the choice of φ_2. Since f is k-stable, we have for each $x \in X_1$ an assignment $b_x \in 2^{X_2}$ such that either $f(a + b_x) = a(x)$ for all $a \in 2^{X_1}$ or $f(a + b_x) = \neg a(x)$ for all $a \in 2^{X_1}$. Let us first assume that only the first case occurs. For $v \in V = \{1,\ldots,k\}$ define $\varphi_2(v) := b_{\pi(v)}$. By this construction, we have for arbitrary $(u,v) \in U \times V$ that $f(\varphi_1(u) + \varphi_2(v)) = \mathrm{INDEX}_k(u,v)$ and thus (φ_1, φ_2) is a CC-BP1 reduction.

Fix ε' arbitrarily such that $\varepsilon < \varepsilon' < 1/2$. Choose ε^* such that $\varepsilon < \varepsilon^* < \varepsilon'$. By Theorem 1 we get

$$D_{\varepsilon^*}^{s,A\to B}(\mathrm{INDEX}_k) = 0$$

for $s := \lceil (n|G|)^{-1}(1 - \varepsilon/\varepsilon^*) \cdot 2^k \rceil$. Applying Lemma 2 yields

$$0 \geq \log s - k\,\mathrm{H}(\varepsilon') \geq k(1 - \mathrm{H}(\varepsilon')) - \log |G| - \log n - c,$$

c some constant. Solving for $|G|$ we get the claimed lower bound.

It remains to handle the case that for some $x \in X_1$, the assignment $b_x \in 2^{X_2}$ yields $f(a + b_x) = \neg a(x)$ for all $a \in 2^{X_1}$. Let $\nu\colon X_1 \to \{0, 1\}$ be the Boolean function which outputs 1 iff the above case occurs.

We slightly extend the notion of CC-BP1 reducibility as follows. To solve the given communication problem, the first player (Alice) still applies the transformation function φ_1 as described in the proof of Theorem 1. The second player (Bob) also applies φ_2 as before, but after following the computation path to a sink with value $c \in \{0, 1\}$ he outputs $c \oplus \nu(\pi(v))$ instead of c. Obviously, the output of the protocol is "corrected" in this way such that again the desired function is computed. □

As a direct consequence, we obtain for the functions defined above:

Theorem 3: $\mathrm{cl}_{n,n/2},\ \mathrm{PM}_n,\ \mathrm{DET}_n,\ B_n,\ \mathrm{ADDR}(\lambda_{\mathrm{SOP}})_n \notin$ BPP-BP1.

The result for $\mathrm{ADDR}(\lambda_{\mathrm{SOP}})_n$ is especially interesting, since Jukna, Razborov, Savický and Wegener [13] have shown that this function is contained in $\mathrm{AC}^0 \cap$ NP-BP1 $\cap$ coNP-BP1. Hence, we also have that

$$\text{BPP-BP1} \not\supseteq \mathrm{AC}^0 \cap \text{NP-BP1} \cap \text{coNP-BP1}\,.$$

Together with the earlier results we obtain that the classes BPP-BP1 and NP-BP1 are incomparable and that RP-BP1 is a proper subset of NP-BP1.

Acknowledgement

I would like to thank Ingo Wegener for helpful discussions and hints, and Detlef Sieling for encouraging remarks on the thoughts contained in this paper at an early stage.

References

[1] F. Ablayev. Randomization and nondeterminism are incomparable for polynomial ordered binary decision diagrams. In *Proc. of the 24th International Colloquium on Automata, Languages, and Programming, LNCS 1256*, 195–202. Springer-Verlag, 1997.

[2] F. Ablayev and M. Karpinski. On the power of randomized branching programs. In *Proc. of the 23th International Colloquium on Automata, Languages, and Programming, LNCS 1099*, 348 – 356. Springer-Verlag, 1996.

[3] F. Ablayev and M. Karpinski. On the power of randomized ordered branching programs. *Manuscript*, Dec. 1996.

[4] F. Ablayev and M. Karpinski. A lower bound for integer multiplication on randomized read-once branching programs. Technical Report TR98-011, Electronic Colloquium on Computational Complexity, 1998.

(This paper is only concerned with randomized *OBDDs*, in spite of the contrary suggestion by the title!).

[5] M. Agrawal and T. Thierauf. The satisfiability problem for probabilistic ordered branching programs. Technical Report TR97-060, Electronic Colloquium on Computational Complexity, 1997. To appear in *Proc. of the 13th Ann. IEEE Conf. on Computational Complexity*, 1998.

[6] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for LOGSPACE and time-space trade-offs. *Journal of Computer and System Sciences*, 45:204–232, 1992.

[7] B. Bollig, M. Sauerhoff, D. Sieling, and I. Wegener. Hierarchy theorems for kOBDDs and kIBDDs. To appear in *Theoretical Computer Science*, 1998.

[8] A. Borodin, A. A. Razborov, and R. Smolensky. On lower bounds for read-k-times-branching programs. *Computational Complexity*, 3:1–18, 1993.

[9] P. E. Dunne. Lower bounds on the complexity of 1-time-only branching programs. In *Proc. of Fundamentals of Computation Theory*, *LNCS 199*, 90–99. Springer-Verlag, 1984.

[10] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 1994.

[11] B. Halstenberg and R. Reischuk. On different modes of communication. *SIAM J. Comput.*, 22(5):913 – 934, Oct. 1993.

[12] J. Hromkovič. *Communication Complexity and Parallel Computing*. Springer-Verlag, 1997.

[13] S. Jukna, A. Razborov, P. Savický, and I. Wegener. On P versus NP $\cap$ co-NP for decision trees and read-once branching programs. In *Proc. of the 22th International Symposium on Mathematical Foundations of Computer Science*, *LNCS 1295*, 319–326. Springer-Verlag, 1997. Accepted for publication in *Computational Complexity*.

[14] S. P. Jukna. Entropy of contact circuits and lower bounds on their complexity. *Theoretical Computer Science*, 57:113 – 129, 1988.

[15] M. Krause. Exponential lower bounds on the complexity of local and real-time branching programs. *Journal of Information Processing and Cybernetics, EIK*, 24(3):99–110, 1988.

[16] I. Kremer, N. Nisan, and D. Ron. On randomized one-round communication complexity. In *Proc. of the 27th Ann. ACM Symp. on Theory of Computing*, 596 – 605, 1995.

[17] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[18] C. Meinel. The power of polynomial size Ω-branching programs. In *Proc. of the 5th Ann. Symp. on Theoretical Aspects of Computer Science*, *LNCS 294*, 81–90. Springer-Verlag, 1988.

[19] A. A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In *Proc. of Fundamentals of Computation Theory*, *LNCS 529*, 47–60. Springer-Verlag, 1991.

[20] M. Sauerhoff. A lower bound for randomized read-k-times branching programs. Technical Report TR97-019, Electronic Colloquium on Computational Complexity, 1997.

[21] M. Sauerhoff. On non-determinism versus randomness for read-once branching programs. Technical Report TR97-030, Electronic Colloquium on Computational Complexity, 1997.

[22] M. Sauerhoff. Lower bounds for randomized read-k-times branching programs. In *Proc. of the 15th Ann. Symp. on Theoretical Aspects of Computer Science*, *LNCS 1373*, 105 – 115. Springer-Verlag, 1998.

[23] D. Sieling and I. Wegener. Graph driven BDD's — a new data structure for Boolean functions. *Theoretical Computer Science*, 141:283 – 310, 1995.

[24] J. Thathachar. On separating the read-k-times branching program hierarchy. Technical Report TR98-002, Electronic Colloquium on Computational Complexity, 1998. To appear in *Proc. of STOC '98*.

[25] I. Wegener. *The Complexity of Boolean Functions*. Wiley-Teubner Series in Computer Science. Wiley-Teubner, 1987.

[26] I. Wegener. On the complexity of branching programs and decision trees for clique functions. *J. ACM*, 35(2):461–471, Apr. 1988.

[27] A. C. Yao. Lower bounds by probabilistic arguments. In *Proc. of the 24th IEEE Symp. on Foundations of Computer Science*, 420 – 428, 1983.

All technical reports from *Electronic Colloquium on Computational Complexity* are available via Internet at `http://www.eccc.uni-trier.de/`.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'