Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT

ACADEMIC PRESS

Journal of Computer and System Sciences 66 (2003) 473–495

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

http://www.elsevier.com/locate/jcss

Guess-and-verify versus unrestricted nondeterminism for OBDDs and one-way Turing machines ☆

Martin Sauerhoff[1]

FB Informatik, LS 2, Universitat Dortmund, 44221 Dortmund, Germany

Received 12 November 2001; revised 4 July 2002

Abstract

It is well known that a nondeterministic Turing machine can be simulated in polynomial time by a so-called *guess-and-verify* machine. It is an open question whether an analogous simulation exists in the context of space-bounded computation. In this paper, a negative answer to this question is given for ordered binary decision diagrams (OBDDs) and one-way Turing machines. If it is required that all nondeterministic guesses occur at the beginning of the computation, this can blow up the space complexity exponentially in the input length for these models. This is a consequence of the following main result of the paper. There is a sequence of boolean functions $f_n : \{0,1\}^n \to \{0,1\}$ such that f_n has nondeterministic OBDDs of polynomial size that use at most $(1/3) \cdot (n/3)^{1/3} \log n \cdot (1 + o(1))$ nondeterministic guesses for each computation, but f_n already requires exponential size if only at most $(1 - \varepsilon) \cdot (1/3) \cdot (n/3)^{1/3} \log n$ nondeterministic guesses may be used, where $\varepsilon > 0$ is an arbitrarily small constant.

Keywords: OBDDs; Branching programs; One-way Turing machines; Nondeterminism; Guess-and-verify; Space complexity; Lower bounds

1. Introduction and definitions

So far, there are only few models of computation for which it has been possible to analyze the power of nondeterminism and randomness. Apart from the obvious question whether or not

☆ A preliminary version of this work has appeared in Proceedings of FST & TCS ’99, Lecture Notes in Computer Science, Vol. 1738, Springer, Berlin, 1999, pp. 342–355.

E-mail address: sauerhof@ls2.cs.uni-dortmund.de.

[1] This work has been supported by DFG Grant We 1066/9.

doi:10.1016/S0022-0000(03)00037-0

nondeterminism or randomization helps at all to decrease the complexity of problems, we may also be interested in the following, more sophisticated questions:

- How much nondeterminism or randomness is required to exploit the full power of the respective model of computation? Is there a general upper bound on the amount of these resources which we can make use of?
- How does the complexity of concrete problems depend on the amount of available nondeterminism or randomness?

For Turing machines, we even do not know whether nondeterminism or randomness helps at all to solve more problems in polynomial time, and we seem to be far away from answers to questions of the above type. On the other hand, nondeterminism and randomness are well understood, e.g., in the context of two-party communication complexity. It is a challenging task to fill the gap in our knowledge between the latter model and the more "complicated" ones.

In this paper, we consider space-bounded models of computation in the nonuniform as well as the uniform setting. Besides boolean circuits and formulae, branching programs are one of the standard nonuniform models of computation.

Definition 1.

- A (*deterministic*) *branching program* (*BP*) *on the set of input variables* $X = \{x_1, \dots, x_n\}$ is a directed acyclic graph with one source and two sinks. The sinks are labeled by 0 and 1 and the interior nodes are labeled by variables from X. Interior nodes have two outgoing edges also labeled by 0 and 1, resp. This graph represents a boolean function $f : \{0,1\}^n \to \{0,1\}$ on X as follows. In order to evaluate f for a given assignment $a = (a_1, \dots, a_n) \in \{0,1\}^n$ to the input variables, one follows a path starting at the source. At an interior node labeled by x_i, the path continues with the edge labeled by a_i (this is called a *test* of the variable x_i). The output for a is the label of the sink reached in this way.
- A *nondeterministic branching program* is syntactically a deterministic branching program on two disjoint sets of variables X and $Y = \{y_1, \dots, y_r\}$. The variables from these sets are called *decision variables* and *nondeterministic variables*, resp. Nodes labeled by the two types of variables are called *decision nodes* and *nondeterministic nodes*, resp. On each path from the source to one of the sinks, each nondeterministic variable is allowed to appear at most once. A nondeterministic branching program computes the output 1 on an assignment a to the variables in X iff there is an assignment b to the nondeterministic variables such that the 1-sink is reached for the path belonging to the complete assignment consisting of a and b. Such a path to the 1-sink is called an *accepting path*.
- The *size of a deterministic or nondeterministic branching program G*, denoted by $|G|$, is the number of nodes in G. The (*nondeterministic*) *branching program size of f* is the minimum size of a deterministic (nondeterministic) branching program representing f.

The size of nondeterministic BPs as defined above is polynomially related to that according to the well-known alternative definition using unlabeled nondeterministic nodes and assuming that the successor of such a node is guessed nondeterministically [18], as well as to the size of switching-and-rectifier networks [22]. A natural measure for the amount of nondeterminism consumed by a nondeterministic BP is obtained as follows.

Definition 2. The (*worst-case*) *number of nondeterministic guesses* of a nondeterministic BP G is defined as the maximal number of nondeterministic nodes on a path from the source to one of the sinks.

Observe that the minimal number of nondeterministic variables required to label all nondeterministic nodes in a BP such that the read-once property for these variables is satisfied is identical to the maximal number of nondeterministic guesses on a path from the source to one of the sinks. Hence, the number of nondeterministic variables is also a measure for the amount of nondeterminism used by the BP.

For a history of results on BPs, we refer to the monograph [25] of Wegener. We only mention that a tight connection between the complexity theory of BPs and that of Turing machines is established by the well-known fact that the logarithm of the size complexity of BPs is essentially equal to the space complexity for the nonuniform (advice taking) variant of Turing machines. This is true for the deterministic as well as for the nondeterministic models.

Proving superpolynomial lower bounds on the size of BPs for explicitly defined boolean functions is a fundamental open problem, even in the deterministic case. Nevertheless, several interesting restricted types of BPs could be analyzed quite successfully, and for some of these models even exponential lower bounds could be proven. The goal in complexity theory is to provide proof methods for more and more general types of BPs, and on the other hand, to extend these methods also to the nondeterministic or randomized case.

Here we deal with ordered binary decision diagrams (OBDDs) that are one of the restricted types of BPs whose structure is especially well understood in the deterministic case.

Definition 3. Let π be a permutation of $\{1, \dots, n\}$. A π-*OBDD on* $X = \{x_1, \dots, x_n\}$ is a BP with the restriction that the order of variables on each path from the source to one of the sinks is consistent with π, i.e., the list of these variables is a sublist of $x_{\pi(1)}, \dots, x_{\pi(n)}$. We call a graph an *OBDD* if it is a π-OBDD for some permutation π. The permutation π is called the *variable order* of the OBDD.

OBDDs have been invented as a data structure for the representation of boolean functions and have proven to be very useful in various fields of application. For many applications it is crucial that one can work with OBDDs of small size for the functions that have to be represented. Hence, lower and upper bounds on the size of OBDDs are also of practical relevance. There are two nondeterministic variants of the OBDD model that we consider in this paper.

Definition 4. Let G be a nondeterministic BP on variables from $X \cup Y$, where $Y = \{y_1, \dots, y_r\}$ contains the nondeterministic variables. Let π be a permutation of $\{1, \dots, n\}$. We call G a *nondeterministic* π-*OBDD* if the order of the X-variables on all paths from the source to one of the sinks in G is consistent with π. If there is a permutation π' of $\{1, \dots, n + r\}$ such that G is syntactically a π'-OBDD on $X \cup Y$ (according to Definition 3), then we call G a *synchronous nondeterministic* π'-*OBDD*.

We only remark that there are also nondeterministic variants of OBDDs that have been proposed for practical use. Since we want to study OBDDs as a model of computation, it is important that nondeterminism may be used in the model in the same way as by Turing machines. We consider the following variants of the standard Turing machine (TM). In this paper, all TMs are equipped with a read-only input tape and a single work tape.

Definition 5.

- A *nonuniform Turing machine* (also called *advice-taking Turing machine*) is a TM with an additional read-only tape, called *advice tape*. On input x, this tape is automatically loaded with an *advice string* $a(|x|) \in \{0,1\}^*$, where $a : \mathbb{N} \to \{0,1\}^*$ is an arbitrary function. The space used for input x is the sum of the space required on the work tape and $\lceil \log|a(|x|)| \rceil$. A nonuniform Turing machine computes a sequence of functions $f_n : \{0,1\}^n \to \{0,1\}$ for $n \in \mathbb{N}$.
- A *one-way Turing machine* moves its input head to the right after reading a cell on the input tape. The input is terminated by a special end marker.
- For $n \in \mathbb{N}$, let π_n be a permutation of $\{1, \ldots, n\}$. A *Turing machine with input orders* $(\pi_n)_{n\in\mathbb{N}}$ is a one-way TM that, prior to its computation, permutes an input string of length n on its input tape according to the order π_n without resource consumption. An *ordered Turing machine* is a TM that works as described for some sequence of input orders.

A one-way or ordered TM with space bound $s(n)$, n the input length, has a work tape of length $s(n)$ that is delimited by start and end markers.

Nondeterministic variants of the above models are defined by allowing transitions to two successor configurations, where such a transition does not depend on the symbol under the input head and does not move this head.

Observe that one-way TMs are special ordered TMs. Nondeterministic OBDDs and the nonuniform variant of nondeterministic ordered TMs are closely related. We obtain the following simulation result, which is proven analogously to the relationship between BPs and unrestricted nonuniform TMs (see, e.g., [25]).

Proposition 6.

(1) *Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of nondeterministic OBDDs with variable orders $(\pi_n)_{n\in\mathbb{N}}$, size $s(n) = \Omega(n)$, and $r(n)$ nondeterministic variables such that each G_n represents a function $f_n : \{0,1\}^n \to \{0,1\}$. Then there is a nonuniform nondeterministic TM with input orders $(\pi_n)_{n\in\mathbb{N}}$ that computes the sequence of functions $(f_n)_{n\in\mathbb{N}}$, runs in space $O(\log s(n))$, and takes at most $r(n)$ nondeterministic steps during each computation.*

(2) *Let a nonuniform nondeterministic TM with input orders $(\pi_n)_{n\in\mathbb{N}}$ be given that runs in space $s(n) = \Omega(\log n)$, takes at most $r(n)$ nondeterministic steps during each computation, and computes the sequence of functions $(f_n)_{n\in\mathbb{N}}$. Then there is a sequence of nondeterministic OBDDs $(G_n)_{n\in\mathbb{N}}$ with variable orders $(\pi_n)_{n\in\mathbb{N}}$ such that G_n represents f_n and has size $2^{O(s(n))}$ and at most $r(n)$ nondeterministic variables.*

This proposition justifies the definition of nondeterministic OBDDs and allows to transfer complexity results for OBDDs to ordered TMs and vice versa.

2. Previous and new results

Previous results for branching programs. Several exponential lower bounds on the size of deterministic, nondeterministic, and randomized variants of OBDDs have been proven by exploiting corresponding results on one-way communication complexity [1,5,11,15,23,24]. In this situation where the standard problem of proving exponential lower bounds on the size of OBDDs can be considered solved, we can dare to tackle the more challenging questions concerning the resources nondeterminism and randomness mentioned at the beginning of the introduction.

The dependence of the OBDD size on the resource randomness has been dealt with to some extent by the author [23]. Analogous to a well-known result of Newman [19] for randomized communication complexity, it could be shown that a logarithmic number of random bits is always sufficient and for some functions also necessary to exploit the full power of randomness for OBDDs.

What do we know about the resource nondeterminism? In the context of communication complexity theory, Hromkovič and Schnitger [9] have proven that, contrary to the randomized case, imposing a logarithmic bound on the number of nondeterministic bits restricts the computational power of the model. An asymptotically exact tradeoff between one-way communication complexity and the number of nondeterministic bits has been proven by Hromkovič and the author [10]. Klauck [14] has established a round hierarchy in the case where the available amount of nondeterminism is limited. This is contrary to the situation for unlimited nondeterminism, where each nondeterministic protocol can be simulated by a nondeterministic one-way protocol.

These results lead to the conjecture that there should be sequences of functions that have nondeterministic OBDDs of polynomial size, but that require exponential size if the amount of nondeterminism, measured in the number of nondeterministic variables, is logarithmically bounded in the input length. But although lower bounds on the size of nondeterministic OBDDs can immediately be obtained by using the results on one-way communication complexity, the upper bounds do not carry over. Proving a tradeoff between size and nondeterminism turns out to be a difficult task even for OBDDs. A first result of this kind has been presented in [10]. The sequence of functions considered there can be represented by (unrestricted) nondeterministic OBDDs of size $(n/\log n)^{O(\log n)}$ in the input length n, while nondeterministic OBDDs with $O(\log n)$ nondeterministic variables require size $2^{\Omega(n)}$. So far, it has been open to obtain a similar result with a polynomial upper bound for unrestricted nondeterministic OBDDs.

Previous results for Turing machines. The dependence of complexity on the available amount of randomness or nondeterminism has also been studied for some time in the uniform setting. In the absence of sufficiently strong lower bound methods for the standard model of polynomial-time TMs, one has resorted to prove relativized separations or collapses of complexity classes, completeness results, or results under yet unproven conjectures. Of the first type is the seminal work of Kintala and Fischer [12,13] who have introduced complexity classes between P and NP defined in terms of nondeterministic polynomial-time TMs using limited amounts of nondeterminism. Several subsequent papers have continued this line of research later on, see, e.g., [2,4,6,7,21].

Here we are concerned with space as the primary resource and look at subclasses of L and NL, the classes of languages recognizable by deterministic and nondeterministic TMs, resp., with logarithmic space bound (“logspace” for short in the following). Because of Proposition 6, we are interested in one-way logspace TMs that have been studied to some extent due to their relationship to some important open problems in formal language theory and complexity theory. Hartmanis and Mahaney [8] have investigated the classes 1L and 1NL of languages recognizable by one-way logspace TMs and nondeterministic one-way logspace TMs, resp. They have shown that $1\mathrm{L} \neq 1\mathrm{NL}$ (by looking at a uniform variant of the string non-equality problem from communication complexity theory) and have defined a natural complete problem for 1NL under one-way logspace reductions. Furthermore, they have proven that $1\mathrm{NL} \subseteq \mathrm{L}$ iff $\mathrm{L} = \mathrm{NL}$. Finally, they have introduced the class ENL of all languages recognizable by nondeterministic TMs that may use nondeterministic moves only after reading the complete input (“nondeterminism at the end”) and have shown that NL has unary complete languages (with respect to logspace reductions) iff $\mathrm{ENL} = \mathrm{NL}$. (In fact, Hartmanis and Mahaney used the name “RNL” for the class called “ENL” here.)

The famous open “LBA problem” is the question of whether the context-sensitive languages coincide with those languages recognizable by deterministic linear-bounded automata. Niepel [20] has introduced the one-way variant 1ENL of the class ENL and has exploited the ideas of Hartmanis and Mahaney to show that the statements $1\mathrm{ENL} = 1\mathrm{L}$ and $\mathrm{ENL} = \mathrm{L}$ are both equivalent to an affirmative answer to the LBA problem.

It is well known that there is a polynomial-time simulation of arbitrary nondeterministic TMs by so-called *guess-and-verify* machines that guess all nondeterministic bits in advance and store them in a special location. It is no longer obvious how this can be done in the context of space-bounded computation. Niepel [20] has investigated the classes BNL and 1BNL of languages recognizable by nondeterministic logspace TMs and nondeterministic one-way logspace TMs, resp., that only use nondeterministic moves before reading their input (“nondeterminism at the beginning”), i.e., they work in the guess-and-verify mode. She has proven that $1\mathrm{ENL} \subseteq 1\mathrm{BNL}$ and $\mathrm{ENL} \subseteq \mathrm{BNL}$. Altogether, this gives us the following relations among the classes for logspace TMs mentioned so far:

$$1\mathrm{L} \subseteq 1\mathrm{ENL} \subseteq 1\mathrm{BNL} \subseteq 1\mathrm{NL}, \quad \text{and} \quad \mathrm{L} \subseteq \mathrm{ENL} \subseteq \mathrm{BNL} \subseteq \mathrm{NL}.$$

It is easy to see that $1\mathrm{L} \subsetneq 1\mathrm{BNL}$ (consider again the string non-equality problem). Hence, at least one of the first two inclusions in the first chain of inclusions is proper, but to the best of our knowledge, this has not yet been proven for either of them. It has also been open so far whether $1\mathrm{BNL} \neq 1\mathrm{NL}$, i.e., whether forcing the machines to work in the guess-and-verify mode really restricts the power of nondeterministic one-way TMs.

Summary of new results. We present a sequence of functions that have synchronous nondeterministic OBDDs of polynomial size with $(1/3) \cdot (n/3)^{1/3} \log n \cdot (1 + o(1))$ nondeterministic variables (where n is the input length), but that require exponential size in the synchronous model if only at most $(1 - \varepsilon) \cdot (1/3) \cdot (n/3)^{1/3} \log n$ nondeterministic variables may be used, where $\varepsilon > 0$ is an arbitrarily small constant. As a corollary, we obtain that even general nondeterministic

OBDDs that may use only at most $O(\log n)$ nondeterministic variables have exponential size for the considered functions. Furthermore, this function also shows that requiring all nondeterministic variables to be tested at the top of a general nondeterministic OBDD may blow up the size exponentially.

For the proof of the tradeoff result, it is important to consider synchronous nondeterministic OBDDs. But it is not obvious whether the general type of nondeterminism is really more powerful. We show that the OBDD sizes for the two models are polynomially related if the number of nondeterministic variables is not limited, but that the size for the synchronous model may be exponentially larger if such a limit exists.

Finally, we apply our first result to derive similar results for one-way TMs. We show that requiring nondeterministic one-way TMs to work in the guess-and-verify mode may increase the space complexity from logarithmic in the input length for the general model to linear for the restricted model. In particular, we obtain that $1\mathrm{BNL} \subsetneq 1\mathrm{NL}$.

Overview on the rest of the paper. In Section 3, we present some tools from communication complexity theory needed for the proofs of our lower bounds. In Section 4, the tradeoff between OBDD size and the number of nondeterministic variables is proven. Section 5 deals with the relationship between general and synchronous nondeterministic OBDDs. Section 6 contains the results for one-way TMs. We conclude the paper with a summary and an open problem.

3. Tools from communication complexity theory

In this section, we define deterministic and nondeterministic communication protocols and state two lemmas required for the proof of the main theorem of the paper. For a thorough introduction to communication complexity theory, we refer to the monographs of Hromkovič [9] and Kushilevitz and Nisan [17].

A *deterministic two-party communication protocol* is an algorithm by which two players, called Alice and Bob, cooperatively evaluate a function $f : X \times Y \to \{0,1\}$, where X and Y are finite sets. Alice obtains an input $x \in X$ and Bob an input $y \in Y$. The players determine $f(x,y)$ by sending messages to each other. Each player is assumed to have unlimited (but deterministic) computational power to compute her (his) messages. 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 $(x,y) \in X \times Y$.

Here we only consider protocols with one round of communication, so-called *one-way communication protocols*. In a one-way communication protocol, 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 $D^{\mathrm{A}\to\mathrm{B}}(f)$ to denote the (*deterministic*) *one-way communication complexity of* f, by which we mean the minimum number of bits sent by Alice in a deterministic one-way protocol for f. Furthermore, we also consider nondeterministic one-way protocols.

Definition 7. A *nondeterministic one-way communication protocol* for a function $f : X \times Y \to \{0,1\}$ is a collection of deterministic one-way protocols $P_1, \dots, P_d$, where $d = 2^r$, with $f(x,y) = 1$ iff there is an $i \in \{1, \dots, d\}$ such that $P_i(x,y) = 1$. The number r is called the number of *nondeterministic bits of* P. Let the *number of witnesses of* P *for an input* (x,y) be $\mathrm{acc}_P(x,y) =$

$|\{i \mid P_i(x,y)=1\}|$. Furthermore, let the (*private*) *nondeterministic complexity* of P be $N(P) = r + \max_{1\leqslant i\leqslant d} D(P_i)$, where $D(P_i)$ denotes the deterministic complexity of P_i. The *nondeterministic one-way complexity of* f, $N^{\mathrm{A}\to\mathrm{B}}(f)$, is the minimum of $N(P)$ over all protocols P as described above. By the *nondeterministic one-way complexity of* f *with restriction to* r *nondeterministic bits and* w *witnesses for accepted inputs*, $N_{r,w}^{\mathrm{A}\to\mathrm{B}}(f)$, we mean the minimum complexity of a nondeterministic protocol P for f that uses r nondeterministic bits and that satisfies $\mathrm{acc}_P(x,y)\geqslant w$ for all $(x,y)\in f^{-1}(1)$. Finally, we define $N_r^{\mathrm{A}\to\mathrm{B}}(f) = N_{r,1}^{\mathrm{A}\to\mathrm{B}}(f)$.

The following well-known function plays a central role in this paper.

Definition 8. For an arbitrary boolean vector $a=(a_1,\dots,a_k)$, let $\mathrm{num}(a)=\sum_{i=1}^k a_i 2^{i-1}$. Define the function IND_n on inputs $x=(x_1,\dots,x_n)$ and $y=(y_1,\dots,y_{\lceil\log n\rceil})$ by $\mathrm{IND}_n(x,y)=x_{\mathrm{num}(y)+1}$ if $\mathrm{num}(y)\in\{0,\dots,n-1\}$ and $\mathrm{IND}_n(x,y)=0$ otherwise.

This function is referred to as "index" or "pointer function" in the literature. We may regard x as a memory contents and $\mathrm{num}(y)+1$ as an address in this memory. Kremer et al. [16] have shown that IND_n has complexity $\Omega(n)$ for randomized one-way communication protocols with bounded error (see [9,17] for a definition of randomized communication protocols). It is easy to see that essentially $\log n$ bits of communication are sufficient and also necessary to compute IND_n by nondeterministic one-way protocols. Here we will require the following more precise lower bound on the nondeterministic one-way communication complexity of IND_n that also takes the number of nondeterministic bits and the number of witnesses for accepted inputs into account:

Lemma 9. $N_{r,w}^{\mathrm{A}\to\mathrm{B}}(\mathrm{IND}_n)\geqslant nw\cdot 2^{-r-1}+r$.

This is a special case of a result derived by Hromkovič and the author [10] for a more complicated function. Since this result is an important ingredient in the proof of our main result, we provide a proof for it.

Proof. Let P be a nondeterministic one-way protocol for IND_n with r nondeterministic bits and $\mathrm{acc}_P(x,y)\geqslant w$ for all $(x,y)\in\mathrm{IND}_n^{-1}(1)$. Hence, there are $d=2^r$ deterministic one-way protocols $P_1,\dots,P_d$ that compute f as described in Definition 7. For $i=1,\dots,d$, let g_i be the function computed by P_i. Obviously, $g_i\leqslant\mathrm{IND}_n$. Observe that, by averaging, there is an index $i_0\in\{1,\dots,d\}$ such that $|g_{i_0}^{-1}(1)|\geqslant w|\mathrm{IND}_n^{-1}(1)|/d$. It is easy to see that $|\mathrm{IND}_n^{-1}(1)|=n\cdot 2^{n-1}$. Hence, $|g_{i_0}^{-1}(1)|\geqslant nw\cdot 2^{n-r-1}$. We prove that for any function g with $g\leqslant\mathrm{IND}_n$, $D^{\mathrm{A}\to\mathrm{B}}(g)\geqslant|g^{-1}(1)|/2^n$. From this, the lemma follows.

We consider the communication matrix M_g of $g\leqslant\mathrm{IND}_n$, which is the $2^n\times 2^{\lceil\log n\rceil}$-matrix with 0- and 1-entries defined by $M_g(x,y)=g(x,y)$ for $x\in\{0,1\}^n$ and $y\in\{0,1\}^{\lceil\log n\rceil}$. It is a well-known fact that $D^{\mathrm{A}\to\mathrm{B}}(g)=\lceil\log k\rceil$, where k is the number of different rows of M_g.

Hence, it is sufficient to prove that $\log k\geqslant|g^{-1}(1)|/2^n$. Let $a_1,\dots,a_k\in\{0,1\}^{2^{\lceil\log n\rceil}}$ be the different rows of M_g. By the definition of IND_n, only the first n entries of each a_i can be nonzero.

Let m_i be the number of occurrences of a_i in the matrix M_g and let ℓ_i be the number of ones in a_i. Then $m_1 + \cdots + m_k = 2^n$. The key observation for the proof is that a_i can occur in at most $2^{n-\ell_i}$ rows of M_g since $g \leqslant \mathrm{IND}_n$. Hence, $m_i \leqslant 2^{n-\ell_i}$ or, equivalently, $\ell_i \leqslant n - \log m_i$. We obtain an upper bound on the number of ones in M_g by maximizing the function $f : \mathbb{R}^n \to \mathbb{R}$ with $f(x_1, \ldots, x_n) = \sum_{i=1}^{k} x_i(n - \log x_i)$ for $x_1, \ldots, x_k \in \mathbb{R}$ subject to the constraints $x_i \geqslant 0$ for all $i = 1, \ldots, k$ and $x_1 + \cdots + x_k = 2^n$. By the method of Lagrangian multipliers, it follows that the maximum is attained for $x_1 = \cdots = x_k = 2^n/k$. We thus obtain the upper bound $|g^{-1}(1)| \leqslant 2^n \cdot \log k$, which yields the desired lower bound on $\log k$. $\square$

Finally, we briefly revisit the well-known standard method for proving lower bounds on the OBDD size, using the formalism from [24]. This makes use of the following reducibility concept from communication complexity theory (due to [3]).

Definition 10 (Rectangular reduction). Let X_f, Y_f and X_g, Y_g be finite sets. Let $f : X_f \times Y_f \to \{0,1\}$ and $g : X_g \times Y_g \to \{0,1\}$ be arbitrary functions. Then we call a pair (φ_1, φ_2) of functions $\varphi_1 : X_f \to X_g$ and $\varphi_2 : Y_f \to Y_g$ a *rectangular reduction from f to g* if $g(\varphi_1(x), \varphi_2(y)) = f(x, y)$ for all $(x, y) \in X_f \times Y_f$. If such a pair of functions exists for f and g, we say that *f is reducible to g*.

We describe the proof method only for the special case of nondeterministic OBDDs that is important for this paper.

Lemma 11. *Let g be a boolean function defined on the variable set $X = \{x_1, \ldots, x_n\}$ and let π be a permutation of $\{1, \ldots, n\}$. W.l.o.g. (by renumbering) we may assume that π is the identity. Suppose there is a boolean function f defined on inputs from $X \times Y$, where X and Y are finite sets, and a $p \in \{1, \ldots, n-1\}$ such that f is reducible to $g : \{0,1\}^p \times \{0,1\}^{n-p} \to \{0,1\}$. Let G be a (general) nondeterministic π-OBDD for g that uses at most r nondeterministic variables and has w accepting paths for each input accepted by g. Then $\lceil \log |G| \rceil \geqslant N_{r,w}^{\mathrm{A}\to\mathrm{B}}(f) - r$.*

4. Tradeoff between OBDD size and nondeterminism

Now we are ready to present the main result of the paper. We consider the following function.

Definition 12. Let $N = 3k^2m$ and $m = n + \lceil \log n \rceil$. We define the function $\mathrm{MIND}_{k,n}$ (“masked k-fold index function”) on (disjoint) variable vectors $s^i = (s_1^i, \ldots, s_{km}^i)$, $t^i = (t_1^i, \ldots, t_{km}^i)$, and $u^i = (u_1^i, \ldots, u_{km}^i)$, where $i \in \{1, \ldots, k\}$. The vectors s^i and t^i are used as bit masks by which variables from u^i are selected. Let an input $((s^1, t^1, u^1), \ldots, (s^k, t^k, u^k)) \in \{0,1\}^{3k^2m}$ be given. If one of the vectors s^i does not contain exactly n ones, or one of the vectors t^i does not contain exactly $\lceil \log n \rceil$ ones, let $\mathrm{MIND}_{k,n}((s^1, t^1, v^1), \ldots, (s^k, t^k, v^k)) = 0$. Otherwise, let $p_{i,1} < \cdots < p_{i,n}$ be the positions of the ones in s^i, and let $q_{i,1} < \cdots < q_{i,\lceil \log n \rceil}$ be the positions of the ones in t^i, where $i = 1, \ldots, k$, and define

$$\begin{aligned}&\mathrm{MIND}_{k,n}((s^1,t^1,u^1),\dots,(s^k,t^k,u^k))\\&\quad=\mathrm{IND}_n((u^1_{p_{1,1}},\dots,u^1_{p_{1,n}}),(u^1_{q_{1,1}},\dots,u^1_{q_{1,\lceil\log n\rceil}}))\\&\qquad\wedge\cdots\wedge\mathrm{IND}_n((u^k_{p_{k,1}},\dots,u^k_{p_{k,n}}),(u^k_{q_{k,1}},\dots,u^k_{q_{k,\lceil\log n\rceil}})).\end{aligned}$$

Theorem 13 (Main Result).

(1) *The function* $\mathrm{MIND}_{k,n}$ *can be represented by synchronous nondeterministic OBDDs using* $k\lceil\log n\rceil$ *nondeterministic variables in size* $O(k^2n^3\log n)$; *but*
(2) *every synchronous nondeterministic OBDD representing the function* $\mathrm{MIND}_{k,n}$ *and using at most r nondeterministic variables has size* $2^{\Omega(n\cdot 2^{-r/k})}$.

Setting $k=n$ in the above theorem, we obtain the following separation result.

Corollary 14. *Let* $N=3n^2(n+\lceil\log n\rceil)$, *which is the input length of* $\mathrm{MIND}_{n,n}$.

(1) *The function* $\mathrm{MIND}_{n,n}$ *can be represented by synchronous nondeterministic OBDDs using* $n\lceil\log n\rceil=(1/3)\cdot(N/3)^{1/3}\log N\cdot(1+o(1))$ *nondeterministic variables in size* $O(N^{5/3}\log N)$; *but*
(2) *every synchronous nondeterministic OBDD representing* $\mathrm{MIND}_{n,n}$ *with at most* $O(\log N)$ *nondeterministic variables requires size* $2^{\Omega(N^{1/3})}$. *It still requires size* $2^{\Omega(N^{\varepsilon/3})}$ *if only* $(1-\varepsilon)\cdot(1/3)\cdot(N/3)^{1/3}\log N$ *nondeterministic variables may be used, where* $\varepsilon>0$ *is any constant.*

Proof. We first express n and $\log n$ in terms of the input length. We have $N=3n^3\cdot(1+\delta_n)$ and $\log N=3\log n\cdot(1+\delta'_n)$ for some $\delta_n,\delta'_n>0$ with $\delta_n,\delta'_n\to 0$ for $n\to\infty$. Thus, $n=(N/3)^{1/3}\cdot(1+\delta_n)^{-1/3}$ and $\log n=(1/3)\cdot\log N\cdot(1+\delta'_n)^{-1}$. Substituting $k=n$, the upper bound in part (1) of Theorem 13 becomes $O(k^2n^3\log n)=O(n^5\log n)=O(N^{5/3}\log N)$. If $r=O(\log N)$, and hence, also $r=O(\log n)$, the lower bound in part (2) is of order $2^{\Omega(n)}=2^{\Omega(N^{1/3})}$. Finally, let $r\leqslant(1-\varepsilon)\cdot(1/3)\cdot(N/3)^{1/3}\log N$. Expressing N in terms of n, this implies $r\leqslant(1-\varepsilon)\cdot n\log n\cdot(1+\delta_n)^{1/3}(1+\delta'_n)$. Thus, $2^{-r/k}=\Omega(n^{-1+\varepsilon})$ and $2^{\Omega(n\cdot 2^{-r/k})}=2^{\Omega(n^\varepsilon)}=2^{\Omega(N^{\varepsilon/3})}$. $\square$

Corollary 15. *The function* $\mathrm{MIND}_{n,n}$ *with input length N requires exponential size in N for* (*general*) *nondeterministic OBDDs with* $O(\log N)$ *nondeterministic variables. Furthermore, it also requires exponential size for* (*general*) *nondeterministic OBDDs with the restriction that no nondeterministic variable may appear after a decision variable on a path from the source to a sink* (*i.e., all nondeterministic variables have to be tested at the top of the graph*).

Proof. Let G be a nondeterministic OBDD for $\mathrm{MIND}_{n,n}$ with nondeterministic variables $y_1,\dots,y_r$, where $r=O(\log n)$. In G we replace $y_1,\dots,y_r$ with constants in all the different possible ways, obtaining deterministic OBDDs $G_1,\dots,G_{2^r}$. We construct a synchronous nondeterministic

OBDD G' representing the same function as G by nondeterministically choosing between $G_1, \ldots, G_{2^r}$ using nondeterministic nodes at the top of the new graph. Since $r = O(\log n)$, $|G'|$ is at most polynomially larger than $|G|$. Hence, the first claim follows from the lower bound in Corollary 14.

Now we consider the second claim. Let G be a nondeterministic OBDD for $\mathrm{MIND}_{n,n}$ where no nondeterministic variable is tested after a deterministic variable. The OBDD G has a part consisting of nondeterministic nodes at the top by which one of the deterministic nodes of G is chosen. We replace the nondeterministic nodes by a tree whose nodes are labeled by the minimal possible number r of nondeterministic variables. This does not change the represented function and can only decrease the size. We obviously have $r \leqslant \lceil \log|G| \rceil$. Now either $r \leqslant (1-\varepsilon) \cdot n \log n$ and we obtain an exponential lower bound by Theorem 13, or $r > (1-\varepsilon) \cdot n \log n$ and we get the lower bound $|G| \geqslant 2^{r-1} = n^{\Omega(n)}$. $\square$

In the remainder of the section, we prove Theorem 13. We start with the easier upper bound.

Proof of Theorem 13, part (1). The function $\mathrm{MIND}_{k,n}$ is the conjunction of k copies of the following function. The "masked index function" MIND_n is defined on the variable vectors $s = (s_1, \ldots, s_{km})$, $t = (t_1, \ldots, t_{km})$, and $u = (u_1, \ldots, u_{km})$, where $m = n + \lceil \log n \rceil$, by

$$\mathrm{MIND}_n(s, t, u) = \mathrm{IND}_n((u_{p_1}, \ldots, u_{p_n}), (u_{q_1}, \ldots, u_{q_{\lceil \log n \rceil}}))$$

in the case that $p_1 < \cdots < p_n$ and $q_1 < \cdots < q_{\lceil \log n \rceil}$ are the positions of ones in the s- and t-vector, resp., and $\mathrm{MIND}_n(s, t, u) = 0$ otherwise. The essence of the proof is to construct sub-OBDDs for the k copies of MIND_n in $\mathrm{MIND}_{k,n}$ and to combine these sub-OBDDs afterwards by identifying the 1-sink of the ith copy with the source of the $(i+1)$th one, for $i = 1, \ldots, k-1$.

Thus, we first construct a synchronous nondeterministic OBDD G for a single function MIND_n. We use the variable order described by

$$y_1, \ldots, y_{\lceil \log n \rceil}, s_1, t_1, u_1, s_2, t_2, u_2, \ldots, s_{km}, t_{km}, u_{km},$$

where $y_1, \ldots, y_{\lceil \log n \rceil}$ are the nondeterministic variables. With a tree of nondeterministic nodes labeled by the y-variables at the top of G, a deterministic sub-OBDD G_d from $G_1, \ldots, G_n$ is chosen. The number $d \in \{1, \ldots, n\}$ is interpreted as a guess of the address for the index function IND_n.

We describe the construction of G_d. While testing the variables in the chosen order, we store the number of ones already seen in the s- and the t-vector, resp. We only need nodes for storing numbers from $\{0, \ldots, n\}$ and $\{0, \ldots, \lceil \log n \rceil\}$, resp., on each level of the OBDD, since the function yields the output 0 for larger numbers of ones. Using this information, we can find the variables $u_{p_1}, \ldots, u_{p_n}$ and $u_{q_1}, \ldots, u_{q_{\lceil \log n \rceil}}$ for the evaluation of MIND_n. We compare the real address $\mathrm{num}(u_{q_1}, \ldots, u_{q_{\lceil \log n \rceil}}) + 1$ with the guessed address d and output the addressed bit u_{p_d} in the positive case and 0 otherwise. During the computation, we only need to store the addressed bit u_{p_d} if it is found before all address bits have been checked.

Altogether, G_d has size $O(k(n + \log n) \cdot n \cdot \log n)$. Thus, the overall size of G is $O(kn^3 \log n)$. The OBDD for $\mathrm{MIND}_{k,n}$ contains k copies of OBDDs of this type and therefore has size $O(k^2 n^3 \log n)$. $\square$

We now turn to the proof of the lower bound. A straightforward idea is to apply the standard proof method from Lemma 11. The "k-fold index function" (without bit masks) defined in the following appears to be a suitable candidate for a rectangular reduction to $\mathrm{MIND}_{k,n}$.

Definition 16. For inputs $x^i = (x^i_1, \ldots, x^i_n) \in \{0,1\}^n$ and $y^i = (y^i_1, \ldots, y^i_{\lceil \log n \rceil}) \in \{0,1\}^{\lceil \log n \rceil}$, where $i = 1, \ldots, k$, let

$$\mathrm{IND}_{k,n}((x^1, \ldots, x^k), (y^1, \ldots, y^k)) = \mathrm{IND}_n(x^1, y^1) \wedge \cdots \wedge \mathrm{IND}_n(x^k, y^k).$$

We would like to consider nondeterministic one-way protocols for $\mathrm{IND}_{k,n}$ according to the partition of the variables where Alice has $(x^1, \ldots, x^k)$ and Bob $(y^1, \ldots, y^k)$. It has been shown in [10] that in this case, the players essentially cannot do better than evaluate all k copies of IND_n independently which requires $k\lceil \log n \rceil$ nondeterministic bits. It is easy to see that a rectangular reduction from $\mathrm{IND}_{k,n}$ (with this partition of the variables) to $\mathrm{MIND}_{k,n}$ is possible if the variables of different blocks (s^i, t^i, u^i) are "completely interleaved" in the given variable order of the nondeterministic OBDD. Let $x^i_1, \ldots, x^i_{3km}$ be the list of variables in s^i, t^i, u^i in arbitrary order. Then an "interleaved order" is described by $x^1_1, x^2_1, \ldots, x^k_1$, $x^1_2, x^2_2, \ldots, x^k_2, \ldots, x^1_{3km}, x^2_{3km}, \ldots, x^k_{3km}$. But there is no reason why we should expect that the nondeterministic OBDD uses such an order. Intuitively, an order of the type defined below is more suitable.

Definition 17. For $i = 1, \ldots, k$, let $x^i = (x^i_1, \ldots, x^i_n)$. An order π of the variables x^i_j, $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant n$, is called *blockwise with respect to* $x^1, \ldots, x^k$ if there are permutations $(b_1, \ldots, b_k)$ of $\{1, \ldots, k\}$ and $(j_{i,1}, \ldots, j_{i,n})$ of $\{1, \ldots, n\}$ for $i = 1, \ldots, k$ such that π is the order described by

$$x^{b_1}_{j_{1,1}}, \ldots, x^{b_1}_{j_{1,n}}, x^{b_2}_{j_{2,1}}, \ldots, x^{b_2}_{j_{2,n}}, \ldots, x^{b_k}_{j_{k,1}}, \ldots, x^{b_k}_{j_{k,n}}.$$

The x^i are called *blocks* in this context. For the ease of notation, we may assume that the blocks are simply ordered according to $x^1, \ldots, x^k$ in π and that the variables within each block are ordered as in the definition of x^i.

For $\mathrm{MIND}_{k,n}$ we consider orders which are blockwise with respect to (s^i, t^i, u^i) if we ignore the nondeterministic variables. Such an order is used in the proof of the upper bound of $\mathrm{MIND}_{k,n}$. We may even believe that this is "the best choice." Our first idea for reducing $\mathrm{IND}_{k,n}$ to $\mathrm{MIND}_{k,n}$ does no longer work for blockwise orders. This is because the standard proof method only allows to bound the number of nodes on a single "cut" through the OBDD, and we do not know how we can avoid "bad cuts" with only few nodes in the case of a blockwise order. Hence, we have to find a new way to deal with this kind of orders.

In general, we even do not have a blockwise order, but an arbitrary one. We have defined the function $\mathrm{MIND}_{k,n}$ in such a way that we can select a suitable suborder by fixing the bit mask vectors. Although we cannot select an interleaved order as a suborder, we can at least turn an arbitrary order into a blockwise one.

Lemma 18. *Leg G be a synchronous nondeterministic OBDD for* $\mathrm{MIND}_{k,n}$. *Let* π *be the suborder of the decision variables in G. Then there are assignments to the s- and t-variables of* $\mathrm{MIND}_{k,n}$ *such that by applying these assignments to G one obtains a synchronous nondeterministic OBDD* G' *for the function* $\mathrm{IND}_{k,n}$ *that is no larger than G, uses at most as many nondeterministic variables as G, and where the decision variables are ordered according to* π_{b} *described by*

$$x_1^1, \dots, x_n^1, y_1^1, \dots, y_{\lceil \log n \rceil}^1, x_1^2, \dots, x_n^2, y_1^2, \dots, y_{\lceil \log n \rceil}^2, \dots, x_1^k, \dots, x_n^k, y_1^k, \dots, y_{\lceil \log n \rceil}^k$$

after renaming the selected u-variables.

Proof. Let L be the list of the u-variables of $\mathrm{MIND}_{k,n}$ ordered according to π. Only by deleting variables, we obtain a sublist of L where the variables appear in a blockwise order with respect to the u^i as blocks. This is done in steps $t = 1, \dots, k$. Let L_t be the list of variables we are still working with at the beginning of step t, and let m_t be the minimum number of variables in all blocks which have not been completely removed in the list L_t. We start with $L_1 = L$ and $m_1 = km$. At the end, the algorithm outputs the list of variables L', which is initially empty.

In step t, let p be the smallest index in L_t such that the sublist of elements with indices $1, \dots, p$ contains exactly $m = n + \lceil \log n \rceil$ variables of a block u^i. Let $b_t = i$ and choose indices $j_{t,1}, \dots, j_{t,m}$ such that the variables $u_{j_{t,1}}^{b_t}, \dots, u_{j_{t,m}}^{b_t}$ are under the first p variables of L_t. Append these variables to the output list L'. Afterwards, delete the first p variables and all variables of the block u^{b_t} from L_t to obtain L_{t+1}.

It is easy to verify that $m_t \geqslant (k - t + 1)m$ for $t = 1, \dots, k$, and hence, the above algorithm can really be carried out sufficiently often. Let

$$u_{j_{1,1}}^{b_1}, \dots, u_{j_{1,m}}^{b_1}, u_{j_{2,1}}^{b_2}, \dots, u_{j_{2,m}}^{b_2}, \dots, u_{j_{k,1}}^{b_k}, \dots, u_{j_{k,m}}^{b_k}$$

be the obtained sublist L' of variables. For $i = 1, \dots, k$, fix s^i such that it contains ones exactly at the positions $j_{i,1}, \dots, j_{i,n}$, and fix t^i such that it contains ones exactly at the positions $j_{i,n+1}, \dots, j_{i,n+\lceil \log n \rceil}$. It is a simple observation that, in general, assigning constants to variables may only reduce the OBDD size and the number of variables, and the OBDD obtained by the assignment (nondeterministically) represents the restricted function. $\square$

Hence, we are left with the task of proving a lower bound on the size of synchronous nondeterministic OBDDs with "few" nondeterministic variables for the function $\mathrm{IND}_{k,n}$ and the blockwise variable order π_{b} on the decision variables. Essentially, our plan is again to decompose the whole OBDD into sub-OBDDs which are responsible for the evaluation of the single copies of IND_n.

It is easy to see that $\lceil \log n \rceil$ nondeterministic variables are already sufficient to evaluate a single copy of IND_n in polynomial size by an OBDD with an arbitrary variable order. Hence, there will always be *some* copies which have only polynomially many nodes in a nondeterministic OBDD of minimal size representing $\mathrm{MIND}_{k,n}$. The difficulty for the proof is that we have to estimate the amount of nondeterminism "used up" for one copy of IND_n. This cannot be done by the known proof methods (it is easy to see that it does not work to simply count the number of nondeterministic variables in each part). The following lemma solves this problem. It is crucial for the proof of this lemma that we consider *synchronous* nondeterministic OBDDs.

Lemma 19 (Nondeterministic Partition Lemma). *Let f_n be a boolean function on n variables and let $f_{k,n}$ be the boolean function defined on the variable vectors $x^1, \dots, x^k$ consisting of n variables each by $f_{k,n}(x^1, \dots, x^k) = f_n(x^1) \wedge \cdots \wedge f_n(x^k)$. Let π_b be a blockwise variable order with respect to $x^1, \dots, x^k$. Let G be a synchronous nondeterministic OBDD for $f_{k,n}$ where the decision variables are ordered according to π_b and which uses r nondeterministic variables.*

Then there are synchronous nondeterministic OBDDs G_1 and G_2 with the order π_b on the decision variables and numbers $r_1 \in \{0, \dots, r\}$ and $w \in \{1, \dots, 2^{r_1}\}$ such that:

(1) $|G_1| \leqslant |G|$ *and* $|G_2| \leqslant |G|$;
(2) G_1 *represents* f_n, *uses at most* r_1 *nondeterministic variables, and there are at least w accepting paths in* G_1 *for each input in* $f_n^{-1}(1)$;
(3) G_2 *represents* $f_{k-1,n}$ *and uses at most* $r - r_1 + \lceil \log w \rceil$ *nondeterministic variables.*

Proof. Let G be as described in the lemma. Suppose that the order π_b of the decision variables is given by $x_1^1, \dots, x_n^1, \dots, x_1^k, \dots, x_n^k$. Let r_1 be the number of nondeterministic variables tested before x_n^1 (thus, $r - r_1$ nondeterministic variables are tested after x_n^1). Let y^1 and y^2 be vectors with the nondeterministic variables tested before and after x_n^1, resp.

For the construction of G_1, we consider the set of nodes in G reachable by assignments to x^1 and y^1. We replace such a node with the 0-sink, if the 1-sink of G is not reachable from it by assignments to $x^2, \dots, x^k$ and y^2, and with the 1-sink, otherwise. The resulting graph is called G_1. It can easily be verified that it represents f_n. We define w as the minimum of the number of accepting paths in G_1 for an input in $f_n^{-1}(1)$. Thus G_1 fulfills the requirements of the lemma.

The OBDD G_2 is constructed as follows. Choose an assignment $a \in f_n^{-1}(1)$ to x^1 such that G_1 has exactly w accepting paths for a. Let G_a be the nondeterministic OBDD on y^1, $x^2, \dots, x^k$ and y^2 obtained from G by fixing the x^1-variables according to a. The top of this graph consists of nondeterministic nodes labeled by y^1-variables. Call the nodes reached by assignments to y^1 "cut nodes." W.l.o.g., we may assume that none of the cut nodes represents the 0-function. (Otherwise, we remove the node, as well as all nodes used to reach it and the nodes only reachable from it. This does not change the represented function.)

By the choice of a and the above assumption, there are at most w paths belonging to assignments to y^1 by which cut nodes are reached, hence, the number of cut nodes is also bounded by w. Now we rearrange the top of the graph G_a consisting of the nodes labeled by y^1-variables such that only the minimal number of nondeterministic variables is used. Obviously, $\lceil \log w \rceil$ nondeterministic variables are sufficient for this. Call the resulting graph G_2. This is a synchronous nondeterministic OBDD that obviously represents $f_{k-1,n}$ and uses at most $r - r_1 + \lceil \log w \rceil$ nondeterministic variables. $\square$

According to the plan outlined above, it remains to prove a lower bound for $\mathrm{IND}_{k,n}$ and blockwise variable orders.

Lemma 20. *Let π_b be a blockwise order on the variables of* $\mathrm{IND}_{k,n}$ *with respect to the blocks* (x^i, y^i), *where* $x^i = (x_1^i, \dots, x_n^i)$ *and* $y^i = (y_1^i, \dots, y_{\lceil \log n \rceil}^i)$, *such that all* x^i*-variables appear before the*

y^i-variables for each i. Let G be a synchronous nondeterministic OBDD for $\mathrm{IND}_{k,n}$ *that has r nondeterministic variables and whose decision variables are ordered according to* π_b. *Then* $\lceil \log|G| \rceil = \Omega(n \cdot 2^{-r/k})$.

Proof. Let $s_{k,r}(n)$ be the minimal size of a synchronous nondeterministic OBDD for $\mathrm{IND}_{k,n}$ with at most r nondeterministic variables and the order π_b for the decision variables. We claim that

$$\lceil \log s_{k,r}(n) \rceil \geqslant 2^{1/k-2} \cdot n \cdot 2^{-r/k}.$$

From this we obtain the lower bound in the lemma. We prove the above inequality by induction on k, using the Partition Lemma for the induction step. The required lower bounds on the size of sub-OBDDs will be derived by the standard lower bound method for OBDDs (Lemma 11).

Case $k = 1$: By Lemma 9, $N_r^{\mathrm{A}\to\mathrm{B}}(\mathrm{IND}_n) \geqslant n \cdot 2^{-r-1} + r$. Lemma 11 yields $\lceil \log s_{1,r}(n) \rceil \geqslant 2^{-1} \cdot n \cdot 2^{-r}$.

Case $k > 1$: Suppose that the claim has been shown for $s_{k-1,r'}$, for arbitrary r'. Let G be a synchronous nondeterministic OBDD for $\mathrm{IND}_{k,n}$ with r nondeterministic variables and order π_b on the decision variables.

We first apply the Partition Lemma to obtain synchronous nondeterministic OBDDs G_1 and G_2 with their decision variables ordered according to π_b and numbers r_1 and w with the following properties:

- G_1 represents IND_n, uses at most r_1 nondeterministic variables, and there are at least w accepting paths for each input accepted by IND_n;
- G_2 represents $\mathrm{IND}_{k-1,n}$ and uses at most $r - r_1 + \lceil \log w \rceil$ nondeterministic variables.

Furthermore, $|G_1| \leqslant |G|$ and $|G_2| \leqslant |G|$. By Lemma 9, $N_{r_1,w}^{\mathrm{A}\to\mathrm{B}}(\mathrm{IND}_n) \geqslant nw \cdot 2^{-r_1-1} + r_1$. Applying Lemma 11, we get a lower bound on $|G_1|$. Together with the induction hypothesis we have

$$\lceil \log|G_1| \rceil \geqslant nw \cdot 2^{-r_1-1} \quad \text{and}$$
$$\lceil \log|G_2| \rceil \geqslant 2^{1/(k-1)-2} \cdot n \cdot 2^{-(r-r_1+\lceil \log w \rceil)/(k-1)}.$$

It follows that

$$\lceil \log s_{k,r}(n) \rceil \geqslant \max\{nw \cdot 2^{-r_1-1}, 2^{1/(k-1)-2} \cdot n \cdot 2^{-(r-r_1+\log w+1)/(k-1)}\},$$

where we have removed the ceiling using $\lceil \log w \rceil \leqslant \log w + 1$. The two functions within the maximum expression are monotonously increasing and decreasing in w, resp. Thus, the minimum with respect to w is attained if

$$nw \cdot 2^{-r_1-1} = 2^{1/(k-1)-2} \cdot n \cdot 2^{-(r-r_1+\log w+1)/(k-1)},$$

or equivalently,

$$w^{1+1/(k-1)} = 2^{-1} \cdot 2^{-r/(k-1)} \cdot 2^{r_1(1+1/(k-1))}.$$

Solving for w, we obtain

$$w = 2^{1/k-2} \cdot 2^{-r/k} \cdot 2^{r_1+1}.$$

By substituting this into the above estimate for $\lceil \log s_{k,r}(n) \rceil$, we obtain the desired result,

$$\lceil \log s_{k,r}(n) \rceil \geqslant 2^{1/k-2} \cdot n \cdot 2^{-r/k}. \qquad \square$$

Finally, we put all things together to complete the proof of the main theorem.

Proof of Theorem 13, part (2). Let G be a synchronous nondeterministic OBDD for $\mathrm{MIND}_{k,n}$ with r nondeterministic variables. By Lemma 18, we obtain a synchronous nondeterministic OBDD G' for $\mathrm{IND}_{k,n}$ with $|G'| \leqslant |G|$, at most r nondeterministic variables and the blockwise order π_{b} on the decision variables as described in the above lemma. Applying the lemma, we get the desired lower bound. $\square$

5. Synchronous versus general nondeterministic OBDDs

In this section, we investigate the relationship between the restricted variant of nondeterministic OBDDs with a variable order on *all* variables (called synchronous nondeterministic OBDDs) and the general variant where only the decision variables have to be ordered. We first observe that the two models are equivalent if the number of nondeterministic variables is not limited.

Theorem 21. *Let G be a (general) nondeterministic OBDD that represents the boolean function f on n variables and uses $r(n)$ nondeterministic variables. Then there is a synchronous nondeterministic OBDD G' that is isomorphic to G apart from the labels at its nondeterministic nodes, also represents f and uses $r'(n) = (n+1)r(n)$ nondeterministic variables.*

Proof. Let $x_1, \dots, x_n$ and $y_1, \dots, y_{r(n)}$ be the decision and nondeterministic variables of G, resp. W.l.o.g., assume that $x_1, \dots, x_n$ is the order of the decision variables in G. In order to obtain the synchronous nondeterministic OBDD G', we replace $y_1, \dots, y_{r(n)}$ with new nondeterministic variables y_j^i, where $0 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant r(n)$. The variable order for G' is described by

$$y_1^0, \dots, y_{r(n)}^0, x_1, y_1^1, \dots, y_{r(n)}^1, x_2, y_1^2, \dots, y_{r(n)}^2, \dots, x_n, y_1^n, \dots, y_{r(n)}^n.$$

We obtain G' as follows. Our aim is to relabel the nondeterministic nodes on paths between x_i- and x_{i+1}-nodes by $y_1^i, \dots, y_{r(n)}^i$, but we have to take into account that not all x_i-variables may be tested on paths from the source to the sinks. The replacements are carried out bottom-up in G by starting breadth-first traversals at all x_i-nodes for $i = n, n-1, \dots, 1$. These traversals stop at nodes that have already been considered or at the sinks. Finally, the nondeterministic variables on paths starting at the source and leading to a node that has already been considered are replaced by $y_1^0, \dots, y_{r(n)}^0$. $\square$

The above construction leads to a considerable increase of the number of nondeterministic variables used in the OBDD. It is an obvious question whether we can also obtain a synchronous nondeterministic OBDD that is not much larger than the original graph and uses essentially the same amount of nondeterminism. In the following, we answer this question in the negative sense. We consider the following function.

Definition 22. Let $k, \ell, n \in \mathbb{N}$ with $k \leqslant \ell$ and let $m = n + \lceil \log n \rceil$. For $i = 1, \dots, \ell$, let $b^i = (s^i, t^i, u^i)$, where $s^i = (s^i_1, \dots, s^i_{km})$, $t^i = (t^i_1, \dots, t^i_{km})$, and $u^i = (u^i_1, \dots, u^i_{km})$ are vectors of boolean variables. Define the function $f_{k,\ell,n}$ on boolean variables $a_1, \dots, a_\ell$ and the vectors $b^1, \dots, b^\ell$ as follows. If $(a_1, \dots, a_\ell)$ does not contain exactly k ones, then let $f_{k,\ell,n}((a_1, b^1), \dots, (a_\ell, b^\ell)) = 0$. Otherwise, let $i_1 < \cdots < i_k$ be the positions of ones in $(a_1, \dots, a_\ell)$ and let $f_{k,\ell,n}((a_1, b^1), \dots, (a_\ell, b^\ell)) = \mathrm{MIND}_{k,n}(b^{i_1}, \dots, b^{i_k})$.

Theorem 23.

(1) *The function $f_{k,\ell,n}$ has a (general) nondeterministic OBDD of polynomial size in the input length with $k\lceil \log n \rceil$ nondeterministic variables.*

(2) *Let $\varepsilon > 0$ be any constant and suppose that ℓ is divisible by k. Let G be a synchronous nondeterministic OBDD that represents $f_{k,\ell,n}$ and uses at most $(1-\varepsilon)\ell\lceil \log n \rceil$ nondeterministic variables. Then there is a constant $c > 0$ such that $\lceil \log|G| \rceil \geqslant \min\{(\varepsilon/2)k \log n,\ cn^{\varepsilon/2}\}$.*

Proof. *Upper bound, part (1)*: We use the variable order $a_1, b^1, \dots, a_\ell, b^\ell$, where the order within each $b^i = (s^i, t^i, u^i)$ is the same as in the construction of the nondeterministic OBDDs for MIND_n in the upper bound for $\mathrm{MIND}_{k,n}$ (proof of Theorem 13, part (1)). While testing the variables in the chosen order, we evaluate k copies of the function MIND_n defined on the variables belonging to the b^i with $a_i = 1$. This can be done in the same way as in the construction for the proof of Theorem 13, we only need k sets of $\lceil \log n \rceil$ nondeterministic variables each that are used at the appropriate places depending on the values of $a_1, \dots, a_\ell$. If $a_1, \dots, a_\ell$ do not contain exactly k ones, then the 0-sink is reached. Using the earlier results, it is obvious that the whole construction can be done in polynomial size. The number of nondeterministic variables is $k\lceil \log n \rceil$.

Lower bound, part (2): Our aim is to show that a synchronous nondeterministic OBDD for $f_{k,\ell,n}$ essentially cannot do better than the obvious one that uses a separate set of $\lceil \log n \rceil$ nondeterministic variables for each of the blocks $b^1, \dots, b^\ell$, since it is not known in advance which of these blocks are used for the evaluation of $\mathrm{MIND}_{k,n}$.

Let G be a synchronous nondeterministic π-OBDD for $f_{k,\ell,n}$ that uses $r \leqslant (1-\varepsilon)\ell\lceil \log n \rceil$ nondeterministic variables, where $\varepsilon > 0$ is any constant. For $i = 1, \dots, \ell/k$, let $c_i \in \{0,1\}^\ell$ be the assignment to $(a_1, \dots, a_\ell)$ such that the blocks $b^{(i-1)k+1}, \dots, b^{ik}$ are used for the evaluation of $\mathrm{MIND}_{k,n}$ in $f_{k,\ell,n}$ if the a-variables are set to constants according to c_i. Let G_i be the OBDD that is obtained from G by the replacement of variables according to c_i. Then G_i is a synchronous nondeterministic π-OBDD for the function $\mathrm{MIND}_{k,n}$ defined on the variables in $b^{(i-1)k+1}, \dots, b^{ik}$.

We apply Lemma 18 to fix the variables in the s- and t-vectors of $b^{(i-1)k+1}, \dots, b^{ik}$ such that we obtain a synchronous nondeterministic OBDD G'_i for $\mathrm{IND}_{k,n}$ from G_i that has its decision variables ordered blockwise with respect to the blocks $u^{(i-1)k+1}, \dots, u^{ik}$ and is no larger than G_i. The complete variable order π_i of G'_i that we obtain by deleting the fixed variables from π contains the same nondeterministic variables as π and the variables in $u^{(i-1)k+1}, \dots, u^{ik}$.

Let r_i^0 be the number of nondeterministic variables tested before the first u-variable according to π_i. As in the proof of Corollary 15, we may assume that only the minimal number of nondeterministic variables needed to reach the first decision nodes in G_i' is used. This implies the lower bound

$$\lceil \log |G_i'| \rceil \geqslant r_i^0. \tag{1}$$

Let r_i be the number of nondeterministic variables tested after the first u-variable and before the last u-variable according to π_i. The sets of nondeterministic variables between the first and the last u-variable according to the variable orders π_i are disjoint by definition. Furthermore, each variable order π_i is a suborder of π. Hence, $r_1 + \cdots + r_{\ell/k} \leqslant r$. By averaging, there is at least one i_0 such that $r_{i_0} \leqslant r/(\ell/k)$. We may assume that no nondeterministic variable is tested after the last u-variable in G_{i_0}', since nodes labeled by such variables can be replaced with the 0- or 1-sink. Thus, G_{i_0}' has at most $r_{i_0}^0 + r/(\ell/k)$ nondeterministic variables altogether. By Lemma 20, we obtain

$$\lceil \log |G_{i_0}'| \rceil = \Omega(n \cdot 2^{-(r_{i_0}^0 + r/(\ell/k))/k}) = \Omega(n \cdot 2^{-r_{i_0}^0/k - r/\ell}). \tag{2}$$

If $r_{i_0}^0 > (\varepsilon/2)k\lceil \log n \rceil$, then $\lceil \log |G_{i_0}'| \rceil > (\varepsilon/2)k\lceil \log n \rceil$ by the lower bound (1). If $r_{i_0}^0 \leqslant (\varepsilon/2)k\lceil \log n \rceil$, then by the lower bound (2) and the fact that $r \leqslant (1-\varepsilon)\ell\lceil \log n \rceil$ due to the hypothesis,

$$\lceil \log |G_{i_0}'| \rceil = \Omega(n \cdot 2^{-r_{i_0}^0/k - r/\ell}) = \Omega(n \cdot 2^{-(\varepsilon/2)\lceil \log n \rceil - (1-\varepsilon)\lceil \log n \rceil}) = \Omega(n^{\varepsilon/2}).$$

Altogether,

$$\lceil \log |G| \rceil \geqslant \lceil \log |G_{i_0}| \rceil \geqslant \lceil \log |G_{i_0}'| \rceil \geqslant \min\{(\varepsilon/2)k\lceil \log n \rceil,\ cn^{\varepsilon/2}\},$$

for some suitable constant $c > 0$, as desired. □

Corollary 24. *For any constant δ with $0 < \delta \leqslant 1$, let g_N be the boolean function on N variables defined by $g_N = f_{k,\ell,n}$ with $k = n$, $\ell = n^c$, and $c = \lceil 3/\delta - 2 \rceil$. Then g_N has the following properties.*

(1) *The function g_N can be represented by a (general) nondeterministic OBDD of polynomial size in N with $r(N) = O(N^{\delta/3} \log N)$ nondeterministic variables; but*
(2) *each synchronous nondeterministic OBDD that represents g_N and has at most $(1-\varepsilon)(N/3)^{1-\delta} r(N)$ nondeterministic variables, $\varepsilon > 0$ any constant, requires exponential size in N.*

Proof. The input length of g_N is $N = 3k\ell m + \ell$. We have $N = 3n^{c+2}(1+\gamma_n)$ and $\log N = (c+2)\log n(1+\gamma_n')$ for some $\gamma_n, \gamma_n' > 0$ with $\gamma_n, \gamma_n' \to 0$ for $n \to \infty$. Hence, $n = (N/3)^{1/(c+2)}(1+\gamma_n)^{-1/(c+2)}$ and $\log n = (1/(c+2))\log N(1+\gamma_n')^{-1}$. Furthermore, $\delta/4 < 1/(c+2) = 1/\lceil 3/\delta \rceil \leqslant \delta/3$ and $(c-1)/(c+2) \geqslant 1-\delta$.

By the upper bound from Theorem 23, g_N can be represented in polynomial size by a nondeterministic OBDD using $r(N) = k\lceil \log n \rceil = n\lceil \log n \rceil = O(N^{\delta/3} \log N)$ nondeterministic variables. This proves part (1). Now let G be a synchronous nondeterministic OBDD for g_N with

$r'(N) \leqslant (1-\varepsilon)(N/3)^{1-\delta} r(N)$ nondeterministic variables. Since $1-\delta \leqslant (c-1)/(c+2)$,

$$\begin{aligned} r'(N) &\leqslant (1-\varepsilon)(N/3)^{(c-1)/(c+2)} r(N) \\ &= (1-\varepsilon) n^{c-1} (1+\gamma_n)^{(c-1)/(c+2)} r(N) \\ &\leqslant (1-\varepsilon/2) n^c \lceil \log n \rceil = (1-\varepsilon/2) \ell \lceil \log n \rceil \end{aligned}$$

for sufficiently large n. Hence, the lower bound from Theorem 23 implies that $\lceil \log |G| \rceil = \Omega(n^{\varepsilon/4}) = \Omega(N^{\varepsilon\delta/16})$ and thus part (2) follows. $\square$

This shows that the construction in Theorem 21 cannot be improved significantly.

6. Results for one-way Turing machines

In this section, we apply the tradeoff between OBDD size and the number of nondeterministic variables from Section 4 to derive a similar result for nonuniform as well as for uniform one-way TMs.

Let 1L/Poly, 1ENL/Poly, 1BNL/Poly, and 1NL/Poly be the nonuniform analogs of the classes 1L, 1ENL, 1BNL, and 1NL considered in the introduction. The new classes are defined by using nonuniform TMs with polynomially bounded advice strings instead of uniform TMs. The uniform classes are obviously contained in their nonuniform counterparts if we identify a sequence boolean functions $(f_n)_{n \in \mathbb{N}}$ with the language $A = \bigcup_{n \in \mathbb{N}} f_n^{-1}(1)$. Analogously to the uniform case, we obtain:

Proposition 25.

(1) 1L/Poly$\subseteq$1ENL/Poly$\subseteq$1BNL/Poly$\subseteq$1NL/Poly; *and*
(2) L/Poly$\subseteq$ENL/Poly$\subseteq$BNL/Poly$\subseteq$NL/Poly.

Proof. Only the inclusions 1ENL/Poly$\subseteq$1BNL/Poly and ENL/Poly$\subseteq$BNL/Poly are nontrivial, and these can be proven using the ideas for the uniform case due to Niepel [20]. To make the paper self-contained, we revisit her construction for the one-way case, the two-way case is handled analogously.

Suppose that M is a 1ENL-TM that uses space $s(n) = \Theta(\log n)$ for input length n. The TM M can be decomposed into two sub-machines M_1 and M_2 that work as follows. Machine M_1 is a deterministic one-way TM with space bound $s(n)$ that reads the input x and generates a word $w(x)$ of length $s(|x|)$ on its work tape. Machine M_2 is a nondeterministic TM that uses space at most n for inputs of length n, reads $w(x)$ and outputs 0 or 1.

We now construct a BNL-TM M' with space bound $s(n)$ for input length n that simulates M. The machine M' uses extra tracks wherever required and thus can store up to $O(\log n)$ additional bits. On an input x, M' first guesses $w(x)$ on its work tape. This can be done in the obvious way, since the work tape of M' as well as the string $w(x)$ have length $s(|x|)$. Then M' simulates M_2 on this word $w(x)$ as input, for which it needs at most space $w(x) \leqslant s(|x|)$ due to the space restriction of M_2. Finally, M' simulates M_1 to check whether $w(x)$ has been correctly guessed. This can again be done with space $s(|x|)$.

It is now easy to see that the whole simulation also works if both machines M_1 and M_2 have access to an advice tape with the same contents. $\square$

For the uniform case, it is an open question whether nondeterminism at the end of the computation of a logspace TM helps anything, i.e., whether $\mathrm{ENL} \neq \mathrm{L}$ and $1\mathrm{ENL} \neq 1\mathrm{L}$. Somewhat surprisingly, it is trivial to answer these questions in the negative sense for the nonuniform case:

Proposition 26. $\mathrm{ENL/Poly} = \mathrm{L/Poly}$ *and* $1\mathrm{ENL/Poly} = 1\mathrm{L/Poly}$.

Proof. This is most easily seen by taking polynomial-size nondeterministic BPs and nondeterministic OBDDs, resp., for representing the sequences of functions in the considered classes. The claim follows by observing that the source of a subgraph of nondeterministic nodes at the end of a BP can be replaced either with the 0- or with the 1-sink, depending on whether the 1-sink is reachable from it. $\square$

On the other hand, we can use the results from Section 4 for proving that one-way TMs with nondeterminism at the beginning of the computation are really weaker than unrestricted nondeterministic one-way TMs in the nonuniform setting:

Theorem 27. $1\mathrm{BNL/Poly} \subsetneq 1\mathrm{NL/Poly}$.

Proof. We consider the sequence of functions $(\mathrm{MIND}_{n,n})_{n \in \mathbb{N}}$ (this can be extended to a sequence defined on arbitrary input lengths by adding dummy variables). Taking Proposition 6 from the introduction into account, the upper bound follows from part (1) of Corollary 14 and the lower bound from Corollary 15. $\square$

Finally, we even have an analogous result for the uniform setting.

Theorem 28. $1\mathrm{BNL} \subsetneq 1\mathrm{NL}$.

Proof. We consider the language of accepted inputs of the functions $\mathrm{MIND}_{n,n}$, $n \in \mathbb{N}$. To make this precise, we regard each $x \in \{0,1\}^*$ of length $|x| = 3n^2(n + \lceil \log n \rceil)$ for some $n \in \mathbb{N}$ as an assignment to the variables of $\mathrm{MIND}_{n,n}$ according to the variable order $b^1, \ldots, b^n$, where $b^i = (s_1^i, t_1^i, u_1^i, \ldots, s_{nm}^i, t_{nm}^i, u_{nm}^i)$ for $i = 1, \ldots, n$ and $m = n + \lceil \log n \rceil$, and define

$$A = \{x \in \{0,1\}^* \mid \exists n \in \mathbb{N} : 3n^2(n + \lceil \log n \rceil) = |x| \wedge \mathrm{MIND}_{n,n}(x) = 1\}.$$

By Corollary 15, this language is not contained in the class 1BNL (since it is not even contained in 1BNL/Poly). Furthermore, the same ideas as in the proof of the upper bound part of Theorem 13 also yield a uniform one-way logspace TM for A. We only have to cope with the technical problem that a one-way TM does not have access to its input length at the beginning.

We construct a nondeterministic one-way TM M for A whose space bound $s(N)$, N the input length, is the logarithm of the upper bound on the size of the OBDDs from Theorem 13, i.e., $s(N) = \lceil (5/3)\log N + c \rceil$ for some constant $c > 0$. Using a counter that records the actual position on the input tape, M can determine the input length after reading the whole input. At the beginning of the computation for an input $x \in \{0,1\}^*$, the machine uses the $s(|x|)$ marked cells on its work tape to nondeterministically guess a number $n \in \{1, \ldots, 2^{s(|x|)}\}$. If $x \in A$, there is a such an n with $3n^2(n + \lceil \log n \rceil) = |x|$, since each suitable n satisfies $n \leqslant |x|$ and we have $|x| \leqslant 2^{s(|x|)}$.

The machine M then carries out the same algorithm as in the proof of the upper bound of Theorem 13 checking whether $\mathrm{MIND}_{n,n}(x) = 1$. If M runs out of input bits during the algorithm or it finds out that $|x| \neq 3n^2(n + \lceil n \rceil)$ at the end, the output is 0. Otherwise, M can output $\mathrm{MIND}_{n,n}(x)$. Hence, $A \in 1\mathrm{NL}$. $\square$

It is obvious that $1\mathrm{L}/\mathrm{Poly} \subsetneq 1\mathrm{BNL}/\mathrm{Poly}$ and $1\mathrm{ENL} \subsetneq 1\mathrm{BNL}$. As a separating language, consider, e.g., $A = \{x\#y \mid x, y \in \{0,1\}^*, x \neq y\}$ (the uniform string non-equality problem mentioned in the introduction). This language is contained in 1BNL, but not in $1\mathrm{L}/\mathrm{Poly} = 1\mathrm{ENL}/\mathrm{Poly}$. Altogether, the known relations among the classes for one-way TMs are now as follows:

$$1\mathrm{L}/\mathrm{Poly} = 1\mathrm{ENL}/\mathrm{Poly} \subsetneq 1\mathrm{BNL}/\mathrm{Poly} \subsetneq 1\mathrm{NL}/\mathrm{Poly}, \quad \text{and} \quad 1\mathrm{L} \subseteq 1\mathrm{ENL} \subsetneq 1\mathrm{BNL} \subsetneq 1\mathrm{NL}.$$

The problem whether nondeterminism at the end really helps for uniform one-way TMs still remains open (which is not surprising given the fact that it is equivalent to the LBA problem).

7. Conclusion

We have shown that the following restrictions may increase the size of nondeterministic OBDDs for a sequence of functions $f_n : \{0,1\}^n \to \{0,1\}$ from a polynomial to an exponential function in n:

- limiting the number of nondeterministic variables to $O(\log n)$;
- requiring that all nondeterministic variables are tested at the top;
- requiring synchronous nondeterminism and allowing only an increase of the number of nondeterministic variables by a factor of $n^{1-\delta}$, $\delta > 0$ any constant.

On the other hand, any general nondeterministic OBDD can be made synchronous while remaining polynomially large if we allow an increase of the nondeterministic variables by a factor of $n + 1$.

As a by-product, these results have also led to a deeper understanding of the structure of nondeterministic OBDDs. Nevertheless, the subject seems rich and interesting enough to warrant further study. For example, one may try to analyze the influence of the resource nondeterminism for more general types of BPs. It is already a challenging task to try to prove a tradeoff between the size of *nondeterministic read-once BPs* and the number of nondeterministic variables. In a deterministic read-once BP, each variable may appear at most once on each path from the source to one of the sinks. A nondeterministic read-once BP fulfills this restriction with respect to decision as well as nondeterministic variables.

Open Problem. Find a sequence of functions $f_n:\{0,1\}^n \to \{0,1\}$ such that f_n has polynomial size for unrestricted nondeterministic read-once BPs, but requires exponential size if only $O(\log n)$ nondeterministic variables may be used.

One may again try the conjunction of several copies of a function that is easy for nondeterministic read-once BPs, but a new approach is required for the proof of the respective lower bound.

Acknowledgments

Thanks to Ingo Wegener for proofreading and improving the upper bound of Theorem 13 and to Klaus-Jörn Lange for the pointer to the literature on one-way TMs and especially reference [20]. I have further benefitted from discussions with Juraj Hromkovič, Detlef Sieling, and Ingo Wegener. Finally, I would like to thank the two anonymous referees for their detailed and helpful comments.

References

[1] F. Ablayev, Randomization and nondeterminism are incomparable for polynomial ordered binary decision diagrams, in: Proceedings of 24th ICALP, Lecture Notes in Computer Science, Vol. 1256, Bologna, Italy, 1997, pp. 195–202.
[2] C. Àlvarez, J. Díaz, J. Torán, Complexity classes with complete problems between P and NP-C, in: Proceedings of 7th FCT, Szeged, Hungary, 1989, pp. 13–24.
[3] L. Babai, P. Frankl, J. Simon, Complexity classes in communication complexity theory, in: Proceedings of 27th FOCS, Toronto, Ontario, Canada, 1986, pp. 337–347.
[4] R. Beigel, J. Goldsmith, Downward separation fails catastrophically for limited nondeterminism classes, in: Proceedings of 9th Conference on Structure in Compl. Theory, Los Alamitos, California, 1994, pp. 134–138.
[5] R.E. Bryant, Graph-based algorithms for Boolean function manipulation, IEEE Trans. Comput. C-35 (8) (1986) 677–691.
[6] J. Díaz, J. Torán, Classes of bounded nondeterminism, Math. Systems Theory 23 (1) (1990) 21–32.
[7] U. Feige, J. Kilian, On limited versus polynomial nondeterminism, Chicago J. Theoret. Comput. Sci. 1 (1997) 1–20.
[8] J. Hartmanis, S. Mahaney, Languages simultaneously complete for one-way and two-way log-tape automata, SIAM J. Comput. 10 (2) (1981) 383–390.
[9] J. Hromkovič, Communication Complexity and Parallel Computing. EATCS Texts in Theoretical Computer Science, Springer, Berlin, 1997.
[10] J. Hromkovič, M. Sauerhoff, Tradeoffs between nondeterminism and complexity for communication protocols and branching programs, in: Proceedings of 17th STACS, Lecture Notes in Computer Science, Vol. 1770, Lille, France, 2000, pp. 145–156.
[11] S. P. Jukna, Entropy of contact circuits and lower bounds on their complexity, Theoret. Comput. Sci. 57 (1988) 113–129.
[12] C.M.R. Kintala, Computations with a restricted number of nondeterministic steps, Ph.D. Thesis, Pennsylvania State University, University Park, PA, 1977.
[13] C.M.R. Kintala, P.C. Fischer, Computations with a restricted number of nondeterministic steps, in: Proceedings of 9th STOC, Boulder, CO, 1977, pp. 178–185.
[14] H. Klauck, Lower bounds for computation with limited nondeterminism, in: Proceedings of 13th Conference on Computational Complexity, Buffalo, NY, 1998, pp. 141–152.

[15] M. Krause, Lower bounds for depth-restricted branching programs, Inform. Comput. 91 (1) (1991) 1–14.
[16] I. Kremer, N. Nisan, D. Ron, On randomized one-round communication complexity, Comput. Complexity 8 (1) (1999) 21–49.
[17] E. Kushilevitz, N. Nisan, Communication Complexity, Cambridge University Press, Cambridge, 1997.
[18] C. Meinel, The power of polynomial size Ω-branching programs, in: Proceedings of 5th STACS, Lecture Notes in Computer Science, Vol. 294, Bordeaux, France, 1988, pp. 81–90.
[19] I. Newman, Private vs. common random bits in communication complexity, Inform. Process. Lett. 39 (2) (1991) 67–71.
[20] I. Niepel, Logarithmisch platzbeschränkte Komplexitätsklassen, Master Thesis, Univ. Hamburg, 1987 (in German).
[21] C. Papadimitriou, M. Yannakakis, On limited nondeterminism and the complexity of the V-C dimension, J. Comput. System Sci. 53 (2) (1996) 161–170.
[22] A.A. Razborov, Lower bounds for deterministic and nondeterministic branching programs, in Proceedings of 8th FCT, Lecture Notes in Computer Science, Vol. 529, Gosen, Germany, 1991, pp. 47–60.
[23] M. Sauerhoff, Complexity Theoretical Results for Randomized Branching Programs, Ph.D. Thesis, Univ. Dortmund. Shaker, Aachen, 1999.
[24] M. Sauerhoff, On the size of randomized OBDDs and read-once branching programs for k-stable functions, Comput. Complexity 10 (2001) 155–178.
[25] I. Wegener, Branching Programs and Binary Decision Diagrams—Theory and Applications. Monographs on Discrete and Applied Mathematics, SIAM, Philadelphia, PA, 2000.