Theoretical Informatics and Applications

Theoret. Informatics Appl. 37 (2003) 51-66

DOI: 10.1051/ita:2003010

COMPLEXITY THEORETICAL RESULTS ON NONDETERMINISTIC GRAPH-DRIVEN READ-ONCE BRANCHING PROGRAMS

BEATE BOLLIG^{1,*}

Abstract. Branching programs are a well-established computation model for boolean functions, especially read-once branching programs (BP1s) have been studied intensively. Recently two restricted nondeterministic (parity) BP1 models, called nondeterministic (parity) graph-driven BP1s and well-structured nondeterministic (parity) graph-driven BP1s, have been investigated. The consistency test for a BP-model M is the test whether a given BP is really a BP of model M. Here it is proved that the consistency test is co-NP-complete for nondeterministic (parity) graph-driven BP1s. Moreover, a lower bound technique for nondeterministic graph-driven BP1s is presented. The method generalizes a technique for the well-structured model and is applied in order to answer in the affirmative the open question whether the model of nondeterministic graph-driven BP1s is a proper restriction of nondeterministic BP1s (with respect to polynomial size).

Mathematics Subject Classification. 68Q05, 68Q15, 94C10.

1. Introduction

1.1. Branching programs or binary decision diagrams

Besides boolean circuits and formulae branching programs (BPs), sometimes also called binary decision diagrams (BDDs), are one of the standard representations for boolean functions. (For a history of results on branching programs see, e.g., the monograph of Wegener [24].)

Keywords and phrases. Computational complexity, read-once branching programs, nondeterminism, lower bounds.

¹ FB Informatik, LS2, University Dortmund, 44221 Dortmund, Germany; e-mail: bollig@ls2.cs.uni-dortmund.de

 $^{^{}st}$ Supported in part by DFG We 1066/9.

Definition 1.1. A branching program (BP) or binary decision diagram (BDD) on the variable set $X_n = \{x_1, \ldots, x_n\}$ is a directed acyclic graph with one source and two sinks labeled by the constants 0 and 1. Each non-sink node (or decision node) is labeled by a boolean variable and has two outgoing edges, one labeled by 0 and the other by 1. A nondeterministic branching program is a branching program with some additional unlabeled nodes, called nondeterministic nodes or guessing nodes, which have out-degree 2.

An input $a \in \{0,1\}^n$ activates all edges consistent with a, *i.e.*, the edges labeled by a_i which leave nodes labeled by x_i . A computation path for an input a in a BP G is a path of edges activated by the input a which leads from the source to a sink. A computation path for an input a which leads to the 1-sink is called accepting path for a.

Let B_n denote the set of all boolean functions $f: \{0,1\}^n \to \{0,1\}$. The BP G represents a function $f \in B_n$ for which f(a) = 1 iff there exists an accepting path for the input a.

The *size* of a branching program G is the number of its nodes and is denoted by |G|. The *branching program size* of a boolean function f is the size of the smallest BP representing f. The *length* of a branching program is the maximum length of a path.

Note that in a deterministic BP each input defines exactly one computation path, whereas in a nondeterministic BP several computation paths are possible for each input.

It is well known that the logarithm of the branching program size is essentially the same as the space complexity of the nonuniform variant of Turing machines. Hence, it is a fundamental open problem to prove superpolynomial lower bounds on the size of branching programs for explicitly defined boolean functions even in the deterministic case. The nondeterministic case seems to be still harder (see e.g. the survey of Razborov [17]). In analogy to the definition for Turing machines different acceptance modes can be studied for branching programs.

Definition 1.2. A parity branching program (or \oplus -BP for short) is syntactically a nondeterministic branching program but instead of the usual existential non-determinism the parity acceptance mode is used. An input a is accepted iff the number of its accepting paths is odd.

In the following if nothing else is mentioned nondeterministic BPs mean BPs according to the usual existential nondeterminism (\vee -BPs for short).

Despite considerable efforts the best lower bound on the size of unrestricted branching programs remains the almost quadratic lower bound of order $\Omega(n^2/\log^2 n)$ proved by Nechiporuk in 1966 [16]. In order to learn more about the power of branching programs, various restricted models have been investigated intensively and several interesting restricted types of BPs could be analyzed quite successfully (for the latest breakthrough for semantic super-linear length BPs see [1,2], and [3], where using a subtle combinatorial reasoning super-polynomial lower bounds were obtained).

There are several possibilities to restrict BPs, among them restrictions on the multiplicity of variable tests or the ordering in which variables may be tested.

Definition 1.3. i) A branching program is called (syntactically) read k times (BPk) if each variable is tested on each path at most k times.

ii) A branching program is called *s-oblivious*, for a sequence of variables $s = (s_1, \ldots, s_l)$, $s_i \in X_n$, if the set of decision nodes can be partitioned into disjoint sets V_i , $1 \le i \le l$, such that all nodes from V_i are labeled by s_i and the edges which leave V_i -nodes reach a sink or a V_i -node where i > i.

Definitions of nondeterministic variants of restricted BPs are derived in a straightforward way. Borodin $et\ al.\ [7]$ have proved one of the first exponential lower bounds for nondeterministic BPks. Moreover, Thathachar [22] was even able to prove an exponential gap between the size of nondeterministic BPks and deterministic BP(k+1)s for an explicitly defined boolean function. This shows that the lower bound techniques for these models are highly developed. For parity branching programs the situation is quite different. Even the problem of proving superpolynomial lower bounds for parity read-once branching programs is still open.

Besides this complexity theoretical viewpoint people have used branching programs in applications. Representations of boolean functions which allow efficient algorithms for many operations, in particular synthesis (combine two functions by a binary operation) and equality test (do two representations represent the same function?) are necessary. Bryant [9] introduced ordered binary decision diagrams (OBDDs) which are up to now the most popular representation for formal circuit verification.

Definition 1.4. An OBDD is a branching program with a variable ordering given by a permutation π on the variable set. On each path from the source to the sinks, the variables at the nodes have to appear in the order prescribed by π (where some variables may be left out). A π -OBDD is an OBDD ordered according to π .

OBDDs in general do not have nice algorithmic properties. There are examples known such that g_n and h_n are two boolean functions which have linear-size OBDDs (for different variable orderings) but $f_n = g_n \vee h_n$ has even exponential BP1 size (for an example see, e.g., Prop. 2 in [4]). If a variable ordering π is fixed, all important operations can be performed efficiently (for a list of the operations see, e.g. [24]). Unfortunately, several important and also quite simple functions have exponential OBDD size. Therefore, more general representations with good algorithmic behavior are necessary. Gergov and Meinel [10,11] and Sieling and Wegener [20] have shown independently how deterministic read-once branching programs can be used for verification. In order to obtain efficient algorithms for many operations they have generalized the concept of variable orderings to graph orderings.

Definition 1.5. A graph ordering is a (deterministic) branching program with a single sink, where on each path from the source to the sink all variables appear

exactly once. A (nondeterministic) graph-driven BP1 is a (nondeterministic) BP1 G for which there exists a graph ordering G_0 with the following property: If for an input a, a variable x_i appears on a computation path of a in G before the variable x_j , then x_i also appears on the unique computation path of a in G_0 before x_j .

A well-structured (nondeterministic) graph-driven BP1 G is a BP1 for which there exists a graph ordering G_0 and a mapping α from the node set of G to the node set of G_0 such that for every node v in G the node $\alpha(v)$ is labeled with the same variable as v, and such that if a computation path of an input a passes through v, then the computation path of a in G_0 passes through $\alpha(v)$.

(Note that in [10,11] only (usual) graph-driven BP1s have been investigated. Furthermore, we use the notation of Sieling and Wegener [20] which is more common.) In (nondeterministic) graph-driven BP1s according to a fixed graph ordering for each input the variables are tested in the same ordering, whereas (different from OBDDs) for different inputs different orderings may be used. For nondeterministic BP1s graph orderings do not exist in general. For an input a there can exist computation paths for the input a according to different orderings. Wellstructured graph-driven BP1s according to a fixed graph ordering G_0 can easily be obtained in the following way. Since we are interested in lower bounds, we may assume that each graph ordering does not contain identical subgraphs. We start by a complete decision tree which is ordered according to G_0 , afterwards we merge all identical subgraphs. Finally, all nodes which have the same 0- and 1-successor are deleted. The difference between the two graph-driven models is the following one. For graph-driven BP1s G according to a graph ordering G_0 it is possible that a node v with label x_i is reached on the computation paths for two inputs a and b in G whereas the nodes with label x_i on the computation paths for the inputs a and b in G_0 are different. This is not allowed in the well-structured case. Figure 1 shows an example of a graph ordering G_0 , a well-structured nondeterministic G_0 -driven BP1 G_1 and a nondeterministic G_0 -driven BP1 G_2 . Sieling and Wegener [20] observed that there is a time-space trade-off between graph-driven and well-structured graph-driven BP1s in the deterministic case. The stronger structural property of the latter model leads to the design of simpler and faster algorithms but the storage space of these algorithms is larger than the storage space of the algorithms for graph-driven BP1s.

Any (nondeterministic) OBDD is well-structured since there exists exactly one x_i -node in any variable ordering for each variable x_i . In [4] it has been shown that even restricted well-structured nondeterministic graph-driven BP1s, called tree-driven nondeterministic BP1s, are a proper generalization of nondeterministic OBDDs.

The concept of graph-driven branching programs has turned out to be also useful in other settings, see e.g. [13] and [21]. Parity variants of these models are defined using the parity acceptance mode. Gergov and Meinel [10] were the first ones who suggested parity graph-driven BP1s as a data structure for boolean functions. Another reason for the investigation of parity graph-driven BP1s is

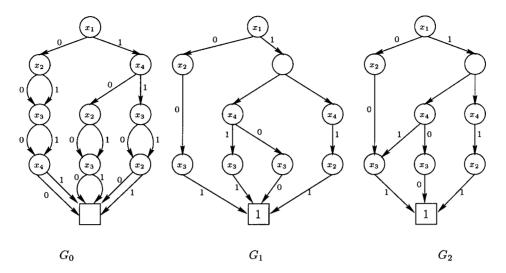


FIGURE 1. A graph ordering G_0 , a (well-structured) nondeterministic G_0 -driven BP1 G_2 (G_1) representing the function $f(x_1, x_2, x_3, x_4) = \overline{x_1} \ \overline{x_2} x_3 \lor x_1 x_2 x_4 \lor x_1 x_3 x_4 \lor x_1 \overline{x_3} \ \overline{x_4}$. Missing edges are leading to the 0-sink.

that until now exponential lower bounds on the size of parity read-once branching programs for explicitly defined boolean functions are unknown. One step towards the proof of such bounds might be to investigate BP models inbetween deterministic and parity BP1s. Nondeterministic and parity graph-driven BP1s have been investigated more intensively in [4]. For the parity case Brosenne $et\ al.$ [8] were the first ones realizing that the property of being well-structured can be used to prove lower bounds on the size of BP1s. They have proved the first (not strongly) exponential lower bound of order $2^{\Omega(n^{1/2})}$ on the size of well-structured parity graph-driven BP1s representing the characteristic function of linear codes. Well-structured parity and nondeterministic BP1s have been further investigated in [5] and [6].

A further variant of restricted nondeterministic branching programs which allows a very fine control of the available amount of nondeterminism is the following one.

Definition 1.6. An (ω, k) -BP1 G, $\omega \in \{\vee, \oplus\}$, consists of k deterministic readonce branching programs G_1, \ldots, G_k . If f_1, \ldots, f_k are the functions represented by G_1, \ldots, G_k , then G represents the function $f = f_1 \omega f_2 \omega \cdots \omega f_k$. The size of Gis $|G_1| + \ldots + |G_k|$. If G_1, \ldots, G_k are OBDDs, G is also called a partitioned binary decision diagram (PBDD).

Jain et al. [15] have introduced PBDDs as a representation of boolean functions for practical purpose. This type of nondeterministic BP is also interesting for complexity theory because it allows a bounded non-oblivious access to the input

56 B. Bollig

variables. Savický and Sieling [18] have presented the first lower bound methods for (\lor, k) -BP1s and (\oplus, k) -BP1s.

1.2. The results

Until now exponential lower bounds on the size of (usual) nondeterministic graph-driven BP1s for functions with polynomial nondeterministic BP1s size have been unknown. Hence, it should be clarified whether nondeterministic BP1s are really more powerful than nondeterministic graph-driven BP1s. The research on such questions may also lead to refinements of the known lower bound methods.

In Section 2, we investigate the consistency test. The consistency test for a BP-model M is the test whether a given BP is really a BP of model M. A rule of thumb that can be obtained by comparing several variants of BPs is that variants with a larger class of functions with small-size representations usually have less efficient algorithms. In Section 2, we prove the surprising result that this rule is not true for nondeterministic BP1s and the consistency test. Although there exist polynomial time algorithms for the consistency test for nondeterministic BP1s and oblivious nondeterministic BP1s (nondeterministic OBDDs) the consistency test for nondeterministic graph-driven BP1s is co-NP-complete.

In Section 3, we describe a criterion for boolean functions such that the lower bound method for well-structured nondeterministic graph-driven BP1s described in [6] still works.

Finally, Section 4 deals with the relationship between the restricted variants of nondeterministic BP1s. We show that the two restrictions being graph-driven or using a restricted number of nondeterministic nodes are incomparable with respect to polynomial size. Applying the lower bound criterion presented in Section 3 we prove that nondeterministic graph-driven BP1s are in fact significantly more restricted than nondeterministic BP1s. This answers an open question stated in [4].

2. The consistency test

The consistency test for nondeterministic BP1s is simple. Let G be the given BP. Let label(u) be the variable which is the label of the node u if u is a decision node and the empty set if u is a nondeterministic node. According to a topological order of the nodes, we compute for each decision node w the set P(w) of variables tested on some path from the source to w excluding the label of w. If $label(w) \in P(w)$, G cannot be a nondeterministic BP1. Otherwise the consistency test succeeds.

The consistency test for oblivious nondeterministic read-once branching programs (nondeterministic OBDDs) is also simple. Let G be the given BP on the variable set $\{x_1, \ldots, x_n\}$. For each variable x_i , $1 \le i \le n$, we merge all nodes labeled by x_i . The resulting graph is called G'. Afterwards we test whether G' has the read-once property. For this reason we test whether $label(u) \notin P(w)$ for all nodes w in G'.

For (usual) nondeterministic graph-driven BP1s the consistency test is more difficult.

Theorem 2.1. The problem to decide for a given branching program G whether G is a nondeterministic graph-driven BP1 is co-NP-complete.

Proof. We prove that the inconsistency test for nondeterministic graph-driven BP1s is NP-complete. The problem is contained in NP since we can guess two computation paths for an input a with contradictory orderings of the variables.

Now we present a polynomial time reduction from 3-SAT. Let (X,C) be an instance of 3-SAT where $X=\{x_1,\ldots,x_n\}$ is the set of variables and $C=\{c_1,\ldots,c_m\}$ is the set of clauses. W.l.o.g. we assume that each clause consists of tree different literals and each variable occurs in each clause at most once. Furthermore, we assume for each clause $c_i=x_{i_1}^{b_{i,1}}\vee x_{i_2}^{b_{i,2}}\vee x_{i_3}^{b_{i,3}},\ 1\leq i\leq m$ and $b_{i,j}\in\{0,1\}$ for $1\leq j\leq 3$, that $i_1< i_2< i_3$.

First, we introduce m new variables h_1, \ldots, h_m . Then we construct for each clause c_i the component G_i , $1 \le i \le m$, shown in Figure 2.

The constructed branching program G represents the disjunction of all these components. The upper part of G looks like a switch and consists of m-1 nondeterministic nodes (see Fig. 3).

We have to prove that the resulting branching program G is not a nondeterministic graph-driven BP1 iff the instance (X, C) for 3-SAT is satisfiable.

- \Leftarrow We prove that there exists an input with at least two computation path with different variable orderings. Let $a = \{0,1\}^n$ be a satisfying assignment for (X,C). Let a' be the assignment which consists of the assignment a to the x-variables and $h_i = 1, 1 \le i \le m$. There exist m accepting paths for a', on each of these accepting paths two h-variables are tested. The corresponding orderings of the h-variables are $h_i \to h_{i+1}, 1 \le i \le m-1$, which means that the variable h_i has to be tested before the variable h_{i+1} is tested and h_m has to be tested before h_1 . Contradiction.
- \Rightarrow We construct a graph-ordering G_0 such that G is a nondeterministic graph-driven BP1 if (X,C) is unsatisfiable. Since (X,C) is unsatisfiable there exists for each assignment a of the x-variables a first clause which is unsatisfied by a. Our graph ordering starts with a complete binary tree of size $2^n 1$ on the x-variables, where the x-variables are ordered according to x_1, \ldots, x_n . If c_i is the first clause that is unsatisfied by the assignment a of the x-variables the path that corresponds to a in G_0 continues with the ordering $h_{i+1}, \ldots, h_m, h_1, \ldots, h_i$.

Obviously Theorem 2.1 can be extended to the parity case.

3. A LOWER BOUND METHOD FOR NONDETERMINISTIC GRAPH-DRIVEN BP1S

We already know that the theory of communication complexity is a powerful tool for proving lower bounds on the size of nondeterministic oblivious BPs. (See,

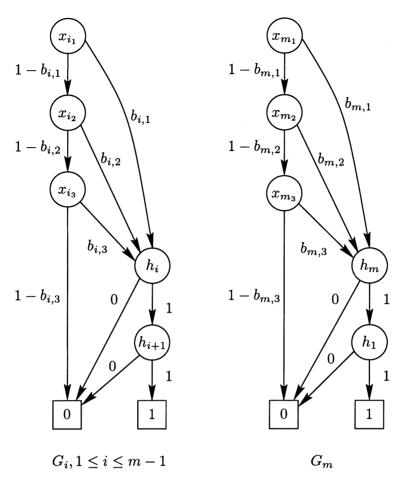


FIGURE 2. The components G_i , $1 \le i \le m-1$, and G_m in the co-NP-completeness proof.

e.g. [12] and [14] for the theory of communication complexity.) In [6] it has been shown how this tool can be used for proving large lower bounds on the size of well-structured nondeterministic and parity graph-driven BP1s. First, we restate the lower bound technique described in [6]. Consider a boolean function $f \in B_n$ which is defined on the variables in $X_n = \{x_1, \ldots, x_n\}$, and let $\Pi = (\mathcal{X}_A, \mathcal{X}_B)$ be a partition of X_n . Assume that Alice has access only to the input variables in \mathcal{X}_A and Bob has access only to the input variables in \mathcal{X}_B . In a one-way communication protocol, upon a given input x, Alice is allowed to send a single message (depending on the input variables in \mathcal{X}_A) to Bob who must then be able to compute the answer f(x). In an ω -nondeterministic communication protocol, $\omega \in \{\vee, \oplus\}$, Alice is allowed to guess a message. The function value is one if the number of guesses upon which Bob accepts the input matches the corresponding acceptance

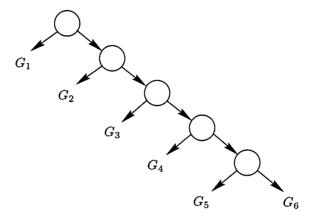


FIGURE 3. The structure of the upper part of G, if m = 6.

mode ω (is at least one in the case of $\omega = \vee$ or odd in the case of $\omega = \oplus$). The ω -nondeterministic one-way communication complexity of the function f is the number of bits of communication which have to be transmitted by such a protocol that computes f. It is denoted by $\mathrm{ND}^{A \to B}_{\omega}(f, \Pi)$. We abbreviate $\mathrm{ND}^{A \to B}_{\vee}(f, \Pi)$ by $\mathrm{ND}^{A \to B}(f, \Pi)$.

A filter of a set X is a closed upward subset of 2^X (i.e. if $S \in \mathcal{F}$, then all supersets of S are in \mathcal{F}). Let \mathcal{F} be a filter of $X_n = \{x_1, \ldots, x_n\}$. A subset $B \subseteq X_n$ is said to be in the boundary of \mathcal{F} if $B \notin \mathcal{F}$ but $B \cup \{x_i\} \in \mathcal{F}$ for some $x_i \in X_n$.

Let f be a function in B_n defined on the variables in X_n and \mathcal{F} be a filter on X_n . For a subset $Z \subseteq X_n$, we denote by $\mathcal{A}(Z)$ the set of all possible assignments to the variables in Z. Let $\Pi = (X_A, X_B)$ be a partition of X_n . If X_B is in the boundary of \mathcal{F} , then Π is called \mathcal{F} -partition of X_n . Finally, a function $f' \in B_n$ is called (ϵ, Π) -close to f, if there exists a set $R \subseteq \mathcal{A}(X_A)$ with $|R| \ge \epsilon \cdot 2^{|X_A|}$, such that f and f' coincide on all inputs in $R \times \mathcal{A}(X_B)$.

Theorem 3.1 ([6]). Let \mathcal{F} be a filter on X_n , $f \in B_n$, $0 < \epsilon \le 1$ and $\ell \in \mathbb{N}$. If for every \mathcal{F} -partition Π of X_n and for every function f' which is (ϵ, Π) -close to f it holds that $\mathrm{ND}_{\omega}^{A \to B}(f', \Pi) > \ell$, then any ω -nondeterministic graph-driven BP1 representing f either has a size of at least 2^{ℓ} or its graph ordering has a size of more than $1/\epsilon$ (for $\omega \in \{\vee, \oplus\}$).

The technique does not yield directly lower bounds for nondeterministic graph-driven BP1s because the size of the graph ordering of such a branching program is not part of the nondeterministic graph-driven BP1 size. In the well-structured case Bollig *et al.* [5] have proved that the size of a well-structured nondeterministic or parity BP1 G and the size of a graph ordering G_0 of minimal size such that G is G_0 -driven is polynomially related.

B. Bollig

Proposition 3.2 ([5]). For any well-structured ω -nondeterministic graph-driven BP1 G on n variables and $\omega \in \{\vee, \oplus\}$, there exists a graph ordering G_0 such that G is G_0 -driven and $|G_0| \leq 2n|G|$.

Using Proposition 3.2, Bollig and Woelfel [6] have proved the following lower bound criterion for well-structured ω -nondeterministic graph-driven BP1s, $\omega \in \{\vee, \oplus\}$.

Corollary 3.3 ([6]). Let $f \in B_n$ be a function satisfying the conditions of Theorem 3.1 for some filter \mathcal{F} on X_n and the parameters ϵ and ℓ . Then any well-structured ω -nondeterministic graph-driven BP1 for f has a size of at least $\min\{2^{\ell}, (\epsilon \cdot 2n)^{-1}\}.$

The proof of Proposition 3.2 cannot be generalized in a straightforward way for (usual) ω -nondeterministic graph-driven BP1s because the existence of the α -function is an essential part of the proof. Therefore, we cannot simply apply Corollary 3.3 to obtain lower bounds on the size of (usual) nondeterministic graph-driven BP1s. Until now it is an open question whether there exists a sequence of functions f_n which has polynomial complexity in the nondeterministic graph-driven BP1 model whereas the size of well-structured nondeterministic graph-driven BP1s representing f_n is exponential.

Definition 3.4. A maxterm of a function $f \in B_n$ is a partial input a^* for which $f_{|a^*} = 0$. A function f is m-dense if the minimal length of a maxterm of f is at least m. A function f is d-rare if the minimal Hamming distance for two arbitrary inputs $a, b \in f^{-1}(1)$ is at least d.

Theorem 3.5. Let \mathcal{F} be a filter on X_n and S be a set of \mathcal{F} of minimal size and m = n - |S| + 2. Let f be a 2-rare and m-dense function, $0 < \epsilon \le 1$ and $\ell \in \mathbb{N}$. Furthermore, for every \mathcal{F} -partition Π of X_n and for every function f' which is (ϵ, Π) -close to f it holds that $\mathrm{ND}^{A \to B}(f', \Pi) > \ell$. Then any nondeterministic graph-driven BP1 representing f has a size of at least $\min\{2^{\ell}, \epsilon^{-1}\}$.

Proof. We assume that $|G| < \min\{2^{\ell}, \epsilon^{-1}\}$. Let G_0 be a graph ordering of minimal size such that G is G_0 -driven.

- 1) $|G_0| \le \epsilon^{-1}$ Combining the assumption $|G_0| \le \epsilon^{-1}$ with Theorem 3.1 we can conclude that $|G| \ge 2^{\ell}$. Contradiction.
- 2) $|G_0| > \epsilon^{-1}$ Let $S^+(v)$ be the set of variables tested on a path in G_0 from v to the sink including the label of v. The filter \mathcal{F} defines a cut in G_0 , called *frontier*, in the following way. A node w is a frontier node if there exists an edge (v, w), $S^+(v) \in \mathcal{F}$ but $S^+(w) \notin \mathcal{F}$. We know that $S^+(s) = X_n$ for the source s and $S^+(t) = \emptyset$ for the sink and $S^+(w) = S^+(v) \setminus \{x_i\}$ for each edge (v, w), where v is labeled by x_i . Hence, each path from the source to the sink passes through exactly one frontier node and if w is a frontier node $S^+(w)$ is in the boundary of \mathcal{F} .

In the next step we define a partition $V_1, \ldots, V_k, k \leq |G_0|$, of the nodes on the cut. Two nodes v_{i_1} and v_{i_2} are in the same set V_i iff $S^+(v_{i_1}) = S^+(v_{i_2})$. If there are at most ϵ^{-1} different sets V_i we can continue similarly to Case 1. Otherwise we know that there exist more than ϵ^{-1} nodes on the cut for which $S^+(v_i) \neq S^+(v_j)$ if $v_i \neq v_j$. For each of theses nodes v we consider a 1-input for f that corresponds to a path from the source through v. We know that such inputs exist since f is m-dense. Similar to the cut in G_0 a cut in G can be defined. We choose one accepting path for each of the chosen 1-inputs. Each accepting path passes through exactly one node of the cut. Two of these accepting paths cannot pass through the same node since f is 2-rare and G has the read-once property. Otherwise, we can conclude that there is an implicant of f of length less than v and v cannot be 2-rare. Contradiction.

For some functions a weaker characterization is helpful.

Corollary 3.6. Let $f \in B_n$ be a function satisfying the conditions of Theorem 3.5 for some filter \mathcal{F} on X_n and the parameters ϵ and ℓ but f is not m-dense. Let $\Pi = (X_A, X_B)$ be an \mathcal{F} -partitioning of X_n and $f_{|X_A}$ be a subfunction of f, where we have replaced the X_A -variables by constants. If for every Π and for every subfunction $f_{|X_A|}$ it holds $f_{|X_A|} \neq 0$, then any nondeterministic graph-driven BP1 representing f has a size of at least $\min\{2^{\ell}, \epsilon^{-1}\}$.

The proof of Corollary 3.6 follows directly from the proof of Theorem 3.5. We only want to mention here that for the proof of Theorem 3.5 it is even sufficient that for every \mathcal{F} -partition $\Pi = (X_A, X_B)$ and for every subfunction $f_{|X_A}$ there exists at least one prime implicant of length $|X_B|$ instead of the 2-rareness of the function f.

4. Restricted nondeterministic read-once branching programs

The main result of this section is the proof that nondeterministic graph-driven BP1s are in fact restricted nondeterministic BP1s (with respect to polynomial size). Bollig and Woelfel [6] have shown that there exists an explicitly defined boolean function representable by $(\omega, 2)$ -BP1s, $\omega \in \{\lor, \oplus\}$, of polynomial size but with exponential well-structured ω -nondeterministic graph-driven BP1 size. Here we complete their result and prove that nondeterministic graph-driven BP1s and (\lor, k) -BP1s, k a constant, are incomparable, which means that there exist explicitly defined boolean functions representable in polynomial size by one of the models but with exponential size for the other model. A first hint that the two models are incomparable has already been given by the observation that the addition of one nondeterministic node may already decrease the size of (\lor, k) -BP1s from an exponential to a polynomial function in n (for small values of k) [18]. This is in contrast to the situation for nondeterministic graph-driven BP1s, where a

constant number of additional nondeterministic nodes may decrease the size only polynomially [4].

Proposition 4.1. There exists a sequence of boolean functions with exponential (\vee, k) -BP1 size, k constant, which can be represented by nondeterministic graph-driven BP1s of polynomial size.

Proof. First, we consider the following function due to Savický and Žák [19]. For a natural number n let p(n) be the smallest prime greater than n. The function weighted sum WS_n is defined on $x = (x_1, \ldots, x_n) \in \{0, 1\}^n$. Let $s = s(n) = (\sum_{i=1}^n ix_i) \mod p(n)$. If $1 \le s \le n$, $WS_n(x) = x_s$ and $WS_n = 0$ otherwise. (Note, that in [19] in the case $s \notin \{1, \ldots, n\}$ the function value of WS_n is defined as x_1 . But this does not change the complexity of the function in the following.) The disjoint conjunction $(WS_n)^k$ on kn variables is defined as follows. $WS_n^k(x) = WS_n(x^1) \wedge \ldots \wedge WS_n(x^k)$, where $x^i = (x_{(i-1)n+1}, \ldots, x_{in})$. Using the same pattern as in the proof of Theorem 6 from [25] a lower bound of $2^{\Omega(n)}$ can be proved, if k is a constant (this follows directly from [25]).

On the other hand the function $(WS_n)^k$ can even be computed by nondeterministic OBDDs. For $x = (x_1, \ldots, x_n) \in \{0, 1\}^n$, let $f_i(x) = 1, 1 \le i \le n$, iff $i = ((\sum_{i=1}^n ix_i) \mod p(n))$. Now we can define WS_n in the following way.

$$WS_n(x) = \bigvee_{1 \le i \le n} f_i(x) \land x_i, \text{ for } x = (x_1, \dots, x_n) \in \{0, 1\}^n.$$

Obviously the functions $f_i(x)$ can be represented by OBDDs of polynomial size and the same holds for $f_i(x) \wedge x_i$. Now WS_n is represented by a nondeterministic OBDD with n-1 nondeterministic nodes at the top. An index $i \in \{1, ..., n\}$ is guessed, afterwards $f_i(x) \wedge x_i = 1$ is checked. The generalization for $(WS_n)^k$ is easy, an index vector $(i_1, ..., i_k)$, $i_j \in \{1, ..., n\}$ and $1 \le j \le k$, is guessed at the top using n^k-1 nondeterministic nodes. (Note that the size of the nondeterministic OBDD is even independent of the chosen variable ordering.)

Now applying Corollary 3.6 we prove that nondeterministic graph-driven BP1s are in fact restricted nondeterministic BP1s.

One step in this direction has been done in [6]. The function n/2-MRC_n is defined on an $n \times n$ boolean matrix X on the variables $X_{n \times n} = \{x_{1,1}, \dots, x_{n,n}\}$. Its function value is 1 if and only if the following two conditions are fulfilled (for the sake of readability we assume that n is an even number).

- (1) The number of ones in the matrix is at least $n^2/4+n$ and at most $(3/4)n^2-n$.
- (2) The matrix either contains exactly n/2 monochromatic rows and each non-monochromatic row contains exactly n/2 ones, or it contains exactly n/2 monochromatic columns and each non-monochromatic column contains exactly n/2 ones.

It has been proved that n/2-MRC_n has exponential well-structured ω -non-deterministic graph-driven BP1 size, $\omega \in \{\vee, \oplus\}$, but can be represented by $(\omega, 2)$ -BP1s of polynomial size [6]. Here we investigate a simpler function.

The function n/2-RC_n is defined on an $n \times n$ boolean matrix X on the variables $X_{n \times n} = \{x_{1,1}, \dots, x_{n,n}\}$. (For the ease of readability we assume that n is an even number.) Its function value is 1 if and only if there exist exactly n/2 ones in each row or exactly n/2 ones in each column.

Theorem 4.2. The function n/2-RC_n can be represented by $(\vee, 2)$ -BP1s of polynomial size but its nondeterministic graph-driven BP1 size is $\Omega(2^{n/4})$.

Proof. The proof of the upper bound is easy. We construct two (deterministic) BP1s G_1 and G_2 , where G_1 (G_2) accepts exactly the satisfying inputs with n/2 ones in each row (column). The BP1 G_1 (G_2) uses a rowwise (columnwise) variable ordering which means that all variables of one row (column) are tested one after another. The size of G_i , $i \in \{1,2\}$, is bounded above by $O(n^3)$. (Note that G_1 and G_2 are even OBDDs.)

Next, we prove the lower bound of $\Omega(2^{n/4})$ for nondeterministic graph-driven BP1s representing n/2-RC_n. First, we have to define an appropriate filter \mathcal{F} on the variables $x_{1,1},\ldots,x_{n,n}$. A set T is in the filter \mathcal{F} if T contains all variables from n/2+1 arbitrary rows and n/2+1 arbitrary columns. If $\Pi=(X_A,X_B)$ is an \mathcal{F} -partition, then by definition $X_B \notin \mathcal{F}$ and there exists a variable $x_{i,j}$ such that $X_B \cup \{x_{i,j}\} \in \mathcal{F}$. Hence, X_A contains exactly n/2 variables from different rows and at most n/2 variables from different columns or vice versa. Every maxterm of n/2-RC_n contains at least n/2+1 literals from different rows and n/2+1 literals from different columns. Therefore n/2-RC_{n|X_A} $\neq 0$ for every subfunction of n/2-RC_n where we have replaced the X_A -variables by constants. Furthermore, the function n/2-RC_n is 2-rare since each 1-input consists of exactly $n/2 \cdot n$ 1-entries.

Let $\epsilon = 2^{-n/4}$. It remains to prove that for every \mathcal{F} -partition Π of $X_{n \times n}$ and for every function f' which is (ϵ, Π) -close to n/2-RC_n it holds that $\mathrm{ND}^{A \to B}(f', \Pi) > n/4$. For this we use some of the ideas presented in [6]. We may assume w.l.o.g. that X_A contains variables from exactly the rows $1, \ldots, n/2$, while there are at most n/2 columns from which variables are contained in X_A . There exists one row for which exactly one variable has been tested. W.l.o.g. we assume that there is exactly one variable of the (n/2)th row in X_A . Since f' is (ϵ, Π) -close to n/2-RC_n, there exists a subset $R \subseteq \mathcal{A}(X_A), |R| \ge \epsilon \cdot 2^{|X_A|}$, such that f' coincides with n/2-RC_n on all inputs in $R \times \mathcal{A}(X_B)$. For $1 \le i \le n/2$ let k_i be the number of variables in row i which are contained in X_A . We consider the mapping

$$\mu: \mathcal{A}(X_A) \to \{0,\ldots,k_1\} \times \ldots \times \{0,\ldots,k_{n/2}\},$$

which maps a partial assignment α to the tuple $\mu(\alpha) = (z_1, \dots, z_{n/2})$, where z_i is the number of bits in row i being fixed to 1 by α .

Let $\mu(R) = {\{\mu(\alpha) | \alpha \in R\}}$. There are exactly 2^{k_i} possible settings of the variables in row i which are contained in X_A and among these, there are $\binom{k_i}{z_i}$

settings for which row *i* contains exactly z_i ones. Hence, for every tuple $z = (z_1, \ldots, z_{n/2}) \in \{0, \ldots, k_1\} \times \ldots \times \{0, \ldots, k_{n/2}\}$ we obtain that

$$\frac{|\mu^{-1}(z)|}{|\mathcal{A}(X_A)|} = \frac{\binom{k_1}{z_1} \cdot \dots \cdot \binom{k_{n/2}}{z_{n/2}}}{2^{k_1} \cdot \dots \cdot 2^{k_{n/2}}} \le \frac{2^{k_1 - 1} \cdot \dots \cdot 2^{k_{n/2} - 1}}{2^{k_1} \cdot \dots \cdot 2^{k_{n/2}}} = 2^{-n/2}. \tag{1}$$

Since R is the union of all $\mu^{-1}(z)$ for $z \in \mu(R)$, there exists by the pigeonhole principle an element $z \in \mu(R)$ for which $|\mu^{-1}(z)| \ge |R|/|\mu(R)|$. Using the precondition that $|R| \ge \epsilon \cdot 2^{|X_A|}$ together with inequality (1) yields

$$|\mu(R)| \ge \frac{|R|}{|\mu^{-1}(z)|} \ge \frac{\epsilon \cdot 2^{|X_A|}}{2^{-n/2} \cdot |\mathcal{A}(X_A)|} = \epsilon \cdot 2^{n/2}.$$

It remains to prove that $ND^{A\to B}(f',\Pi) > \log |\mu(R)|$. We show that the communication matrix contains a diagonal $s \times s$ -submatrix, where $s = |\mu(R)|$. (See, e.g. [12] and [14] for the theory of communication complexity.) For an arbitrary partial assignment $\alpha \in R$ let $\mu(\alpha) = (\mu_1(\alpha), \dots, \mu_{n/2}(\alpha))$. We fix for each such α a corresponding partial assignment $\beta \in \mathcal{A}(X_B)$ as follows. By definition of X_A there exists at least one column c_j for which the variables $x_{n/2,j}, \ldots, x_{n,j}$ are in X_B . We fix theses variables to 0. This guarantees that it is impossible that all columns have exactly n/2 1-entries. Afterwards the assignment β fixes in row $i, 1 \le i \le n$, exactly $n/2 - \mu_i(\alpha)$ variables to 1, where $\mu_i(\alpha) = 0$ for i > n/2. Then for $(\alpha\beta)$ all rows contain exactly n/2 ones each. Hence, n/2-RC_n($\alpha\beta$) = 1. We consider now s arbitrary partial assignments $\alpha_1, \ldots, \alpha_s \in R$ such that $\mu(\alpha_i) \neq \mu(\alpha_j)$ for $i \neq j$. Let β_1, \ldots, β_s be the corresponding partial assignments in $\mathcal{A}(X_B)$. (It is obvious that also $\beta_i \neq \beta_j$ for $i \neq j$.) Clearly, the $s \times s$ -matrix which has in row i and column j the entry n/2-RC_n($\alpha_i\beta_i$) is a submatrix of the communication matrix of n/2-RC_n. It suffices to show that this matrix is a diagonal matrix. For the diagonal elements, we have already proved above that n/2-RC_n($\alpha_i\beta_i$) = 1. Consider now an element in row i and column j, $i \neq j$. Since $\alpha_i \neq \alpha_j$, there exists an index $t, 1 \le t \le n/2$, for which $\mu_t(\alpha_i) \ne \mu_t(\alpha_i)$. Hence, by construction the matrix X defined by the input $\alpha_i\beta_j$ contains in row t not exactly n/2 ones. The construction also ensures that the jth column contains less than n/2 ones. Altogether, this yields that n/2-RC_n $(\alpha_i\beta_i) = 0$.

Figure 4 summarizes the results. For a branching program model M we denote by P(M) the class of all boolean functions representable in polynomial size by the model M. Well-structured nondeterministic graph-driven BP1s are denoted by $w.s. \lor -G_0$ -BP1s and (usual) nondeterministic graph-driven BP1s by $\lor -G_0$ -BP1s. Solid arrows indicate inclusions and slashes through the lines proper inclusions. A dotted line between two classes means that these classes are incomparable.

Acknowledgements. The author would like to thank Ingo Wegener for fruitful discussions on the subject of the paper.

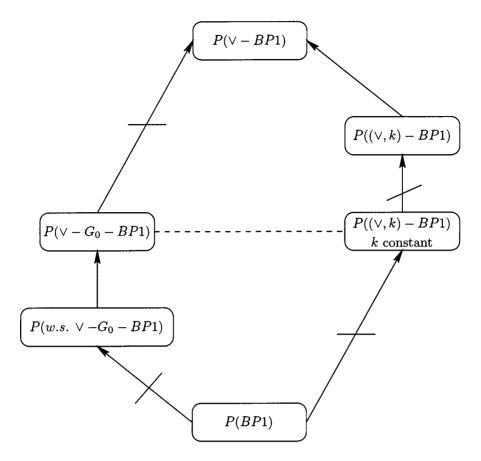


FIGURE 4. The complexity landscape for nondeterministic readonce branching programs.

References

- M. Ajtai, A non-linear time lower bound for boolean branching programs, in Proc. of 40th FOCS (1999) 60-70.
- [2] P. Beame, M. Saks, X. Sun and E. Vee, Super-linear time-space tradeoff lower bounds for randomized computation, in *Proc. of 41st FOCS* (2000) 169-179.
- [3] P. Beame and E. Vee, Time-space trade-offs, multiparty communication complexity, and nearest neighbor problems, in *Proc. of 34th STOC* (2002) 688-697.
- [4] B. Bollig, Restricted nondeterministic read-once branching programs and an exponential lower bound for integer multiplication. RAIRO: Theoret. Informatics Appl. 35 (2001) 149-162
- [5] B. Bollig, St. Waack and P. Woelfel, Parity graph-driven read-once branching programs and an exponential lower bound for integer multiplication, in *Proc. of 2nd IFIP International Conference on Theoretical Computer Science* (2002) 83-94.

- [6] B. Bollig and P. Woelfel, A lower bound technique for nondeterministic graph-driven readonce branching programs and its applications, in *Proc. of MFCS 2002*. Springer, *Lecture Notes in Comput. Sci.* 2420 (2002) 131-142.
- [7] A. Borodin, A. Razborov and R. Smolensky, On lower bounds for read-k-times branching programs. Comput. Complexity 3 (1993) 1-18.
- [8] H. Brosenne, M. Homeister and St. Waack, Graph-driven free parity BDDs: Algorithms and lower bounds, in *Proc. of MFCS*. Springer, *Lecture Notes in Comput. Sci.* 2136 (2001) 212-223
- [9] R.E. Bryant, Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.* 35 (1986) 677-691.
- [10] J. Gergov and C. Meinel, Frontiers of feasible and probabilistic feasible boolean manipulation with branching programs, in *Proc. of STACS*. Springer, *Lecture Notes in Comput. Sci.* 665 (1993) 576-585.
- [11] J. Gergov and C. Meinel, Efficient boolean manipulation with OBDDs can be extended to FBDDs. IEEE Trans. Comput. 43 (1994) 1197-1209.
- [12] J. Hromkovič, Communication Complexity and Parallel Computing. Springer (1997).
- [13] M. Krause, BDD-based cryptanalysis of keystream generators, in Proc. of EUROCRYT (2002) 222-237.
- [14] E. Kushilevitz and N. Nisan, Communication Complexity. Cambridge University Press (1997).
- [15] J. Jain, J. Bitner, D.S. Fussell and J.A. Abraham, Functional partitioning for verification and related problems. Brown/MIT VLSI Conference (1992) 210-226.
- [16] E.I. Nechiporuk, On a boolean function. Soviet Math. Dokl. 7 (1966) 999-1000.
- [17] A.A. Razborov, Lower bounds for deterministic and nondeterministic branching programs, in Proc. of FCT. Springer, Lecture Notes in Comput. Sci. 529 (1991) 47-60.
- [18] P. Savický and D. Sieling, A hierarchy result for read-once branching programs with restricted parity nondeterminism, in Proc. of 25th MFCS. Springer, Lecture Notes in Comput. Sci. 1893 (2000) 650-659.
- [19] P. Savický and S. Žák, A read-once lower bound and a (1,+k)-hierarchy for branching programs. Theoret. Comput. Sci. 238 (2000) 347-362.
- [20] D. Sieling and I. Wegener, I. (1995). Graph driven BDDs A new data structure for boolean functions. Theoret. Comput. Sci. 141 (1995) 283-310.
- [21] D. Sieling and I. Wegener, A comparison of free BDDs and transformed BDDs. Formal Meth. System Design 19 (2001) 223-236.
- [22] J. Thathachar, On separating the read-k-times branching program hierarchy, in *Proc. of 30th Ann. ACM Symposium on Theory of Computing (STOC)* (1998) 653-662.
- [23] I. Wegener, The Complexity of boolean Functions. Wiley-Teubner (1987).
- [24] I. Wegener, Branching Programs and Binary Decision Diagrams Theory and Applications. SIAM Monographs on Discrete Mathematics and Applications (2000).
- [25] P. Woelfel, A lower bound technique for restricted branching programs and applications, in Proc. of 19th STACS. Springer, Lecture Notes in Comput. Sci. 2285 (2002) 431-442.

Communicated by J. Hromkovič.

Received May, 2002. Accepted January, 2003.

To access this journal online: www.edpsciences.org