An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams

Jan Krajíček[*][†]

Academy of Sciences and Charles University, Prague

Abstract

We prove an exponential lower bound on the size of proofs in the proof system operating with ordered binary decision diagrams introduced by Atserias, Kolaitis and Vardi [2]. In fact, the lower bound applies to semantic derivations operating with sets defined by OBDDs. We do not assume any particular format of proofs or ordering of variables, the hard formulas are in CNF. We utilize (somewhat indirectly) feasible interpolation.

We define a proof system combining resolution and the OBDD proof system.

Atserias, Kolaitis and Vardi [2] generalized refutation proof systems from Boolean logic to the realm of Constraint Satisfaction Problems (CSP), viewing it as a special case of constraint propagation. This brings constraint propagation within the reach of proof complexity methods and, on the other hand, introduces a new class of propositional proof systems (pps) in the sense of Cook and Reckhow [6].

In the Boolean case Atserias et.al. [2] introduced and studied a particular pps operating with ordered binary decision diagrams (OBDD), and they obtained a number of proof complexity results about it. In particular,

[*]Keywords: proof complexity, OBDD, constraint propagation, feasible interpolation.

[†]Supported in part by grants A1019401, AV0Z10190503, MSM0021620839, 201/05/0124, and LC505.

they compared the pps with several well-known pps', including resolution, constant-depth Frege systems and small-coefficients cutting planes.

A problem left open in Atserias et.al.[2] is to prove a lower bound for this new pps, although they have obtained an interesting partial result: a feasible interpolation theorem (also monotone, and hence a lower bound too) for refutations using OBDDs with certain *specific* orders of variables ([2, Thm.9]). In this paper we prove an exponential lower bound without any restrictions. We deduce first, as a corollary of the feasible interpolation theorem for semantic derivations from Krajíček [11], a feasible interpolation theorem for refutations with OBDDs for special orders of variables. Then we show how to deduce even from such a restricted interpolation a lower bound for OBDD refutations satisfying no apriori restrictions[1].

The paper is organized as follows. The OBDDs and the new proof system of Atserias et.a.[2] are recalled in Section 1. Section 2 recalls the feasible interpolation theorem for semantic derivations of Krajíček [11]. The lower bound is proved in Section 3.

We do not review basics of proof complexity or constraint propagation. The reader my consult [10, 11, 14] for the former and [2, 7] for the latter.

1 The OBDD proof system

Proofs in a lot of usual pps' are organized into proof lines, the lines being syntactic objects representing Boolean functions. For example, in resolution the lines are clauses, in cutting planes these are integer inequalities, and in Frege systems these are arbitrary propositional formulas.

The lines in the proof system of Atserias et.al.[2] are OBDDs. OBDDs are particular branching programs (BP). A BP is a directed acyclic graph with one root which has no incoming edges and exactly two outgoing edges, with two sinks (nodes with no outgoing edges) labeled by 0 and 1 respectively, and with all other nodes (inner nodes) having exactly two outgoing edges and any number of incoming edges. Every node except the sinks is labeled by one of the variables x_i and the two edges leaving the node are labeled $x_i = 0$ and $x_i = 1$, respectively. A truth assignment to variables thus determines

[1] We could have used the interpolation theorem of Atserias et.al.[2]; however, we give our own feasible interpolation theorem because it is a simple corollary of the general interpolation for semantic derivations and because we want that our presentation is complete (the proof of the interpolation theorem from [2] is due to appear in the full paper only).

a unique path from the root to a sink. In this way a BP defines a Boolean function, sending the assignment to the value labeling the sink the path ends in. The size of a BP is the number of its nodes. Every Boolean function can be represented by a BP and, in fact, the minimal size of such a BP is tightly linked with the space complexity of the function, cf. Wegener [21].

An OBDD is a BP in which variables are queried on every path at most once and in an order consistent with one specific linear ordering π of all variables; an OBDD consistent with π is called a π-OBDD. This class of BPs has been introduced by Bryant[4]. The main feature of OBDDs is that every Boolean function can be represented by a unique OBDD in a *reduced* form, and the algorithm transforming any OBDD into its reduced form is p-time. We shall not repeat the reduction and the arguments here (they are straightforward). As a consequence of this fact one can decide in p-time whether two OBDDs represent the same function and, in fact, decide various other relations between functions or perform various manipulations with them, also in p-time. In particular, one can decide in p-time whether a function defined by one OBDD majorizes a function defined by another OBDD (Boolean function f majorizes function g if $f(a) \geq g(a)$ holds for all inputs a). This makes OBDDs very useful data structure representations with applications in verification, model checking, computer-aided design of VLSI circuits, and other areas. See Bryant[5] or Wegener[22].

Atserias et.al.[2, Def.1] define a fairly general concept of *CSP Refutations*. This is a semantic concept, not a proof system in the sense of Cook and Reckhow [6]. It operates with constraints. A constraint is a pair consisting of a tuple of variables (not necessary all of them) and a relation on the set of all possible values of these variables. There are four inference rules: (1) initial constraints, (2) the join of two constraints which is simply the intersection of the two relations extended naturally to all variables occurring in either one of them, (3) the projection of a constraint which is the existential quantification, and (4) the weakening which allows to relax the constraint by enlarging its relation.

Atserias et.al.[2] studied in detail CSP refutations for Boolean logic and with constraints represented by OBDDs. The lower bound we prove is independent of any particular choice of inference rules such as above, as long as they are binary (this is inessential) and sound. Hence rather than defining the OBDD system with the four rules above, we define a "semantic" version of it. We shall call it the *OBDD proof system* (Atserias et.al.[2] do not use any particular name).

For a OBDD P in n variables let f_P denote the Boolean function with domain $\{0,1\}^n$ defined by P. For two Boolean functions f and g, the symbol $f \geq g$ denotes that f majorizes g at every input. Note that a clause over n variables is definable by an OBDD (under any order of variables) of size at most $2+n$.

Definition 1.1 *Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a set of clauses in variables $x = (x_1, \ldots, x_n)$. Let π be a linear ordering of variables x. A* **π-OBDD refutation** *of $\mathcal{C}$ is a sequence*

$$P_1, \ldots, P_k$$

of π-OBDDs such that $f_{P_k} \equiv 0$, and every P_i is either a π-OBDD representing a Boolean function defined by a clause from $\mathcal{C}$, or:

$$f_{P_i} \geq f_{P_{j_1}} \wedge f_{P_{j_2}}$$

for some $j_1, j_2 < i$. An **OBDD refutation** *is a π-OBDD refutation, for some linear ordering π of variables.*

The size of the refutation is the sum of the sizes of P_i', the number of steps (sometimes called also the length) is k.

It is obvious that OBDD refutations are sound and complete. Note that as a consequence of the above remarks on the uniqueness of reduced OBDDs it is decidable in p-time, given π-OBDDs P_1, P_2 and P_3, if $f_{P_3} \geq f_{P_1} \wedge f_{P_2}$. Hence it can be decided in p-time if a sequence of π-OBDDs is a valid OBDD refutation. In other words, the OBDD proof system is a proof system in the sense of Cook and Reckhow [6].

Atserias et.al.[2] showed that OBDD refutations p-simulate resolution but have an exponential speed-up over it, p-simulate cutting planes with small coefficients CP^*, and are incomparable with constant depth Frege systems.

2 Feasible interpolation

Let $A(x,y)$ and$B(x,z)$ be two propositional formulas having the indicated occurrences of variables from tuples $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_s)$ and $z = (z_1, \ldots, z_t)$. Propositional interpolation theorem says that if $A \wedge B$ is unsatisfiable then there is a formula $I(x)$, an interpolant of A and B, such that both $A \wedge I$ and $B \wedge \neg I$ are unsatisfiable.

The idea of feasible interpolation is simple: For a given proof system **P** establish an upper bound on the computational complexity of an interpolant of A and B in terms of the size of a **P**-proof of the unsatisfiability of $A \wedge B$. Then any pair A and B which is hard to interpolate yields formula $A \wedge B$ that must have large **P**-proofs of the unsatisfiability. Unconditional lower bounds are obtained by considering a monotone version of the above idea (see below).

A useful way to look at interpolation (and one that we tacitly use below) is in terms of separating a pair of disjoint sets. Let

$$U := \{x \in \{0,1\}^n \mid \exists y A(x,y)\} \ , \quad V := \{x \in \{0,1\}^n \mid \exists z B(x,z)\} \ ,$$

and

$$W := \{x \in \{0,1\}^n \mid I(x)\} \ .$$

The unsatisfiability of $A \wedge B$ is then equivalent to the disjointness of U and V, and the fact that I is an interpolant of A and B means that W separates U and V:

$$U \cap W = \emptyset \ \text{ and } \ V \subseteq W \ .$$

We also say that a circuit separates two disjoint sets U and V if the set W defined by the circuit does.

Feasible interpolation has been first formulated in Krajíček[9](circulating in 1991), and subsequently (but apparently independently in some cases) used for a variety of proof complexity results (new lower bounds, independence results for bounded arithmetic, establishing links between proof complexity and cryptography, automatizability of proof search, etc.) in Razborov[20], Bonet, Pitassi and Raz[3], Krajíček[11], Krajíček and Pudlák[15], Pudlák[18], and a number of other papers; see [14, 19] for overviews.

In our lower bound proof we apply the general theorem on feasible interpolation for semantic derivations from Krajíček[11] that uses communication complexity[2]. We now recall the necessary definitions and facts from that paper. We assume the reader is familiar with the concept of communication complexity of a function, with inputs distributed among two players: It is the minimal number of bits they need to communicate in the worst case (i.e. the maximum over all inputs of the same length) in order to determine the

[2]The referee required that we mention that Impagliazzo, Pitassi and Urquhart [8] used communication complexity to establish a lower bound for tree-like cutting planes.

value of the function at the given input. See Kushilevitz and Nisan[16] for details.

Let $N = n+s+t$ be natural numbers fixed for the rest of the section. The **semantic rule** allows to infer a subset $C \subseteq \{0,1\}^N$ from two subsets $A, B \subseteq \{0,1\}^N$ iff $C \supseteq A \cap B$. A **semantic refutation** of the sets $A_1, \ldots, A_m \subseteq \{0,1\}^N$ is a sequence of sets $B_1, \ldots, B_k \subseteq \{0,1\}^N$ such that $B_k = \emptyset$, each B_i is either one of A_j or derived from two previous B_{i_1}, B_{i_2} by the semantic rule.

Let $A \subseteq \{0,1\}^N$ and assume $u, v \in \{0,1\}^n$, $y^u \in \{0,1\}^s$ and $z^v \in \{0,1\}^t$. Consider three tasks:

1. Decide whether $(u, y^u, z^v) \in A$.
2. Decide whether $(v, y^u, z^v) \in A$.
3. If $(u, y^u, z^v) \in A \not\equiv (v, y^u, z^v) \in A$ find $i \leq n$ such that $u_i \neq v_i$.

These tasks can be solved by two players, one knowing u, y^u (the U-player) and the other one knowing v, z^v (the V-player). The **communication complexity** of A, $CC(A)$, is the minimal number of bits they need to exchange in the worst case in solving any of these three tasks[3]. That is, $CC(A) = \max(t_1, t_2, t_3)$ where t_i is the communication complexity of task i, $i = 1, 2, 3$.

Consider one more task:

4. If $(u, y^u, z^v) \in A$ and $(v, y^u, z^v) \notin A$ either find $i \leq n$ such that
$$u_i = 1 \wedge v_i = 0$$
or agree (and indicate this by their outputs) that there is some u' satisfying
$$u' \geq u \wedge (u', y^u, z^v) \notin A$$
($u \leq u'$ means $\bigwedge_{i \leq n} u_i \leq u'_i$; the players are not required to find any u'.)

It should be noted that the task 4. has always a solution.

[3]We tacitly assume that the decomposition $N = n + s + t$, and hence the distribution of input bits to the players, is apriori determined.

The **monotone communication complexity w.r.t.** U of A, $MCC_U(A)$, is the minimal $r \geq CC(A)$ such that the task 4. can be solved communicating $\leq r$ bits in the worst case.

For $A \subseteq \{0,1\}^{n+s}$ define the set $\tilde{A}$ by:

$$\tilde{A} := \bigcup_{(a,b)\in A} \{(a,b,c) \mid c \in \{0,1\}^t\}$$

where a, b, c range over $\{0,1\}^n$, $\{0,1\}^s$ and $\{0,1\}^t$ respectively, and similarly for $B \subseteq \{0,1\}^{n+t}$ define $\tilde{B}$:

$$\tilde{B} := \bigcup_{(a,c)\in B} \{(a,b,c) \mid b \in \{0,1\}^s\} .$$

Theorem 2.1 (Krajíček[11]) *Let $A_1, \dots, A_m \subseteq \{0,1\}^{n+s}$ and $B_1, \dots, B_\ell \subseteq \{0,1\}^{n+t}$. Assume that there is a semantic refutation $D_1, \dots, D_k$ of the sets $\tilde{A}_1, \dots, \tilde{A}_m, \tilde{B}_1, \dots, \tilde{B}_\ell$ such that $CC(D_i) \leq r$ for all $i \leq k$.*

Then the two sets

$$U = \{u \in \{0,1\}^n \mid \exists y^u \in \{0,1\}^s; (u, y^u) \in \bigcap_{j\leq m} A_j\}$$

and

$$V = \{v \in \{0,1\}^n \mid \exists z^v \in \{0,1\}^t; (v, z^v) \in \bigcap_{j\leq \ell} B_j\}$$

can be separated by a circuit of size at most $(k+2n)2^{O(r)}$.

Moreover, if the sets $A_1, \dots, A_m$ satisfy the following monotonicity condition w.r.t. U:

$$(u, y^u) \in \bigcap_{j\leq m} A_j \wedge u \leq u' \ \rightarrow \ (u', y^u) \in \bigcap_{j\leq m} A_j$$

and $MCC_U(D_i) \leq r$ for all $i \leq k$, then there is even a monotone circuit separating U from V of size at most $(k+n)2^{O(r)}$.

3 The lower bound

Assume $N = n + s + t$, and let x, y and z be disjoint tuples of n, s and t variables, as in the previous section. Let Var be the set of all these variables in x, y, z. A linear ordering π of Var is **block consistent** with $y < z < x$ iff π puts all y-variables before all z- and x-variables, and all x-variables after all z-variables.

Lemma 3.1 *Let π be a linear ordering of Var that is block consistent with $y < z < x$. Assume a subset $A \subseteq \{0,1\}^N$ is definable by a π-OBDD P of size S. Then both $CC(A)$ and $MCC_U(A)$ are bounded above by $O(\log(S)\cdot\log(n))$.*

Proof :

The U-player, knowing y^u, starts the path through P and sends to the other player $\log(S)$ bits indicating the last node he reached, i.e. the first node not querying a y-variable. Then the V-player, knowing z^v, continues in the path and sends to the U-player $\log(S)$ bits naming the node with first query of an x-variable he got to. Call this node p.

After this stage they both continue from p the computation individually, using u and v respectively, and send each other 1 bit - their output. If the outputs differ then they use binary search looking for a node whose x-query they answered differently. This involves at most $\log(n)$ steps, each time sending each other $\log(S)$ bits naming the node querying the particular x-variable their paths got to. Hence the total estimate to $CC(A)$ is $O(\log(S) \cdot \log(n))$.

For the monotone complexity assume $(u, y^u, z^v) \in A$ while $(v, y^u, z^v) \notin A$. Let $p_0, \ldots, p_\ell$ with $p_0 := p$ be the path from p determined by v (call it the v-path), with p_ℓ being the sink labeled by 0. In particular, $\ell \leq n$.

The U-player indicates by sending 1 bit whether or not there is a $u' \geq u$ such that $(u', y^u, z^v) \notin A$, i.e. the u'-path from p leads to p_ℓ. If so, the players stop, having answered the 4. task.

Otherwise the V-player picks a point in the middle of his path, say $p_{\ell/2}$, and sends its name ($\log(S)$ bits) to the U-player. The U-player sends back 1 bit indicating whether or not there is $u' \geq u$ such that the u'-path from p leads to $p_{\ell/2}$.

If there is such a u' they will move to sub-path $p_{\ell/2}, \ldots, p_\ell$, otherwise to the sub-path $p_0, \ldots, p_{\ell/2}$. Note that the U-player does not know the whole v-path from p, only the endpoints of the current sub-path.

In general, after r rounds of this process, the players have a sub-path of length $\leq n/2^r$ (the U-player knows its endpoints) such that there is a $u' \geq u$ for which the u'-path from p leads to the starting node of the sub-path but no such $u' \geq u$ exists for the end-node of the sub-path. Hence in at most $\log(n)$ rounds, in which they have exchanged at most $\log(n) \cdot (1 + \log(S))$ bits, they find a node p_w on the v-path such that, in particular, for no $u' \geq u$ does the u'-path leads from p_w to p_{w+1}. If x_i is the label of p_w this means (as x_i is not queried at any earlier node) that $v_i = 0$ while $u_i = 1$, and the players have found what they wanted.

q.e.d.

The following lemma is then an immediate consequence of Theorem 2.1 and Lemma 3.1 (we estimate the number of lines by the size). Atserias et.al.[2, Thm.9] give an estimate to the circuit complexity of an interpolant for orderings π block consistent with $y < x < z$.

Lemma 3.2 *Let π be a linear ordering of Var that is block consistent with $y < z < x$. Let $A_1, \ldots, A_m \subseteq \{0,1\}^{n+s}$ and $B_1, \ldots, B_\ell \subseteq \{0,1\}^{n+t}$. Assume that there is a semantic OBDD refutation of the sets $\tilde{A}_1, \ldots, \tilde{A}_m, \tilde{B}_1, \ldots, \tilde{B}_\ell$ of size S.*

Then the two sets

$$U = \{u \in \{0,1\}^n \mid \exists y^u \in \{0,1\}^s; (u, y^u) \in \bigcap_{j \le m} A_j\}$$

and

$$V = \{v \in \{0,1\}^n \mid \exists z^v \in \{0,1\}^t; (v, z^v) \in \bigcap_{j \le \ell} B_j\}$$

can be separated by a circuit of size at most $S^{O(\log(n))}$.

Moreover, if the sets $A_1, \ldots, A_m$ satisfy the monotonicity condition from Theorem 2.1 then there is even a monotone circuit separating U from V of size at most $S^{O(\log(n))}$.

Let $A(x, y)$ and $B(x, z)$ be arbitrary 3CNF formulas (i.e. sets of 3-clauses), where x, y and z are n-, s- and t-tuples of atoms respectively, as above, and $N = n + s + t$. We are going to construct a new CNF formula $D_{A,B}(w, f)$, where w is an N-tuple of variables w_i, and f an N^2-tuple of variables f_{ij} with $i, j \in [N]$.

Let $Map(f)$ be the following CNF formula expressing that f_{ij} defines a graph $\{(i,j) \mid f_{ij} = 1\}$ of a permutation on $[N]$:

$$\bigwedge_i \bigvee_j f_{ij} \ \wedge \ \bigwedge_j \bigvee_i f_{ij} \ \wedge \ \bigwedge_{i_1 \neq i_2, j} (\neg f_{i_1 j} \vee \neg f_{i_2 j}) \ \wedge \ \bigwedge_{i, j_1 \neq j_2} (\neg f_{i j_1} \vee \neg f_{i j_2}) \ .$$

Next define formulas X_j^1 and X_j^0 by:

$$X_j^1 := \bigwedge_{i\in[N]} \neg f_{ij} \vee w_i \quad \text{and} \quad X_j^0 := \bigwedge_{i\in[N]} \neg f_{ij} \vee \neg w_i$$

for $j \in [n]$, and similarly formulas Y_j^1 and Y_j^0:

$$Y_j^1 := \bigwedge_{i\in[N]} \neg f_{i(n+j)} \vee w_i \quad \text{and} \quad Y_j^0 := \bigwedge_{i\in[N]} \neg f_{i(n+j)} \vee \neg w_i$$

for $j \in [s]$, and Z_j^1 and Z_j^0:

$$Z_j^1 := \bigwedge_{i\in[N]} \neg f_{i(n+s+j)} \vee w_i \quad \text{and} \quad Z_j^0 := \bigwedge_{i\in[N]} \neg f_{i(n+s+j)} \vee \neg w_i$$

for $j \in [t]$.

Note that, assuming $Map(f)$, formulas X_j^1 and X_j^0 (and similarly the Ys and Zs) are complementary.

The CNF formula $D_{A,B}(w, f)$ consists of the conjunction of all clauses of $Map(f)$ together with all clauses obtained by the following process: For any 3-clause C from either A or B do the following:

1. Replace each positive literal x_j by formula X_j^1 and each negative literal $\neg x_j$ by X_j^0, and similarly for variables y and z.

2. The resulting formula is a disjunction of 3 conjunctions, each being a conjunction of N 2-clauses: Use distributivity to replace this formula by a conjunctions of N^3 6-clauses.

3. Each 6-clause obtained in this way is a clause of $D_{A,B}$.

Lemma 3.3 *Let σ be an arbitrary linear ordering of variables w and f, and π be an arbitrary linear ordering of variables x, y and z.*

Assume that the formula $D_{A,B}(w, f)$ has a σ-OBDD refutation of size S. Then the formula $A(x, y) \wedge B(x, z)$ has a π-OBDD refutation of size at most S.

Proof :

Assume σ induces on variables w ordering

$$w_{i_1} < w_{i_2} < \ldots < w_{i_N} \ ,$$

$\{i_1, \ldots, i_N\} = [N]$. Let F be a permutation of $[N]$ such that $F(i_r)$ equals

- to j, if the r-th element of π is x_j,
- to $n+j$, if the r-th element of π is y_j,
- to $n+s+j$, if the r-th element of π is z_j,

respectively.

Substitute in the whole refutation $f_{uv} := 1$ if $F(u) = v$, and $f_{uv} := 0$ otherwise. If x_j is the r-th element of the ordering π, after the substitution the formula X_j^1 reduces to w_{i_r} (the r-th variable w in the ordering σ) and X_j^0 reduces to $\neg w_{i_r}$ (and analogously for the variables y and z). Hence the ordering of these reduced formulas X, Y and Z induced by the σ ordering of the variables w is identical to the π ordering of x, y and z.

The substitution for the variables f satisfies $Map(f)$. Hence the original σ-OBDD refutation becomes after the substitution a π-OBDD refutation of $A(x,y) \wedge B(x,z)$.

q.e.d.

To prove the lower bound we only need a pair of 3CNF formulas hard to interpolate, i.e. a pair of NP sets U and V hard to separate. The set U will be closed upwards and will satisfy the monotonicity condition of Theorem 2.1, so that we can use the monotone feasible interpolation. This is essential as lower bounds for monotone circuits are known (Theorem 3.5) but not for general circuits.

Denote the set of two-element subsets of $[m] := \{1, \ldots, m\}$ by the suggestive symbol $\binom{m}{2}$. Truth assignments a to variables x_{ij}, $\{i,j\} \in \binom{m}{2}$, are naturally identified with undirected graphs on $[m]$; such graph will be denoted G_a.

Recall that a clique in a graph is a complete subgraph, and that a graph is ξ-colorable if every vertex can be assigned one of ξ colors such that no two adjacent vertices have the same color.

The following formulas (essentially) have been first discussed by Razborov [20] and used many times since then in connection with feasible interpolation (see the references given above).

Definition 3.4 *Let $m, \omega, \xi \geq 1$. $Clique_{m,\omega}(x,y)$ is any conjunction of 3-clauses in variables x_{ij}, $\{i,j\} \in \binom{m}{2}$, and $m^{O(1)}$ additional variables y such that:*

- *The condition* $\exists y Clique_{m,\omega}(a,y)$ *defines the set* U *of graphs* G_a *having a clique of size at least* ω.

- *Formula* $Clique_{m,\omega}(x,y)$ *satisfies the monotonicity condition of Theorem 2.1:* $a \le a' \wedge Clique_{m,\omega}(a,b) \to Clique_{m,\omega}(a',b)$.

$Color_{m,\xi}(x,z)$ *is any conjunction of 3-clauses in variables* x_{ij}, $\{i,j\} \in \binom{m}{2}$, *and additional* $m^{O(1)}$ *variables* z *such that the condition* $\exists z Color_{m,\xi}(a,z)$ *defines the set* V *of graphs* G_a *that are colorable by* ξ *colors.*

It is obvious that U is closed upwards and that one can find its definition $Clique_{m,\omega}(x,y)$ satisfying the monotonicity condition. A suitable definition is constructed as follows. The additional variables y consists of $\omega \cdot m$ variables $y'_{t,i}$, one for each $t \in [\omega]$ and $i \in [m]$, together with auxiliary variables y'' whose role will be explained below.

Consider the conjunction of the following clauses:

1. $\bigvee_i y'_{t,i}$, one for each $t \in [\omega]$.
2. $\neg y'_{t,i} \vee \neg y'_{s,i}$, one for each $t \neq s \in [\omega]$ and $i \in [m]$.
3. $\neg y'_{t,i} \vee \neg y'_{s,j} \vee x_{ij}$, one for each $t \neq s \in [\omega]$ and $i \neq j \in [m]$.

The clauses in the first and in the second group enforce that $\{(t,i) \mid y'_{t,i} = 1\}$ is the graph of an injective map $F : [\omega] \to [m]$, and the clauses in the last group stipulate that the range of F is a clique in the graph determined by x_{ij}.

The formula $Clique_{m,\omega}(x,y)$ is the conjunction of these clauses after one first uses the auxiliary variables y'' to replace them by 3-clauses. The monotonicity condition is clearly satisfied: If F (represented by y) witnesses that a graph G has a clique of size $[\omega]$ then F continues to witness this fact for any graph H that results from G by adding more edges but not deleting any.

A suitable formula $Color_{m,\xi}(x,z)$ is constructed analogously, with z encoding a ξ-coloring of the graph. Note that the formula

$$\bigwedge Clique_{m,\omega} \ \wedge \ \bigwedge Color_{m,\xi}$$

is obviously unsatisfiable if $\omega > \xi$.

We need to use the following well-known lower bound (we formulate it only for a particular set of parameters).

Theorem 3.5 (Alon-Boppana[1]) *Let $m \geq 3$, and put $\xi := m^{1/2}$ (rounded to the nearest integer) and $\omega := \xi + 1$. Let U and V be the two sets defined from the parameters m, ω, ξ in Definition 3.4.*

Then any monotone circuit separating the sets U and V must have the size at least $2^{\Omega(m^{1/4})}$.

Now we are ready to state and prove the lower bound. Denote by $A_m(x, y)$ the formula $Clique_{m,m^{1/2}+1}(x, y)$ and by $B_m(x, z)$ the formula $Color_{m,m^{1/2}}(x, z)$. Hence $n = m(m + 1)/2$, $s = m^{O(1)}$ and $t = m^{O(1)}$. The total size of formulas A_m and B_m, and also of formula D_{A_m,B_m}, is $m^{O(1)}$.

Theorem 3.6 (main) *Let $m \geq 3$. Then any OBDD refutation of the set of clauses $D_{A_m,B_m}(w, f)$ must have the size at least $2^{\Omega(m^{1/5})}$.*

Proof :

Let π be any linear ordering of variables x, y and z of formulas A_m and B_m that is block consistent with $y < z < x$. Let σ be any ordering of variables w and f. Assume that D_{A_m,B_m} has a size S σ-OBDD refutation.

By Lemma 3.3 then there is a π-OBDD refutation of $A_m \wedge B_m$ of size at most S. By Lemma 3.2 any such refutation yields a monotone circuit separating U from V of size at most $S^{O(\log(n))}$.

By Theorem 3.5 then $S^{O(\log(n))} \geq 2^{\Omega(m^{1/4})}$, i.e. $S \geq 2^{\Omega(m^{1/5})}$.

q.e.d.

One can consider a refutation proof system $R(OBDD)$ combining resolution R with the OBDD proof system. In particular, $R(OBDD)$ operates with clauses $\Gamma = \{P_1, \ldots, P_k\}$ where P_i are π-OBDDs (the same ordering π in the whole refutation) and has just one rule:

$$\frac{\Gamma \cup \{P_1\} \qquad \Delta \cup \{P_2\}}{\Gamma \cup \Delta \cup \{P_3\}}\ , \quad \text{if} \ \ f_{P_3} \geq f_{P_1} \wedge f_{P_2}\ .$$

The objective is to derive clause $\{0\}$ where 0 denotes the reduced OBDD representing the constant 0. Mikle-Barát [17] considered a similar system but with specific syntactic rules modeled upon [2] in place of the semantic rule.

It is easy to see that $R(OBDD)$ p-simulates systems $R(k)$ of [13] and it is thus unlikely that feasible interpolation applies to $R(OBDD)$. Proving lower bounds for the system appears to be an interesting open problem. In particular, its relation to $R(CP)$ of [12] is also unknown.

Acknowledgments: I thank A. Atserias (Barcelona) for discussions about [2] and to E. Jeřábek (Prague), L. Kolodziejczyk (Warsaw/Prague) and N. Thapen (Prague) for critical comments on an earlier draft of the paper. I am also indebted to the anonymous referee for careful reading of the paper.

References

[1] N. Alon, and R. Boppana, The monotone circuit complexity of Boolean functions, *Combinatorica*, **7(1)**, (1987), pp.1-22.

[2] A. Atserias, P. Kolaitis, and M. Vardi, Constraint propagation as a proof system, 10th Int.Conf. on Principles and Practice of Constraint Programing, LN in Computer Science vol.**3258**, Springer, (2004), pp.77-91.

[3] M. L. Bonet, T. Pitassi, and R. Raz, Lower bounds for cutting planes proofs with small coefficients, *J. of Symbolic Logic*,(1997), pp.708-728.

[4] R. E. Bryant, Graph-based algorithms for Boolean function manipulation, *IEEE Transactions on Computing*, C-**35**, (1986), pp.677-691.

[5] R. E. Bryant, Syntactic Boolean manipulation with ordered binary decision diagrams, *ACM Computing Surveys*, **2493)**, (1992), pp.293-318.

[6] S. A. Cook and A. R. Reckhow, The relative efficiency of propositional proof systems, *J. Symbolic Logic*,**44(1)**, (1979), pp.36-50.

[7] R. Dechter, *Constraint processing*, Morgan and Kaufman, (2003).

[8] R. Impagliazzo, T. Pitassi, and A. Urquhart, Upper and lower bounds for tree-like cutting planes proofs, in Proc. *Logic in Computer Science*, (1994), pp.220-228.

[9] J. Krajíček, Lower bounds to the size of constant-depth propositional proofs, *Journal of Symbolic Logic*, **59(1)**, (1994), pp.73-86.

[10] J. Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications, Vol. **60**, Cambridge University Press, (1995).

[11] J. Krajíček, Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic, *J. of Symbolic Logic*, **62(2)**, (1997), pp.457-486.

[12] J. Krajíček, Discretely ordered modules as a first-order extension of the cutting planes proof system, *J. Symbolic Logic*, **63(4)**, (1998), pp.1582-1596.

[13] J. Krajíček, On the weak pigeonhole principle, *Fundamenta Mathematicae*, Vol.**170(1-3)**, (2001), pp.123-140.

[14] J. Krajíček, Propositional proof complexity I., lecture notes available at `http://www.math.cas.cz/~krajicek/ds1.ps`

[15] J. Krajíček, and P. Pudlák, Some consequences of cryptographical conjectures for S^1_2 and EF", in: *Logic and Computational Complexity* (Proc. of the meeting held in Indianapolis, October 1994), Ed. D. Leivant, Springer-Verlag, Lecture Notes in Computer Science, Vol. **960**, (1995), pp.210-220.

Revised version in: *Information and Computation*, Vol. **140 (1)**, (January 10, 1998), pp.82-94.

[16] E. Kushilevitz, and N. Nisan, *Communication complexity*, Cambridge University Press, (1996).

[17] O. Mikle-Barát, *Strong proof systems*, MSc. Thesis, Charles University, (2007). (Available at the ECCC.)

[18] P. Pudlák, Lower bounds for resolution and cutting plane proofs and monotone computations, *J. of Symbolic Logic*,(1997), pp.981-998.

[19] P. Pudlák, The lengths of proofs, in: *Handbook of Proof Theory*, S.R.Buss ed., Elsevier, (1998), pp.547-637.

[20] A. A. Razborov, Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, *Izvestiya of the R.A.N.*, **59(1)**, (1995), pp.201-224.

[21] I. Wegener, *The complexity of Boolean functions*, John Willey and Sons and Teubner Verlag, (1987).

[22] I. Wegener, *Branching programs and binary decision diagrams - theory and applications*, SIAM Monographs in Discrete Mathematics and Its Applications, (2000).

Mailing address:

Mathematical Institute
Academy of Sciences of the Czech Republic
Žitná 25, Prague, 115 67
The Czech Republic
`krajicek@math.cas.cz`