Journal on Satisfiability, Boolean Modeling and Computation 7 (2010) 35-58

Ordered Binary Decision Diagrams, Pigeonhole Formulas and Beyond[*]

Olga Tveretina olga.tveretina@kit.edu
Carsten Sinz carsten.sinz@kit.edu
Institute for Theoretical Computer Science
Karlsruhe Institute of Technology (KIT)
Germany
Hans Zantema h.zantema@tue.nl
Department of Computer Science, TU Eindhoven
Institute for Computing and Information Sciences, Radboud University, Nijmegen
The Netherlands

Abstract

Groote and Zantema proved that a particular OBDD computation of the pigeonhole formula has exponential size, and that limited OBDD derivations cannot simulate resolution polynomially. Here we show that an arbitrary OBDD refutation of the pigeonhole formula has exponential size: we prove that for any order of computation at least one intermediate OBDD in the proof has size $\Omega(1.14^n)$. We also present a family of CNFs that show an exponential blow-up for all OBDD refutations compared to unrestricted resolution refutations.

Keywords: *ordered binary decision diagrams, resolution, pigeonhole formulas, lower bounds*

Submitted October 2009; revised March 2010; published March 2010

1. Introduction

The reason for this study comes from the interest in giving theoretical explanations of the efficiency of algorithms for satisfiability testing. Many of these algorithms are based either on resolution or on ordered binary decision diagrams (OBDDs).

The resolution rule in propositional logic is a single valid inference rule that produces a new clause implied by two clauses containing complementary literals [11]. This technique uses proof by contradiction and is based on the fact that any sentence in propositional logic can be transformed into an equivalent sentence in Conjunctive Normal Form (CNF).

Presently, many of the state-of-the-art satisfiability solvers are based on the DPLL procedure, which is a variant of resolution in combination with search. At the same time, resolution based solvers can be highly inefficient for solving some structured problems and require time exponential in the size of an input instance. The most famous example of such

[*] This work was supported in part by the "Concept for the Future" of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

CNF is the pigeonhole formula that formalizes a very simple principle that $n+1$ objects cannot be placed into n holes.

An OBDD, also referred to as a Reduced OBDD (ROBDD) or just a BDD, is a data structure that is used to represent Boolean functions [2, 19]. OBDDs have some interesting properties: they provide compact and canonic representations of Boolean functions, and there are efficient algorithms for performing logical operations on OBDDs. As a result, OBDDs have been successfully applied to a wide variety of tasks, particularly in VLSI design and CAD verification.

The OBDD approaches for SAT solving can be divided into two groups:

(1) The first group is based on using the Apply operator (*join* rule) to build an OBDD for a conjunction of clauses. Thus, for a given order on variables, an OBDD for the CNF is built, which is then checked for equality to the terminal node 0.

(2) The second group utilizes *symbolic quantifier elimination* and allows, besides using the Apply operation, to eliminate variables via existential quantification. Allowing existential quantification can lead to significant speed-ups for certain kinds of structured instances. E.g., it is known that there are proofs of polynomial size for the pigeonhole principle using this proof system [3].

A proof system based on OBDDs was proposed by Atserias *et al.* [1]. The authors introduce a very general proof system based on constraint propagation. OBDDs are a special case of this proof system. Their proof system has four rules: Axiom, Join, Projection, and Weakening. The first two rules, Axiom and Join, correspond to an application of the Apply operator. Projection and Weakening are introduced to reduce the size of intermediate OBDDs. The Projection rule corresponds to an application of existential quantification. Hence, this proof system contains lines that are OBDDs derived by any of the above rules. It was shown that the OBDD proof system containing all four rules is strictly stronger than resolution [1] but it is still exponential [8].

It was proven for the first time in [16] that OBDD proof systems with the two rules Axiom and Join, corresponding to the Apply method, have an exponential lower bound on refutations of the pigeonhole formula. However, the lower bound $\Omega(1.14^n)$ presented in this paper is stricter in comparison with $\Omega(1.025^n)$ in [16]. We also demonstrate a family of CNFs that requires exponential increase for all OBDD refutations based on Apply method, i.e. OBDD refutations without existential quantification, to simulate unrestricted resolution refutation. The formulas are the pigeonhole formulas extended with additional clauses as in [4]. These formulas are CNFs parameterized by n and have size $O(n^3)$. Cook has shown that there is a resolution refutation for these formulas of size $O(n^4)$ [4]. We show that an arbitrary OBDD Apply refutation has size $2^{\Omega(n)}$.

Related work. There has been a lot of research on the relation of different propositional proof systems [5, 18] and, in particular, on the relation of different forms of resolution and OBDDs [9, 15, 17].

In [6] Groote and Zantema proved that limited OBDD derivations cannot simulate resolution refutations polynomially. The considered OBDD system joins the clauses of a CNF in the order as they are listed, following the shape of the formula, i.e. to build the OBDD for $C_1 \wedge (C_2 \wedge C_3)$, first the OBDD for $C_2 \wedge C_3$ is built and then the one for $C_1 \wedge (C_2 \wedge C_3)$. They present a lower bound for refutations of a formula of the form $\neg x \wedge (x \wedge \varphi)$, where φ is a formula that is hard for both OBDDs and resolution. But this formula is refuted trivially if we proceed as $(\neg x \wedge x) \wedge \varphi$.

In [3] a direct construction of polynomial size OBDD refutation of pigeonhole formulas in presence of existential quantification is presented. Another interesting result by Segerlind in [13] is that the OBDD derivations with the Axiom rule, a tree-like application of the Join rule and the Projection rule cannot efficiently simulate DAG-like resolution derivations.

Contribution. Our result differs from previous work in various ways. We strengthen the result of [6]. In [6] the only OBDD computation of the pigeonhole formulas considered that first computes the conjunction of all positive clauses, then the conjunction of all negative clauses, and finally the conjunction of these two. In our setting, the clauses of the pigeonhole formula may be processed in any arbitrary order. We show that for any OBDD refutation of the pigeonhole formula some of the intermediate OBDDs have size exponential in n. A consequence of our result is that the gap between polynomial and exponential in the OBDD refutation framework for pigeonhole formula is caused by existential quantification, i.e. by the Projection rule.

The difference with respect to [13] and [3] is the following. We consider a weaker OBDD proof system containing only two rules, Axiom and Join. For this proof system we show that an unrestricted application of it cannot simulate resolution polynomially. At present it is not known whether there is an exponential separation between tree-like and DAG-like OBDD proof systems based on the Apply method. Therefore, we cannot say whether a tree-like proof system from [13] subsumes the OBDD proof system considered in this paper. Still a direct proof of exponential separation between an unrestricted OBDD Apply proof system and unrestricted resolution is presented for the first time in this paper. Moreover, although for a weaker proof system, we quantitatively improve the lower bounds on OBDD refutations presented in [12, 13].

2. Propositional proof systems

We consider propositional formulas in *Conjunctive Normal Form* (CNF). Basic blocks for building CNFs are propositional variables that take the values false or true. The set of propositional variables is denoted by Var. A literal is either a variable x or its negation $\neg x$. A clause is a disjunction of literals, and a CNF is a conjunction of clauses. By $\perp$ we denote the empty clause. In the following, for convenience, we consider clauses as sets of variables, and a CNF as a set of clauses.

By $\mathsf{Cls}(\varphi)$ we denote the set of clauses contained in a CNF φ and by $\mathsf{Var}(\varphi)$ we denote the set of variables contained in the CNF φ. By $\mathsf{A} : \mathsf{Var} \to \{\mathsf{true}, \mathsf{false}\}$ we denote a function that assigns variables either to true or to false. We write $F \models_{\mathsf{A}}$ true if a CNF F takes a value true for an assignment A and $F \models_{\mathsf{A}}$ false if F takes a value false.

2.1 Resolution

The resolution principle, due to Robinson [11], is a method to construct proofs by contradiction. The resolution rule produces a new clause implied by two clauses containing complementary literals. The resulting clause contains all literals except the complementary ones. Formally this can be presented as following.

$$\textbf{Resolution:} \quad \frac{C \cup \{l\} \quad D \cup \{\neg l\}}{C \cup D}$$

Thus, from clauses $C \cup \{l\}$ and $D \cup \{\neg l\}$ a new clause $C \cup D$ is derived. A clause $C \cup D$ is called a resolvent of $C \cup \{l\}$ and $D \cup \{\neg l\}$. The resolution proof rule defines a proof system in which there are no axiom schemata, and only one proof rule, resolution. The proofs by resolution start with clauses of the input CNF and derive new clauses until a contradiction which is expressed as the empty clause is obtained.

Definition 1 (Resolution refutation)**.** *A resolution refutation of an unsatisfiable CNF φ is a sequence of CNFs $\varphi \equiv \varphi_0, \varphi_1, \ldots, \varphi_n$ with the following properties.*

- $\varphi_i \equiv \varphi_{i-1} \cup \{C_i\}$, $i = 1, \ldots, n$, *where C_i is a resolvent of two clauses in φ_{i-1}.*
- $\bot \in \varphi_n$ *and* $\bot \notin \varphi_i$ *for* $i = 0, \ldots, n-1$.

We say that n is the size of the resolution refutation.

2.2 OBDDs as a proof system

A binary decision diagram (BDD) is a a rooted, directed, acyclic graph, which consists of decision nodes and two terminal nodes 0 and 1. Each decision node is labeled by a propositional variable from Var and has two child nodes called a low child and a high child. The edge from a node to a low (high) child represents an assignment of the variable to 0 (1). Such a BDD is called an *ordered* BDD (OBDD) if different variables appear in the same order on all paths from the root. Therefore, OBDDs assume that there is a total order $\prec$ on the set of variables, and every node in the OBDD is less then its children with respect to this order.

An OBDD is said to be *reduced* if the following two rules are not applicable: 1) merge isomorphic subgraphs; 2) eliminate any node whose two children are isomorphic. We assume all our OBDDs to be reduced.

These OBDDs have the following property: For a fixed order $\prec$ on the set of variables, every propositional formula φ is uniquely represented by an OBDD $\mathsf{B}(\varphi, \prec)$. Together with the efficient computation, this unicity is the main property to be exploited in BDD technology. In particular, two formulas φ and ψ are equivalent if and only if $\mathsf{B}(\varphi, \prec) = \mathsf{B}(\psi, \prec)$.

Given a propositional formula φ and an order on variables $\prec$, we define the size of an OBDD $\mathsf{B}(\varphi, \prec)$ representing φ with respect to $\prec$ as the number of its internal nodes and denote it by $\mathsf{size}(\mathsf{B}(\varphi, \prec))$.

In this paper we consider OBDDs as a propositional proof system. Since we are dealing only with unsatisfiable CNFs, we give a definition of an OBDD refutation adapting the definition from [3].

Definition 2 (OBDD refutation)**.** *Given a total order on variables $\prec$, an OBDD refutation of an unsatisfiable CNF φ is a sequence of OBDDs*

$$\mathsf{B}_1(\varphi_1, \prec), \ldots, \mathsf{B}_n(\varphi_n, \prec)$$

such that $\mathsf{B}_n(\varphi_n, \prec)$ is the OBDD representing the constant false*, and for each $\mathsf{B}_i(\varphi_i, \prec)$, $1 \leq i \leq n$, exactly one of the following holds:*

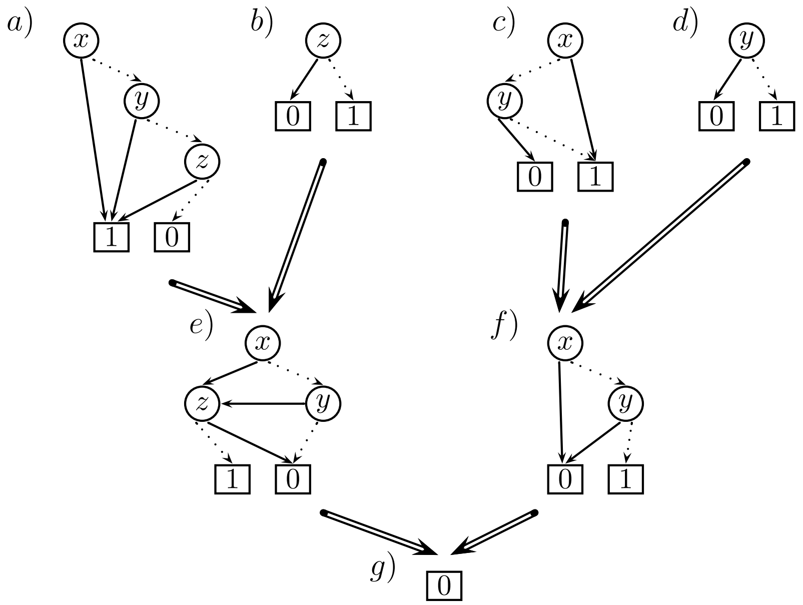

Figure 1. OBDD refutation of $\varphi \equiv (x \vee y \vee z) \wedge (\neg x \vee y) \wedge \neg y \wedge \neg z$ for the order on variables $x \prec y \prec z$.

- *(AXIOM)* $\mathsf{B}_i(\varphi_i, \prec)$ *represents one of the clauses* $C \in \varphi$*;*
- *(JOIN) there are OBDDs* $\mathsf{B}_{i'}(\varphi_{i'}, \prec)$ *and* $\mathsf{B}_{i''}(\varphi_{i''}, \prec)$ *such that* $1 \leq i' < i'' < i$ *and* $\varphi_i = \varphi_{i'} \wedge \varphi_{i''}$*.*

The size of the OBDD refutation is defined as $\sum_{i=1}^{n} \mathsf{size}(\mathsf{B}_i(\varphi_i, \prec))$*.*

When it is convenient, instead of $\mathsf{B}(\varphi, \prec)$ we write $\mathsf{B}(\varphi)$ or just B. By $\mathsf{Cls}(\mathsf{B}(\varphi))$ we mean the set of clauses and by $\mathsf{Var}(\mathsf{B}(\varphi))$ the set of variables contained in φ.

Example 1. *Figure 1 depicts an OBDD refutation of CNF* $\varphi \equiv (x \vee y \vee z) \wedge (\neg x \vee y) \wedge \neg y \wedge \neg z$ *for the order on variables* $x \prec y \prec z$*. OBDDs* $a) - d)$ *correspond to applications of* Axiom *rule and OBDDs* $e) - g)$ *correspond to applications of* Join *rule.*

The size of the OBDD representing a propositional formula F for a given order on variables $\prec$ is described by the structure theorem from [14].

Theorem 1 (Sieling and Wegener, 1993)**.** *Let* m_i*,* $i < n$*, be the number of subfunctions of a Boolean function* $f(x_i, \dots, x_n)$*, which are obtained by replacing the variables* $x_1, \dots, x_{i-1}$ *by constants and which depend essentially on* x_i *(a function* f *depends essentially on a variable* y *if* $f_{|y=0} \neq f_{|y=1}$*). Then the OBDD for* f *with respect to the order* $x_1 \prec x_2 \prec \cdots \prec x_n$ *contains exactly* m_i *nodes labelled* x_i *which are reached for the different subfunctions.*

The above observation is very simple and helpful to prove lower bounds. In this paper we use Theorem 2 which is a variant of Theorem 1 and was presented in [6]. We use $\mathbb{B} = \{0, 1\}$ to denote the set of Boolean constants.

Theorem 2. *Suppose for a given formula φ the following holds:*

- $|\mathsf{Var}(\varphi)| = n$*;*
- $\prec$ *is a total order on the set of variables* $\mathsf{Var}(\varphi)$*;*
- $x_1, \ldots, x_k$ *are the smallest* k *elements with respect to* $\prec$ *for some* $k < n$*;*
- $A \subseteq \{1, \ldots, k\}$*;*
- $z = (z^1, \ldots, z^k) \in \mathbb{B}^k$*.*
- *For all distinct* $\overrightarrow{x}_1, \overrightarrow{x}_2 \in \mathbb{B}^k$ *such that* $x_1^i = x_2^i = z^i$ *for all* $i \notin A$ *there exists a* $\overrightarrow{y} \in \mathbb{B}^{n-k}$ *such that* $\varphi(\overrightarrow{x}_1, \overrightarrow{y}) \neq \varphi(\overrightarrow{x}_2, \overrightarrow{y})$*.*

Then the size of the OBDD $\mathsf{B}(\varphi, \prec)$ *is at least* $2^{|A|}$*.*

The proof of the lower bounds presented in Section 4 is based on Theorem 2. However, in order to obtain a lower bound we still have to solve some combinatorial problems.

3. Pigeonhole formulas and beyond

The pigeonhole formulas is a family of unsatisfiable CNFs parameterized by n. They are often used as a standart benchmark for checking efficiency of (UN)SAT algorithms. It is very easy to give an argument for unsatisfiability of these formulas but most of the techniques need time exponential in n to produce a formal proof of unsatisfiability.

In our paper we consider also another class of unsatisfiable CNFs that we call as *extended pigeonhole formulas*. These formulas were introduced by Cook in his paper on the extended resolution proof of the pigeonhole formulas [4].

3.1 Pigeonhole formulas

The pigeonhole principle states that n holes can hold at most n objects with one object in a hole. The propositional formulas describing this principle were introduced as following. Atomic proposition P_{ij} says that i is mapped to j, and the set of clauses PHP_n states that there is a one-to-one map from the set $\{1, \ldots, n+1\}$ to the set $\{1, \ldots, n\}$.

Definition 3 (Pigeonhole Formulas)**.** *The pigeonhole formula* PHP_n*,* $n > 0$*, is defined as follows.*

$$\mathsf{PC}_n = \bigwedge_{i=1}^{n+1} [\bigvee_{j=1}^{n} P_{i,j}], \quad \mathsf{NC}_n = \bigwedge_{\substack{1 \le i < j \le n+1 \\ 1 \le k \le n}} [\neg P_{i,k} \vee \neg P_{j,k}],$$

$$\mathsf{PHP}_n = \mathsf{PC}_n \wedge \mathsf{NC}_n.$$

The formula PC_n states that at least one variable is true in all $n+1$ rows and the formula NC_n states that at most one variable is true in all n columns. These formulas were studied intensively in relation to complexity of different propositional proof systems, and in particular, it has been proved in [7] that every resolution proof for PHP_n has size exponential in n.

The variables of the pigeonhole formula can be seen as entries of a matrix with $n+1$ rows and n columns, where the variables are placed according to the indexes. We denote such a matrix by $\mathsf{Matrix}(\mathsf{PHP}_n)$. Then the i-th row corresponds to the clause $\bigvee_{j=1}^{n} P_{ij}$ and vice versa. Therefore, if it is needed, we can refer to a row as to a clause.

3.2 Extended pigeonhole formulas

Years before a proof of an exponential lower bound on resolution refutation for the pigeonhole formulas was found by Haken, Cook showed that there exists a short proof of PHP_n with extended resolution of size polynomial in n [4]. The idea of Cook was to define new variables Q_{ij} as $Q_{ij} \equiv P_{ij} \vee (P_{in} \wedge P_{n+1,j})$, $1 \le i \le n, 1 \le j \le n-1$ and to describe this equivalence by the set $\mathbb{Q}_n$ of the following clauses.

(1) $Q_{ij} \vee \neg P_{ij}$,

(2) $Q_{ij} \vee \neg P_{in} \vee \neg P_{n+1,j}$,

(3) $\neg Q_{ij} \vee P_{ij} \vee \neg P_{in}$,

(4) $\neg Q_{ij} \vee P_{ij} \vee \neg P_{n+1,j}$.

We rename the variables as follows: We denote P_{ij} by P_{ij}^0 and $P_{ij}^k \equiv P_{ij}^{k-1} \vee (P_{in}^{k-1} \wedge P_{n+1,j}^{k-1})$ for $1 \le k \le n-1$, $1 \le i \le n-k+1$, $1 \le j \le n-k$. Then using the idea of Cook, we can define extended pigeonhole formulas.

Definition 4 (Extended Pigeonhole Formulas)**.** *The extended pigeonhole formula* EPHP_n *for* $n > 1$ *is defined as* $\mathsf{EPHP}_n = \mathsf{PHP}_n \wedge \bigwedge_{i=1}^{4} \mathsf{EC}_n^i$*, where clauses* EC_n^i *are constructed as follows.*

(1) $\mathsf{EC}_n^1 = \bigwedge_{\substack{1\le k\le n-1,\\ 1\le i\le n-k+1,\\ 1\le j\le n-k}} [P_{ij}^k \vee \neg P_{ij}^{k-1}]$,

(2) $\mathsf{EC}_n^2 = \bigwedge_{\substack{1\le k\le n-1,\\ 1\le i\le n-k+1,\\ 1\le j\le n-k}} [P_{ij}^k \vee \neg P_{in}^{k-1} \vee \neg P_{n+1,j}^{k-1}]$,

(3) $\mathsf{EC}_n^3 = \bigwedge_{\substack{1\le k\le n-1,\\ 1\le i\le n-k+1,\\ 1\le j\le n-k}} [\neg P_{ij}^k \vee P_{ij}^{k-1} \vee P_{in}^{k-1}]$,

(4) $\mathsf{EC}_n^4 = \bigwedge_{\substack{1\le k\le n-1,\\ 1\le i\le n-k+1,\\ 1\le j\le n-k}} [\neg P_{ij}^k \vee P_{ij}^{k-1} \vee P_{n+1,j}^{k-1}]$.

The resulting EPHP_n formula has interesting properties. It is constructed by adding $4n(n-1)(n+1)/3$ new clauses to PHP_n. Hence, it is a simple unsatisfiable CNF with size polynomial in n. There is a resolution refutation of EPHP_n with size $O(n^4)$ [4]. But, as we prove in Section 5, all OBDD refutations of EPHP_n have size exponential in n. Moreover, for each OBDD refutation of EPHP_n there is a corresponding OBDD refutation of PHP_n such that lower bound on the OBDD proof of EPHP_n is not smaller than lower bound on the OBDD proof of PHP_n.

Theorem 3 (Cook)**.** *There is a resolution refutation of* EPHP_n*,* $n > 1$*, of size* $O(n^4)$*.*

We present here a proof of the above theorem because it is missing in the original paper and we think that it is of interest itself. In our proof we follow the idea from [4] that from EPHP_n one can derive the clauses PHP_{n-1} in $O(n^3)$ resolution steps.

Proof of Theorem 3. Let $\mathbb{P}_n$ be the set $\mathbb{Q}_n$ but after renaming the variables, i.e.

$$\mathbb{P}_n = \{P^1_{ij} \vee \neg P^0_{ij},\ \ P^1_{ij} \vee \neg P^0_{in} \vee \neg P^0_{n+1,j},\ \ \neg P^1_{ij} \vee P^0_{ij} \vee \neg P^0_{in},\ \ \neg P^1_{ij} \vee P^0_{ij} \vee \neg P^0_{n+1,j}\}.$$

The proof has the following steps.

(1) Show that $P^1_{i1} \vee \cdots \vee P^1_{i,n-1}$, $1 \leq i \leq n$, can be derived from PHP_n and the set of clauses $\mathbb{P}_n$ in $O(n)$ resolution steps.

(2) Show that $\neg P^1_{ik} \vee \neg P^1_{jk}$, $1 \leq i < j \leq n, 1 \leq k \leq n-1$, can be derived from PHP_n and the set of clauses $\mathbb{P}_n$ in $O(n^2)$ resolution steps.

After repeating the above steps $n-1$ times one produces the set of clauses PHP_1 from which the empty clause can be derived in two resolution steps. It results in a resolution refutation of size $O(n^4)$. The size of the refutation can be expressed alternatively as $O(N^{4/3})$, where N is a number of clauses in EPHP_n.

(1) We show how to derive $P^1_{i1} \vee \cdots \vee P^1_{i,n-2}$ from PHP_n and the set of clauses $\mathbb{P}_n$.

- (a) $P^1_{i1} \vee \cdots \vee P^1_{i,n-1} \vee P^0_{in}$ is derived from $P^0_{i1} \vee \cdots \vee P^0_{in}$ and $P^1_{ij} \vee \neg P^0_{ij}$, $1 \leq j \leq n-1$.
- (b) $P^1_{i1} \vee \cdots \vee P^1_{i,n-1} \vee \neg P^0_{n+1,j}$, $1 \leq j \leq n-1$, is derived from (a) and $P^1_{ij} \vee \neg P^0_{in} \vee \neg P^0_{n+1,j}$.
- (c) $\neg P^0_{i,n} \vee P^0_{n+1,1} \vee \cdots \vee P^0_{n+1,n-1}$ is derived from $P^0_{n+1,1} \vee \cdots \vee P^0_{n+1,n}$ and $\neg P^0_{in} \vee \neg P^0_{n+1,n}$.
- (d) $P^0_{n+1,1} \vee \cdots \vee P^0_{n+1,n-1} \vee P^1_{i1} \vee \cdots \vee P^1_{i,n-1}$ is derived from (a) and (c).
- (e) $P^1_{i1} \vee \cdots \vee P^1_{i,n-2}$ is derived from (b) and (d).

(2) We show how $\neg P^1_{ik} \vee \neg P^1_{jk}$ can be derived from PHP_n and the set of clauses $\mathbb{P}_n$ in $O(n^2)$ resolution steps.

- (a) $\neg P^1_{ik} \vee \neg P^1_{jk} \vee P^0_{n+1,k}$ is derived from $\neg P^0_{ik} \vee \neg P^0_{jk}$ and $\neg P^1_{ik} \vee P^0_{ik} \vee P^0_{n+1,k}$ and $\neg P^1_{jk} \vee P^0_{jk} \vee P^0_{n+1,k}$.
- (b) $\neg P^1_{ik} \vee \neg P^1_{jk} \vee \neg P^0_{ik}$ is derived from (a) and $\neg P^0_{ik} \vee \neg P^0_{n+1,k}$.
- (c) $\neg P^1_{ik} \vee \neg P^1_{jk} \vee \neg P^0_{jk}$ is derived from (a) and $\neg P^0_{jk} \vee \neg P^0_{n+1,k}$.
- (d) $\neg P^1_{ik} \vee \neg P^1_{jk} \vee P^0_{in}$ is derived from (b) and $\neg P^1_{ik} \vee P^0_{ik} \vee P^0_{in}$.
- (e) $\neg P^1_{ik} \vee \neg P^1_{jk} \vee P^0_{j,n}$ is derived from (c) and $\neg P^1_{ik} \vee P^0_{jk} \vee P^0_{jn}$.
- (f) $\neg P^1_{ik} \vee \neg P^1_{jk} \vee \neg P^0_{jn}$ is derived from (d) and $\neg P^0_{in} \vee \neg P^0_{jn}$.
- (g) $\neg P^1_{ik} \vee \neg P^1_{jk}$ is derived from (e) and (f).

Hence, we have shown the correctness of the theorem by presenting the resolution steps. □

4. Technical background

In this section we introduce notations and technical lemmas that will be used throughout the paper. Some combinatorial properties of square matrices are presented in Lemma 1. Lemma 2 generalizes a well-known fact about binary trees claiming the existence of subtrees with a weight lying between a and 2a for any definition of weight as a sum of the weights of its leaves.

4.1 Notations

Let $S_\prec$ denote a set containing the $\lfloor n^2/2 \rfloor$ smallest elements of $\mathsf{Var}(\mathsf{PC}_n^*)$, where $\prec$ is a given order on variables and PC_n^* is obtained from PC_n by removing an arbitrary clause. And $S_\succ = \mathsf{Var}(\mathsf{PHP}_n) \backslash S_\prec$. We denote by $S_\prec^*$ and by $S_\succ^*$ the following sets:

$$S_\prec^* = \{P_{ab} \in \mathsf{Var}(\mathsf{PHP}_n) \mid P_{ab} \preceq \max_{P_{cd} \in S_\prec} P_{cd}\} \text{ and } S_\succ^* = \mathsf{Var}(\mathsf{PHP}_n) \backslash S_\prec^*.$$

Suppose $\mathsf{B}_1, \ldots, \mathsf{B}_l$ is an OBDD refutation on PHP_n. Then for each B_i we define

$$S_\prec^i = S_\prec^* \cap \mathsf{Var}(\mathsf{B}_i) \text{ and } S_\succ^i = \mathsf{Var}(\mathsf{B}_i) \backslash S_\prec^i.$$

Moreover, we define

$$\mathsf{Cls}^{neg}(\mathsf{B}_i) = \mathsf{Cls}(\mathsf{B}_i) \cap \mathsf{Cls}(\mathsf{NC}_n) \text{ and } \mathsf{Cls}^{pos}(\mathsf{B}_i) = \mathsf{Cls}(\mathsf{B}_i) \cap \mathsf{Cls}(\mathsf{PC}_n).$$

4.2 Technical lemmas

Lemma 1 was presented for the first time in [16], but with a smaller coefficient $c = \frac{1}{2} - \frac{1}{4}\sqrt{2} \approx 0.146$. This lemma is of interest from a point of view of Ramsey Theory that typically asks questions of the form: How many elements of some structure must there be to guarantee that a particular property will hold?

Groote and Zantema in [6] considered an $n \times m$ matrix containing entries equally colored white and black and proved that such a matrix has either $\sqrt{2}(n-1)/2$ rows or $\sqrt{2}(m-1)/2$ columns containing both a black and a white entry. Lemma 1 presents another combinatorial property of a matrix containing entries equally colored white and black. In comparison with [16] we present another proof that gives us a better $c = \frac{3}{4} - \frac{1}{4}\sqrt{5} \approx 0.19098$.

Lemma 1. *Consider a matrix $M = \{m_{ij}\}$, $1 \le i \le n$, $1 \le j \le n$. Let the matrix entries be colored equally white and black, i.e. the difference between the number of white entries and the number of black entries is at most one. Let $m = \lfloor cn \rfloor$ for $c = \frac{3}{4} - \frac{1}{4}\sqrt{5} \approx 0.19098$. Then at least one of the following holds.*

- *One can choose m rows, and in every of these rows a white and a black entry, such that all these $2m$ entries are in different columns.*
- *One can choose m columns, and in every of these columns a white and a black entry, such that all these $2m$ entries are in different rows.*

Proof. Starting by the given matrix repeat the following process as long as possible.

> Choose a row in the matrix containing both a white and a black entry and not chosen previously. Remove both the column containing the white entry and the column containing the black entry.

Assume this repetition stops after k steps. Write $x = k/n$. If $x \geq c$ the first property of the lemma holds and we are done. In the remaining case we have $x < c$. We assume that the second property of the lemma does not hold, and then we will derive a contradiction.

The remaining matrix M' consists of n rows and $n(1-2x)$ columns. The xn chosen rows in M' can be either mixed or monochromatic, and the other $n - xn$ rows consist either only of white entries or only of black entries (otherwise the process of choosing rows could continue).

We denote by $R(M')$ the set of the xn rows chosen by the above process and contained in the remaining matrix M', and we denote by $R'(M')$ the set of $n(1-x)$ rows that were not chosen and that are also contained in M'.

Assume that in $R'(M')$, pn of the rows are totally white and qn of the rows are totally black. Then $p + q = 1 - x$, where all the numbers p, q, x are reals in the interval $[0, 1]$.

Assume that in $R(M')$, there are in total axn^2 white entries and bxn^2 black entries, where a, b are real numbers in the interval $[0, 1]$. It is easy to see that such a and b exist since the total number of the entries in $R(M')$ is less than xn^2. Since the total number of the entries in $R(M')$ is $(a + b)xn^2 = (1 - 2x)xn^2$, we obtain $a + b = 1 - 2x$.

The total number of white entries in the remaining matrix M' is $p(1 - 2x)n^2 + axn^2$. This is strictly less than $n^2/2$ since at least one row was chosen. So

$$p(1-2x) + ax < \frac{1}{2},$$

and similarly $q(1 - 2x) + bx < \frac{1}{2}$ for the black entries.

Now assume that $q \geq c$ and $p + a \geq c$. We will construct at least $m = \lfloor cn \rfloor$ columns in M' satisfying the second property of the lemma. For the first an choose a white entry from a mixed row and a black entry in the same column from a full black row. This can be repeated at least $l = \min(an, qn)$ times. If $l = qn$, we are done. If $l < qn$, the process is continued by choosing pn entries from the full white rows. Since $q \geq c$ and $p + a \geq c$ we have chosen at least cn columns in this way, yielding the second property of the lemma. Since we assume this second property does not hold, we conclude

$$q < c \vee p + a < c.$$

By symmetry we similarly obtain $p < c \vee q + b < c$. Since the combination of $q < c$ and $p < c$ can not occur due to $x < c < .2$ and $p + q = 1 - x$, we either have $p + a < c$ or $q + b < c$. By symmetry we may assume without loss of generality that $p + a < c$. Now substituting $b = 1 - 2x - a$ and $q = 1 - x - p$ in $q(1 - 2x) + bx < \frac{1}{2}$ we obtain

$$(1-x-p)(1-2x) + (1-2x-a)x < \frac{1}{2}$$

hence

$$1 - p + (2p - a - 2)x < \frac{1}{2}.$$

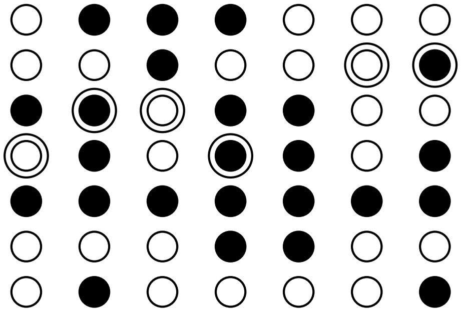

Figure 2. An example of a 7×7 matrix with entries equally colored black and white.

Since $x < c$ and $2p - a - 2 < 0$ (the latter since $p < 1$ and $a \geq 0$), we conclude

$$1 - p + (2p - a - 2)c < \frac{1}{2}.$$

Since $p + a < c$ we conclude

$$1 - p + (3p - c - 2)c < \frac{1}{2}.$$

Hence, $1 - c^2 - 2c - p(1 - 3c) < \frac{1}{2}$. Since $c > p$ and $1 - 3c > 0$, this yields

$$\frac{1}{2} = 2c^2 - 3c + 1 = 1 - c^2 - 2c - c(1 - 3c) < \frac{1}{2},$$

contradiction, using $c = \frac{3}{4} - \frac{1}{4}\sqrt{5}$. □

By fine-tuning the argument the constant c in Lemma 1 can be improved. We conjecture that it also holds for $c = 1 - \frac{1}{2}\sqrt{2} \approx 0.293$. Choosing the $n \times n$ matrix in which the left upper $k \times k$-square is black for $k \approx \frac{n}{\sqrt{2}}$ and the rest is white, one observes that this value will be sharp. As our main result involves an exponential lower bound, we do not focus on the precise optimal value of c.

Example 2. *Consider a square 7×7 matrix with 24 black and 25 white entries as depicted in Figure 2. For this example there are three rows such that one can pick up one black and one white entry in each row in such a way that all entries are in different columns. At the same time Lemma 1 gives us much lower but a guaranteed bound.*

The intuition behind Lemma 1 and how it will be used in the subsequent proof is as follows: The matrix elements correspond one-to-one to propositional variables of the PHP_n formula where the last positive clause is dropped. For a given order $\prec$, the colors black and white correspond to variables in the upper and lower part of an OBDD. As Lemma 1 covers all possible colorings of the matrix, it is applicable to all possible orders $\prec$. Then, depending on the order of variables, we either apply Lemma 3 or Lemma 4 to obtain an intermediate OBDD (containing a subset of all PHP_n clauses) with a given property. For each variable in the upper part of the OBDD we can find a variable in the lower part that influences the truth value. Thus, we can apply Theorem 2.

The OBDD representing an unsatisfiable CNF is just a terminal node 0. Therefore, we have to show that for an arbitrary order on variables and an arbitrary way to combine clauses there is an intermediate OBDD of a size exponential in n. Hence, we start by the simple observations describing some properties of intermediate OBDDs. And the following lemma generalizes a well-known fact about binary trees claiming the existence of subtrees with a weight lying between a and 2a.

Lemma 2. *Let C be a finite set, $R \subseteq C$ with $|R| \geq 2$, and $B_1, \dots, B_l \subseteq C$ a sequence with:*

1. *$B_l = C$*
2. *For each B_i ($1 \leq i \leq l$), either $B_i = \emptyset$, $B_i = \{c\}$ for $c \in C$, or $B_i = B_j \cup B_k$ for some j, k with $j < k < i$.*

Then, for each a with $\frac{1}{|R|} < a \leq \frac{1}{2}$, there is a $j < l$ such that

$$a|R| \leq |B_j \cap R| < 2a|R|.$$

Proof. We give a proof by contradiction. Suppose, for each B_j, either $|B_j \cap R| < a|R|$ or $|B_j \cap R| \geq 2a|R|$.

As $B_l \cap R = C \cap R = R$, the inequality $|B_l \cap R| \geq 2a|R|$ holds for the final element B_l of the sequence. On the other hand, for singletons $B_j = \{c\}$, we have $|B_j \cap R| = 0 < a|R|$ for $c \notin R$, and $|B_j \cap R| = 1 < a|R|$ for $c \in R$, as $a > 1/|R|$. Moreover, for $B_i = \emptyset$, $|B_i \cap R| < a|R|$ obviously holds. Following now the predecessors of B_l (via the construction by set union) in the sequence B_i backwards, we finally arrive at an index k for which the following holds:

- $|B_k \cap R| \geq 2a|R|$, and
- $B_k = B_{k'} \cup B_{k''}$, where $|B_{k'} \cap R| < a|R|$ and $|B_{k''} \cap R| < a|R|$.

As $B_k \cap R = (B_{k'} \cup B_{k''}) \cap R = (B_{k'} \cap R) \cup (B_{k''} \cap R)$, and thus $|B_k \cap R| \leq |B_{k'} \cap R| + |B_{k''} \cap R| < 2a|R|$, we arrive at a contradiction to $|B_k \cap R| \geq 2a|R|$. □

Lemma 3. *Suppose $\mathsf{B}_1, \dots, \mathsf{B}_l$ is an OBDD refutation either on PHP_n or on EPHP_n and $R \subseteq \mathsf{Cls}(\mathsf{PC}_n)$ with $|R| \geq 4$. Then there is an $i < l$ such that*

$$|R|/4 \leq |\mathsf{Cls}(B_i) \cap R| < |R|/2.$$

Proof. Follows directly from Lemma 2.

□

Let $\mathsf{B}_1, \dots, \mathsf{B}_l$ be an OBDD refutation either on PHP_n or on EPHP_n. For each $i \leq l$, we define J_i as follows:

$$J_i = \{j \in \{1, \dots, n\} \mid \exists a, b : \neg P_{aj} \vee \neg P_{bj} \in \mathsf{Cls}(\mathsf{B}_i) \ \& \ P_{aj} \in S_{\prec} \ \& \ P_{bj} \in S_{\succ}\}.$$

Lemma 4. *Suppose $\mathsf{B}_1, \dots, \mathsf{B}_l$ is an OBDD refutation either on PHP_n or on EPHP_n for a total order on variables $\prec$. Let $G \subseteq \{1, \dots, n\}$ such that $|G| \geq 4$. Then there is an $i < l$ such that*

$$|G|/4 \leq |J_i \cap G| < |G|/2.$$

Proof. Follows from Lemma 2, using $C = \{1, \dots, n\}$, $R = G$, $a = 1/4$, and $J_1, \dots, J_l$ for the sequence $(B_i)_{1 \leq i \leq l}$, for which the precondition of Lemma 2 holds, as is easily checked. □

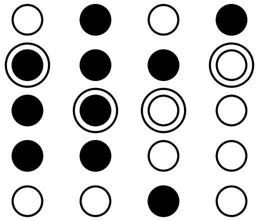

Figure 3. A 5×4 matrix for PHP_5. The black and the white entries represent elements from the sets $S_{\prec}$ and $S_{\succ}$ correspondingly.

5. Exponential lower bound on OBDD refutations of PHP_n and EPHP_n

In this section we prove lower bounds on OBDD refutations of the pigeonhole formula PHP_n and related extended pigeonhole formula EPHP_n. We start by proving lower bound for PHP_n and the proof of lower bound for EPHP_n is a direct consequence of it.

5.1 Lower bound on OBDD refutations of PHP_n

Our proof of lower bound on OBDD refutations of PHP_n is based on Theorem 2 and Lemmas 1-4. Before presenting the details of a formal proof we start with an example to give some intuition behind it.

Example 3. Let us consider PHP_4. This formula can be presented with a 5×4 matrix, as for example in Figure 3.

Suppose one of the intermediate OBDDs is an OBDD depicted in Figure 4 and it represents

$$\bigwedge_{i=2}^{3}[\bigvee_{j=1}^{4} P_{ij}] \wedge [\neg P_{24} \vee \neg P_{34}],$$

where $P_{21} \prec P_{31} \prec P_{32} \prec P_{22} \prec P_{23} \prec P_{33} \prec P_{24} \prec P_{34}$.

Our proofs of lower bounds on OBDD refutations are based on Theorem 2. Hence, we need to choose set A satisfying the theorem conditions. For this we use Lemma 1. The black and white entries represent elements of sets $S_{\prec}$ and $S_{\succ}$ correspondingly. We collect the black entries satisfying Lemma 1 in A. The white entries satisfying Lemma 1 are used to prove the conditions of Theorem 2.

We apply Lemma 1 and Theorem 2 to this example and collect the variables P_{21} and P_{32} in A. According Theorem 2 the size of the OBDD is at least $2^{|\{P_{21},P_{32}\}|} = 4$. For this particular example the size of the OBDD is much larger. This raises an open question whether lower bounds presented in this paper can be improved.

Lemma 5. *Let $\mathsf{B}_1, \ldots, \mathsf{B}_l$ be an OBDD refutation of PHP_n and $\prec$ be an order on variables. Assume that there are two sets, a set R of rows and a set S^R of entries of $\mathsf{Matrix}(\mathsf{PHP}_n)$ such that the following holds:*

- *For each $r \in R$ there are $P_{ra}, P_{rb} \in S^R$ such that $P_{ra} \in S_{\prec}$ and $P_{rb} \in S_{\succ}$.*
- *For distinct $P_{ab}, P_{cd} \in S^R$, $b \neq d$.*

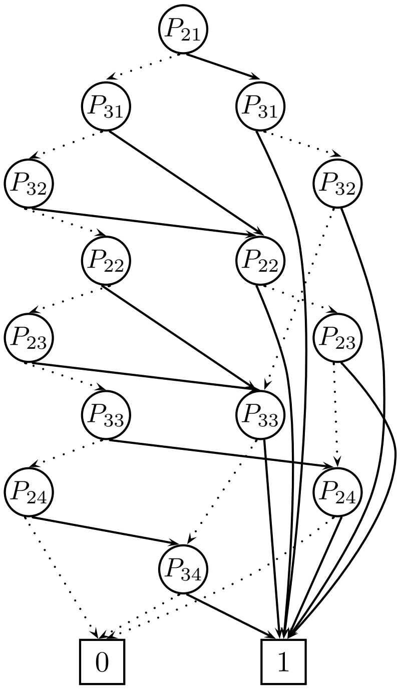

Figure 4. An OBDD for $\bigwedge_{i=2}^{3}[\bigvee_{j=1}^{4} P_{ij}] \wedge [\neg P_{24} \vee \neg P_{34}]$, where $P_{21} \prec P_{31} \prec P_{32} \prec P_{22} \prec P_{23} \prec P_{33} \prec P_{24} \prec P_{34}$.

Then there is an $i < l$ such that

$$\mathsf{size}(\mathsf{B}_i) \geq 2^{|R|/4}.$$

Proof. Let for $1 \leq i \leq l$,

$$R^i = \mathsf{Cls}(\mathsf{B}_i) \cap R.$$

We apply Lemma 3. Thus we know that there is an $i < l$ such that

$$|R|/4 \leq |R^i| < |R|/2,$$

and we get

$$2|R^i| + 1 \leq |R|.$$

Since for each $C \in \mathsf{Cls}^{pos}(\mathsf{B}_i)$, either $C \in R^i$ or $C \in \mathsf{PC}_n$ and $|\mathsf{PC}_n| = n + 1$, we compute

$$\begin{array}{rcl} |\mathsf{Cls}^{pos}(\mathsf{B}_i)| & \leq & (n+1) - (|R| - |R^i|) \\ & \leq & (n+1) - ((2|R^i| + 1) - |R^i|) \\ & = & n - |R^i|. \end{array}$$

We denote $\overline{R^i} = \mathsf{Cls}^{pos}(\mathsf{B}_i) \backslash R^i$. By definition $R^i \subseteq \mathsf{Cls}^{pos}(\mathsf{B}_i)$. Hence, we obtain

$$\begin{array}{rcl} |\overline{R^i}| & = & |\mathsf{Cls}^{pos}(\mathsf{B}_i)| - |R^i| \\ & \leq & n - 2|R^i|. \end{array}$$

For each row $r \in R^i$ we fix an entry that is in the set $S_{\prec}$. We collect these elements in the set A. For each row $r \in R^i$ we also fix an entry that is in $S_{\succ}$ and collect these elements in the set Y. Suppose

$$R^c = \{j \mid \exists i : P_{ij} \in A \cup Y\}.$$

Since the set of rows R^i satisfies Lemma 1, we get

$$|R^c| = 2|R^i|.$$

Let $J = n - |R^c|$. Then we obtain

$$J = n - 2|R^i|$$

and

$$|\overline{R^i}| \leq |J|.$$

Taking into account $|\overline{R^i}| \leq |J|$, for each row in $\overline{R^i}$ we fix one entry, collect these entries in the set X. We require the following.

- for distinct $P_{ab}, P_{cd} \in X$, $b \neq d$;
- for each $P_{ab} \in X$, $b \notin R^c$.

We define

$$X_{\prec} = S^*_{\prec} \cap X, \text{ and } X_{\succ} = S^*_{\succ} \cap X.$$

We apply Theorem 2 on

$$k = |S^i_{\prec}|,$$

where $S^i_{\prec} = S^*_{\prec} \cap \mathsf{Var}(\mathsf{B}_i)$. Let for $j = 1, \ldots, k$,

$$z_j = \begin{cases} 1, & \text{if } z_j \in X_{\prec} \\ 0, & \text{otherwise} \end{cases}$$

Choose distinct $\overrightarrow{x}, \overrightarrow{x}' \in \mathbb{B}^k$ such that $x_j = x'_j = z_j$ for all $z_j \notin A$. Then there is j' such that $x_{j'} \neq x'_{j'}$. Let $\overrightarrow{y} = (y_{k+1}, \ldots, y_q)$, where $q = |\mathsf{Var}(B_i)|$, be the vector defined for $y_j \in Y$ by

$$y_j = \begin{cases} 0, & \text{if } y_j \text{ is in the same row as } x_{j'} \\ 1, & \text{otherwise} \end{cases}$$

and for $y_j \notin Y$ by

$$y_j = \begin{cases} 1, & \text{if } y_j \in X_{\succ} \\ 0, & \text{otherwise} \end{cases}$$

Hence, the subset of clauses represented by B_i evaluates to $x_{j'}$ for the assignment $(\overrightarrow{x}, \overrightarrow{y})$ and to $x'_{j'}$ for the assignment $(\overrightarrow{x}', \overrightarrow{y})$. Taking into account that $|A| \geq |R|/4$, by Theorem 2, we obtain

$$\mathsf{size}(\mathsf{B}_i) \geq 2^{|A|} \geq 2^{|R|/4}.$$

□

Lemma 6. *Let $B_1, \ldots, B_l$ be an OBDD refutation of PHP_n and $\prec$ be a given order on variables. Assume that there is a set Q of columns and a set S^Q of entries of $\mathsf{Matrix}(\mathsf{PHP}_n)$ such that the following holds:*

- *For each $q \in Q$ there are $P_{aq}, P_{bq} \in S^Q$ such that $P_{aq} \in S_\prec$ and $P_{bq} \in S_\succ$.*
- *For distinct $P_{ab}, P_{cd} \in S^Q$, $a \neq c$.*

Then there is an $i < l$ such that

$$\mathsf{size}(B_i) \geq 2^{|Q|/4}.$$

Proof. Let

$$Q_i^c = \{j \mid \exists a, b : \neg P_{aj} \vee \neg P_{bj} \in \mathsf{Cls}(\mathsf{B}_i) \ \& \ P_{aj} \in S_\prec \ \& \ P_{bj} \in S_\succ\}.$$

By Lemma 4, there is an $i < l$ such that

$$|Q|/4 \leq |Q^c| < |Q|/2.$$

For each column in Q^c we fix one entry that is in the set $S_\prec$ and collect these elements in A. For each column in Q^c we also fix one entry that is in the set $S_\succ$ and collect these elements in the set Y. Let

$$Q^r = \{i \mid \exists j : P_{ij} \in A \cup Y\}.$$

Suppose

$$\overline{Q^c} = Q \backslash Q_i^c.$$

Then we get

$$\overline{Q^c} > |Q|/2.$$

For each $j \in \overline{Q^c}$ we fix $P_{a_j j}, P_{b_j j} \in S^Q$, where $P_{a_j j} \in S_\prec$ and $P_{b_j j} \in S_\succ$. We collect $P_{a_j j}$ in $X_\prec$ and we collect $P_{b_j j}$ in $X_\succ$ for all $j \in \overline{Q^c}$. We define

$$\overline{Q^r} = \{a \mid \exists b : P_{ab} \in X_\prec \cup X_\succ\}.$$

By Lemma 1 all entries collected in $\overline{Q^r}$ are from different rows. Hence, we obtain

$$|\overline{Q^r}| = 2|\overline{Q^c}|.$$

Taking into account that $\overline{Q^c} > |Q|/2$, we get

$$\overline{Q^r} > |Q|$$

and since $\overline{Q^r}$ is a natural number we get

$$\overline{Q^r} \geq |Q| + 1.$$

We denote

$$Q^* = \mathsf{Cls}^{pos}(\mathsf{B}_i) \backslash \overline{Q^r}.$$

No restrictions are posed on the size of the set $\mathsf{Cls}^{pos}(\mathsf{B}_i)$. Hence,

$$1 \leq |\mathsf{Cls}^{pos}(\mathsf{B}_i)| \leq n + 1.$$

We take into account that $|\overline{Q^r}| \geq |Q| + 1$ and compute

$$\begin{array}{rcl} |Q^*| & \leq & (n+1) - |\overline{Q^r}| \\ & \leq & (n+1) - (|Q|+1) \\ & = & n - |Q|. \end{array}$$

We define $J = \{j \mid \exists a : P_{aj} \in \mathsf{Var}(\mathsf{PHP}_n) \;\&\; j \notin Q\}$. Then

$$|J| = n - |Q|.$$

Therefore,

$$|Q^*| \leq |J|.$$

We take into account $|Q^*| \leq |J|$ and for each row $r \in Q^*$ we fix one entry and collect these entries in the set W. We require the following:

- for distinct $P_{ab}, P_{cd} \in W$, $b \neq d$;
- for each $P_{ab} \in W$, $b \notin Q^c$.

We apply Theorem 2 on

$$k = |S^i_{\prec}|,$$

where $S^i_{\prec} = S^*_{\prec} \cup \mathsf{Var}(\mathsf{B}_i)$. We denote $W_{\prec} = S^i_{\prec} \cap W$ and $W_{\succ} = S^i_{\succ} \cap W$. For $j = 1, \ldots, k$ we define

$$z_j = \begin{cases} 1, & \text{if } z_j \in X_{\prec} \cup W_{\prec} \\ 0, & \text{otherwise} \end{cases}$$

Choose $\overrightarrow{x}, \overrightarrow{x}' \in \mathbb{B}^k$ such that $\overrightarrow{x} \neq \overrightarrow{x}'$ and $x_j = x'_j = z_j$ for all $z_j \notin A$. Since $x \neq x'$ there is a j' such that $x_{j'} \neq x'_{j'}$. Let $\overrightarrow{y} = (y_{k+1}, \ldots, y_q)$, where $q = |\mathsf{Var}(B_i)|$, be the vector defined for $y_j \in Y$ by

$$y_j = \begin{cases} 1, & \text{if } y_j \text{ is in the same column as } x_{j'} \\ 0, & \text{otherwise} \end{cases}$$

and for $y_j \notin Y$ by

$$y_j = \begin{cases} 1, & \text{if } y_j \in X_{\succ} \cup W_{\succ} \\ 0, & \text{otherwise} \end{cases}$$

Hence, the subset of clauses represented by B_i evaluates to $\neg x_{j'}$ for the assignment $(\overrightarrow{x}, \overrightarrow{y})$ and to $\neg x'_{j'}$ for the assignment $(\overrightarrow{x}', \overrightarrow{y})$. Taking into account that $|A| \geq |Q|/4$, by Theorem 2 we obtain

$$\mathsf{size}(\mathsf{B}_i) \geq 2^{|A|} \geq 2^{|Q|/4}.$$

□

Theorem 4. *For every order $\prec$ on the set of variables, the size of each OBDD refutation of PHP_n is $2^{\Omega(n)}$.*

Proof. Let $n > 20$, and $\mathsf{B}_1, \ldots, \mathsf{B}_l$ be a OBDD refutation of PHP_n. We prove that for an arbitrary total order on variables $\prec$ there is $i \leq l$ such that

$$\mathsf{size}(\mathsf{B}_i) \geq 2^{n(\frac{3}{4}-\frac{1}{4}\sqrt{5})/4} > 1.14^n.$$

Hence, the size of an arbitrary OBDD refutation on PHP_n is $2^{\Omega(n)}$. First we apply Lemma 1 to the matrix representing PC^*_n, where PC^*_n is obtained from PC_n by removing one (arbitrary) clause. Then one of the following holds.

(1) There is a set of $\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5})\rfloor$ rows (we denote this set by R) and there is a set of $2\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5})\rfloor$ entries (we denote this set by S^R) such that the following holds:

- For each $r \in R$ there are $P_{ra}, P_{rb} \in S^R$ such that $P_{ra} \in S_{\prec}$ and $P_{rb} \in S_{\succ}$.
- For distinct $P_{ab}, P_{cd} \in S^R$, $b \neq d$.

(2) There is a set of $\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5})\rfloor$ columns (we denote this set by Q) and there is a set containing $2\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5})\rfloor$ entries (we denote this set by S^Q) such that the following holds:

- For each $q \in Q$ there are $P_{aq}, P_{bq} \in S^Q$ such that $P_{aq} \in S_{\prec}$ and $P_{bq} \in S_{\succ}$.
- For distinct $P_{ab}, P_{cd} \in S^Q$, $a \neq c$.

We obtain by Lemma 5 in the first case

$$\mathsf{size}(\mathsf{B}_i) \geq 2^{|R|/4} = 2^{n(\frac{3}{4}-\frac{1}{4}\sqrt{5})/4},$$

and by Lemma 6 in the second case

$$\mathsf{size}(\mathsf{B}_i) \geq 2^{|Q|/4} = 2^{n(\frac{3}{4}-\frac{1}{4}\sqrt{5})/4}.$$

From this we conclude that an arbitrary OBDD refutation of PHP_n has size exponential in n. □

5.2 Lower bound on OBDD refutations of EPHP_n

In this section we give a formal proof that an arbitrary OBDD refutation of EPHP_n has a lower bound exponential in n.

Theorem 5. *For every order $\prec$ on the set of variables, the size of each OBDD refutation of EPHP_n is $2^{\Omega(n)}$.*

First we need to prove intermediate lemmas.

Lemma 7. *Let F and G be CNFs such that $F \subset \mathsf{PHP}_n$ and $G \subseteq \bigwedge_1^4 \mathsf{EC}^i_n$. Assume that $\mathsf{A} : \mathsf{Var} \to \{\mathsf{true}, \mathsf{false}\}$ is an assignment of variables such that $F \models_A \mathsf{true}$. Then there is an assignment $\mathsf{A}' : \mathsf{Var} \to \{\mathsf{true}, \mathsf{false}\}$ such that for each $P_{ij} \in \mathsf{Var}(F)$, $\mathsf{A}'(P_{ij}) = \mathsf{A}(P_{ij})$ and $F \cup G \models_{A'} \mathsf{true}$.*

Proof. It follows straightforwardly from the construction of $\bigwedge_1^4 \mathsf{EC}_n^i$. □

Lemma 8. *Let $F \subseteq \mathsf{PHP}_n$, $G \subseteq \bigwedge_1^4 \mathsf{EC}_n^i$. Then for any order on variables $\prec$*

$$\mathsf{size}(\mathsf{B}(F \cup G, \prec)) \geq \mathsf{size}(\mathsf{B}(F, \prec)).$$

Proof. Our proof is based on Theorem 1. It is sufficient to show that if $\mathsf{B}(F, \prec)$ has k nodes labeled with a variable P_{ij} then $\mathsf{B}(F \cup G, \prec)$ has at least k nodes labeled with P_{ij}. To prove this we need to show the following.

(1) If there is a node in $\mathsf{B}(F, \prec)$ labeled with a variable P_{ij} then there is a corresponding node in $\mathsf{B}(F \cup G, \prec)$ labeled with P_{ij}.

(2) For two distinct nodes in $\mathsf{B}(F, \prec)$ labeled with a variable P_{ij} there are two distinct nodes in $\mathsf{B}(F \cup G)$ labeled with P_{ij}.

Now we prove the above statements.

(1) Suppose $n_1 \in \mathsf{B}(F, \prec)$ is labeled with a variable P_{ij}. Then the sub-OBDDs rooted at the left child and the right child of the node are not isomorphic and therefore cannot be merged. It follows from Lemma 7 that there is a node $n_2 \in \mathsf{B}(F \cup G, \prec)$ labeled with P_{ij} such that the sub-OBDDs rooted at the left child and the right child of this node are not isomorphic and therefore cannot be merged. Hence, there is a node in $\mathsf{B}(F \cup G, \prec)$ labeled with a variable P_{ij}.

(2) Let $n_1, n_1' \in \mathsf{B}(F, \prec)$ be distinct nodes labeled with a variable P_{ij}. Then the sub-OBDDs rooted either at the left children of the nodes or at the right children of the nodes (or both) are not isomorphic and therefore cannot be merged. Let us assume that the sub-OBDDs that are not isomprphic rooted at the left children of the nodes. It follows from Lemma 7 that there are nodes $n_2, n_2' \in \mathsf{B}(F \cup G, \prec)$ labelled with a variable P_{ij} such that the sub-OBDDs rooted at the left children of these node are not isomorphic and therefore cannot be merged. We conclude that there are distinct nodes $n_2, n_2' \in \mathsf{B}(F \cup G, \prec)$ labeled with a variable P_{ij}.

By Theorem 1, we conclude that $\mathsf{size}(\mathsf{B}(F \cup G, \prec)) \geq \mathsf{size}(\mathsf{B}(F, \prec))$. □

Now we are ready to give a proof of Theorem 5.

Proof of Theorem 5. Let $n > 20$, and $\mathsf{B}_1, \ldots, \mathsf{B}_l$ be an OBDD refutation of EPHP_n. Similar to the proof of Theorem 4 we show that for an arbitrary total order on variables $\prec$ there is an $i < l$ such that

$$\mathsf{size}(\mathsf{B}_i) \geq 2^{n(\frac{3}{4} - \frac{1}{4}\sqrt{5})/4}.$$

We apply Lemma 1 to the matrix representing PC_n^*, and then one of the following holds.

(1) There is a set of $\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5}) \rfloor$ rows (we denote this set by R) and there is a set of $2\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5}) \rfloor$ entries (we denote this set by S^R) such that the following holds:

- For each $r \in R$ there are $P_{ra}, P_{rb} \in S^R$ such that $P_{ra} \in S_\prec$ and $P_{rb} \in S_\succ$.

- For distinct $P_{ab}, P_{cd} \in S^R$, $b \neq d$.

(2) There is a set of $\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5})\rfloor$ columns (we denote this set by Q) and there is a set containing $2\lfloor n(\frac{3}{4} - \frac{1}{4}\sqrt{5})\rfloor$ entries (we denote this set by S^Q) such that the following holds:

- For each $q \in Q$ there are $P_{aq}, P_{bq} \in S^Q$ such that $P_{aq} \in S_{\prec}$ and $P_{bq} \in S_{\succ}$.
- For distinct $P_{ab}, P_{cd} \in S^Q$, $a \neq c$.

For each $i < l$ we denote by B_i^* the OBDD representing $\mathsf{Cls}(\mathsf{B}_i) \cap \mathsf{Cls}(\mathsf{PHP}_n)$ with the same order on variables $\prec$. We conclude by Lemmas 5 and 8 in case (1) that there is an $i < l$ such that

$$\mathsf{size}(\mathsf{B}_i) \geq \mathsf{size}(\mathsf{B}_i^*) \geq 2^{|R|/4} = 2^{n(\frac{3}{4}-\frac{1}{4}\sqrt{5})/4},$$

and by Lemmas 6 and 8 in case (2) that there is an $i < l$ such that

$$\mathsf{size}(\mathsf{B}_i) \geq \mathsf{size}(\mathsf{B}_i^*) \geq 2^{|Q|/4} = 2^{n(\frac{3}{4}-\frac{1}{4}\sqrt{5})/4}.$$

Hence, for an arbitrary OBDD refutation of EPHP_n there is an intermediate OBDD with size exponential in n. □

6. Unrestricted OBDDs do not simulate resolution polynomially

The above observations establish that unrestricted OBDD proof system without existential quantification cannot simulate unrestricted resolution proofs polynomially. In particular, there are contradictory CNFs for which there is a resolution refutation exponentially stronger than any OBDD refutation containing only two rules, Axiom and Join.

Theorem 6. *There is a sequence of contradictory CNFs φ_i, $i > 0$, of size $O(N^{3/4})$ for which there is a resolution refutation of size $O(N)$ and an arbitrary OBDD refutation has size $2^{\Omega(N^{3/4})}$.*

Proof. Let φ_i be EPHP_i and $N = n^{4/3}$. Then the size of φ_i is $O(N^{3/4})$ and by Theorems 3 and 4 there is a resolution refutation of size $O(N)$ and an arbitrary OBDD refutation has size $2^{\Omega(N^{3/4})}$. □

7. Experiments

To give additional, empirical evidence for our theoretical results, we made experiments with a SAT solver and a BDD package. We used MiniSAT 2.0 and the BDD package *buddy 2.4* for our tests[1].

MiniSAT implements a CDCL (conflict driven clause learning) algorithm, which is a modification of the well-known DPLL method. The runs of CDCL solvers directly correspond to resolution proofs. Buddy is a BDD package that provides the usual Boolean operations on BDDs.

[1] MiniSAT is available at http://minisat.se, buddy can be downloaded from http://buddy.sourceforge.net.

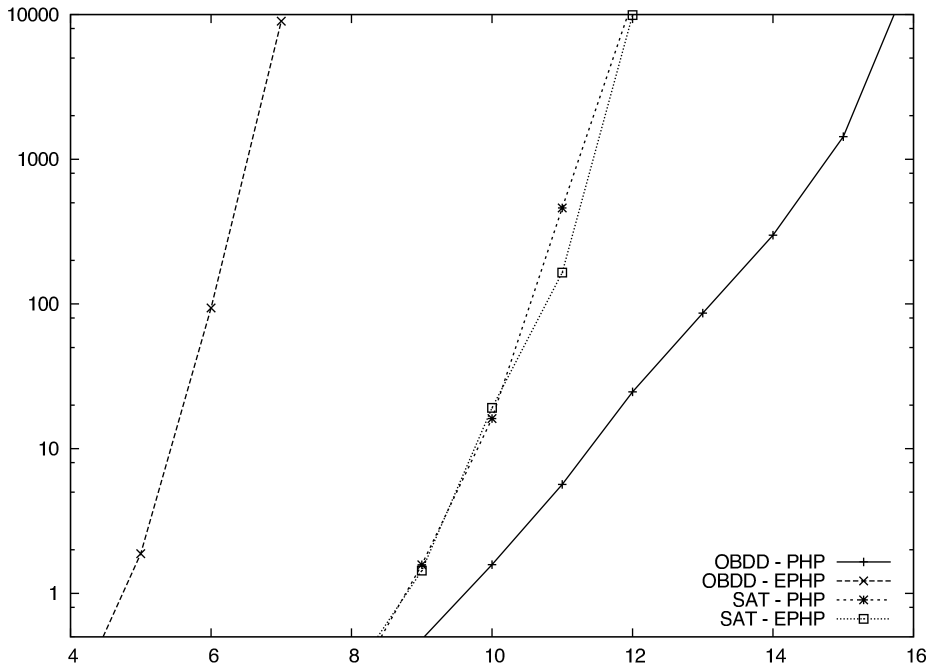

Figure 5. Run-time comparison of a DPLL-based SAT solver (MiniSAT 2.0) and an OBDD package (buddy 2.4) on PHP_n and EPHP_n formulas.

We ran both MiniSAT and buddy on the PHP_n and EPHP_n formulas, using a machine equipped with an Intel Xeon CPU running at 2.66 GHz and 4 GB RAM under Ubuntu Linux. We measured run-times for values of n ranging between 4 and 16, using a run-time limit of ten hours.

The results are shown in Fig. 5, where the run-times (in seconds, on a logarithmic scale) are plotted against parameter n. Regarding BDDs, it turned out that adding the extension clauses (thus switching from PHP to EPHP) lead to dramatically decreased performance[2]. This is in accordance with Lemma 8, which claims that adding extension clauses to a subset of the pigeon hole clauses increases the size of the BDD. Regarding MiniSAT, adding the extension clauses did not increase performance, although a short (polynomial) resolution proof for EPHP_n exists. In principle, a CDCL SAT solver, such as MiniSAT, can polynomially simulate any (general) resolution proof—given the right heuristics for restarts and branching [10]. However, our experiments indicate that the standard heuristic of MiniSAT is not able to find the existing short proof.

[2] We have chosen a fixed variable ordering with $P_{i,j} \prec P_{i',j'}$ iff $(i \prec i') \vee (i = i' \wedge j \prec j')$. Clauses have been added in the following order: first negative clauses, then extension clauses, then positive clauses (as this has shown best performance).

8. Conclusions and future research

One of the results of the paper is a class of CNFs that for infinitely many values of N has a resolution refutation of size $O(N)$, and an arbitrary OBDD Apply refutation of these formulas has size at least $2^{\Omega(N^{3/4})}$. This extends earlier work on comparison of OBDD-based proof systems and resolution-based systems in the following ways.

(1) An exponential separation between a particular OBDD proof system and resolution is presented in [6]. The problem whether there are CNFs of size $O(N)$ that have resolution refutation of size polynomial in N and an arbitrary refutation for a more efficient OBDD Apply proof system, like for example the one in [20], has size at least exponential in N was open in [6]. In comparison with [6], we considered a stronger OBDD proof system that allows clauses to be proceed in an arbitrary order. In this paper we solved the above open problem by presenting a class of formulas that are easy for resolution and hard for an arbitrary OBDD Apply method.

(2) We have improved from $1.025^{\Omega(n)}$ to $1.14^{\Omega(n)}$ lower bound on OBDD refutations of PHP_n presented in [16] .

(2) The main open question in [12] is to improve lower bound on arbitrary OBDD refutations by increasing the constant in the $\Omega()$ of the $2^{\Omega(\sqrt[7]{N/\ln N})}$. This constant is extremely small and it is below 2^{-500}. We considered a family of CNFs that have a higher lower bound on OBDD refutations. But the OBDD proof system we considered is weaker than the one in [12].

(3) A lot of research has been done on exponential lower bounds on the sizes of OBDDs for Boolean functions. But most of the methods to obtain such lower bounds are based on one-way communication complexity and the results from monotone circuits complexity. Clearly, solving structured combinatorial problems in style of Ramsey Theory may lead to new approaches for proving lower bounds.

Still some interesting questions related to comparison of OBDD-based and resolution-based proof systems remain unsolved. It is shown in [6] that biconditional formulas have short OBDD proofs and after transforming them into CNFs they require exponentially long resolution proofs. But OBDD proofs of the transformed formulas need exponential size OBDD proofs too.

For OBDD methods that allow existential quantification we know that there are formulas that have polynomial size OBDD refutations [3], but resolution refutations of only exponential size, i.e. the OBDD proof system with existential quantification is stronger than resolution. An open question is whether the OBDD Apply method can be simulated by resolution polynomially for formulas in CNF.

Another open problem is to give a proof of the tight constant in Lemma 1. The constant c can be improved, and we conjecture that the lemma also holds for $c = 1 - \frac{1}{2}\sqrt{2} \approx 0.293$. Although, it is very easy to give an intuitive explanation why it holds, a precise proof is still needed. Such a proof would result in a better lower bound on OBDD refutations presented in this paper.

References

[1] A. Atserias, P. Kolaitis, and M. Vardi. Constraint propagation as a proof system. In *Principles and Practice of Constraint Programming (CP 2004)*, **3258** of *LNCS*, pages 77–91, 2004.

[2] R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, **8**(C-35):677–691, 1986.

[3] W. Chén and W. Zhang. A direct construction of polynomial-size OBDD proof of pigeon hole problem. *Information Processing Letters*, **109**(10):472–477, 2009.

[4] S. Cook. A short proof of the pigeon hole principle using extended resolution. *ACM SIGACT News*, **8**(4):28–32, 1976.

[5] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, **44**(1):36–50, 1979.

[6] J. F. Groote and H. Zantema. Resolution and binary decision diagrams cannot simulate each other polynomially. *Discrete Applied Mathematics*, **130**:157–171, 2003.

[7] A. Haken. The intractability of resolution. *Theoretical Computer Science*, **39**:297–308, 1985.

[8] J. Krajíček. An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams. *The Journal of Symbolic Logic*, **73**(1):227–237, 2008.

[9] N. Peltier. Extended resolution simulates binary decision diagrams. *Discrete Applied Mathematics*, **156**(6):825–837, 2008.

[10] K. Pipatsrisawat and A. Darwiche. On the power of clause-learning SAT solvers with restarts. In *Principles and Practice of Constraint Programming*, 2009.

[11] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)*, **12**(1):23–41, 1965.

[12] N. Segerlind. Nearly-exponential size lower bounds for symbolic quantifier elimination algorithms and OBDD-based proofs of unsatisfiability. *Electronic Colloquium on Computational Complexity (ECCC)*, **14**(009), 2007.

[13] N. Segerlind. On the relative efficiency of resolution-like proofs and ordered binary decision diagram proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, **14**(126), 2007.

[14] D. Sieling and I. Wegener. NC-algorithms for operations on binary decision diagrams. *Parallel Processing Letters*, **3**:3–12, 1993.

[15] C. Sinz and A. Biere. Extended resolution proofs for conjoining BDDs. In *Computer Science - Theory and Applications, First International Computer Science Symposium in Russia*, **3967** of *LNCS*. Springer, 2006.

[16] O. Tveretina, C. Sinz, and H. Zantema. An exponential lower bound on OBDD refutations for pigeonhole formulas. In *Athens Colloquium on Algorithms and Complexity*, Electronic Proceedings in Theoretical Computer Science, pages 13–21, 2009.

[17] T. E. Uribe and M. E. Stickel. Ordered binary decision diagrams and the davis-putnam procedure. In *First International Conference on Constraints in Computational Logics*, **845**, pages 34–49. Lecture Notes in Computer Science, 1994.

[18] A. Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, pages 425–467, 1995.

[19] I. Wegener. *Branching programs and binary decision diagrams: theory and applications.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[20] H. Zantema and J. C. van de Pol. A rewriting approach to binary decision diagrams. *Journal of Logic and Algebraic Programming*, **49**(1-2):61–86, 2001.