MaxSAT Resolution and Subcube Sums

Yuval Filmus[1], Meena Mahajan[2], Gaurav Sood[2], and Marc Vinyals[1]

[1] Technion – Israel Institute of Technology, Haifa, Israel
yuvalfi@cs.technion.ac.il, marcviny@cs.technion.ac.il
[2] The Institute of Mathematical Sciences (HBNI), Chennai, India
meena@imsc.res.in, gauravs@imsc.res.in

Abstract. We study the MaxRes rule in the context of certifying unsatisfiability. We show that it can be exponentially more powerful than tree-like resolution, and when augmented with weakening (the system MaxResW), p-simulates tree-like resolution. In devising a lower bound technique specific to MaxRes (and not merely inheriting lower bounds from Res), we define a new semialgebraic proof system called the SubCubeSums proof system. This system, which p-simulates MaxResW, is a special case of the Sherali–Adams proof system. In expressivity, it is the integral restriction of conical juntas studied in the contexts of communication complexity and extension complexity. We show that it is not simulated by Res. Using a proof technique qualitatively different from the lower bounds that MaxResW inherits from Res, we show that Tseitin contradictions on expander graphs are hard to refute in SubCubeSums. We also establish a lower bound technique via lifting: for formulas requiring large degree in SubCubeSums, their XOR-ification requires large size in SubCubeSums.

1 Introduction

The most well-studied propositional proof system is Resolution (Res), [3,19]. It is a refutational line-based system that operates on clauses, successively inferring newer clauses until the empty clause is derived, indicating that the initial set of clauses is unsatisfiable. It has just one satisfiability-preserving rule: if clauses $A \vee x$ and $B \vee \neg x$ have been inferred, then the clause $A \vee B$ can be inferred. Sometimes it is convenient, though not necessary in terms of efficiency, to also allow a weakening rule: from clause A, a clause $A \vee x$ can be inferred. While there are several lower bounds known for this system, it is still very useful in practice and underlies many current SAT solvers.

While deciding satisfiability of a propositional formula is NP-complete, the MaxSAT question is an optimization question, and deciding whether its value is as given (i.e. deciding, given a formula and a number k, whether the maximum number of clauses simultaneously satisfiable is exactly k) is potentially harder since it is hard for both NP and coNP. A proof system for MaxSAT was proposed in [5,12]. This system, denoted MaxSAT Resolution or more briefly MaxRes, operates on multi-sets of clauses. At each step, two clauses from the multi-set are resolved and removed. The resolvent, as well as certain "disjoint" weakenings of the two clauses, are added to the multiset. The invariant maintained is that for each assignment ρ, the number of clauses in the multi-set falsified by ρ remains unchanged. The process stops when the multi-set has a satisfiable instance along with k copies of the empty clause; k is exactly the minimum number of clauses of the initial multi-set that must be falsified by every assignment.

Since MaxRes maintains multi-sets of clauses and replaces used clauses, this suggests a "read-once"-like constraint. However, this is not the case; read-once resolution is not even complete [11], whereas MaxRes is a complete system for certifying the MaxSAT value (and in particular, for certifying unsatisfiability). One could use the MaxRes system to certify unsatisfiability, by stopping the derivation as soon as one empty clause is produced. Such a proof of unsatisfiability, by the very definition of the system, can be p-simulated by Resolution. (The MaxRes proof is itself a proof with resolution and weakening, and weakening can be eliminated at no cost.) Thus, lower bounds for Resolution automatically apply to MaxRes and to MaxResW (the augmenting of MaxRes with an appropriate weakening rule) as well. However, since MaxRes needs to maintain a stronger invariant than merely satisfiability, it seems reasonable that for certifying unsatisfiability, MaxRes is weaker than Resolution. (This would explain why, in practice, MaxSAT solvers do not seem to use MaxRes – possibly with the exception of [17], but they instead directly call SAT solvers, which use standard resolution.) Proving this would require a lower bound technique specific to MaxRes.

Associating with each clause the subcube (conjunction of literals) of assignments that falsify it, each MaxRes step manipulates and rearrange multi-sets of subcubes. This naturally leads us to the formulation of a static semi-algebraic proof system that we call the SubCubeSums proof system. This system, by its very definition, p-simulates MaxResW and is a special case of the Sherali–Adams proof system. Given this position in the ecosystem of simple proof systems, understanding its capabilities and limitations seems an interesting question.

Our contributions and techniques

1. We observe that for certifying unsatisfiability, the proof system MaxResW p-simulates the tree-like fragment of Res, TreeRes (Lemma 3). This simulation seems to make essential use of the weakening rule. On the other hand, we show that even MaxRes without weakening is not simulated by TreeRes (Theorem 10). We exhibit a formula, which is a variant of the pebbling contradiction [2] on a pyramid graph, with short refutations in MaxRes (Lemma 4), and show that it requires exponential size in TreeRes (Lemma 9).
2. We initiate a formal study of the newly-defined semialgebraic proof system SubCubeSums, which is a natural restriction of the Sherali–Adams proof system. We show that this system is not simulated by Res (Theorem 11).
3. We show that the Tseitin contradiction on an odd-charged expander graph is hard for SubCubeSums (Theorem 13) and hence also hard for MaxResW. While hardness for MaxResW already follows from the known fact that these formulas are hard for Res, our lower-bound technique is qualitatively different;

it crucially uses the fact that a stricter invariant is maintained in MaxResW and SubCubeSums refutations.

4. Abstracting the ideas from the lower bound for Tseitin contradictions, we devise a lower-bound technique for SubCubeSums based on lifting (Theorem 14). Namely, we show that if every SubCubeSums refutation of a formula F must have at least one wide clause, then every SubCubeSums refutation of the formula $F \circ \oplus$ must have many cubes. We illustrate how the Tseitin contradiction lower bound can be recovered in this way.

The relations among these proof systems is summarized in the figure below.

$$\begin{tikzcd}
 & \text{Sherali–Adams} \arrow[dl] \arrow[dr] & \\
\text{Res} \arrow[rr, dotted] \arrow[dr, dashed] \arrow[ddd] & & \text{SubCubeSums} \arrow[dl] \\
 & \text{MaxResW} \arrow[dr, dashed] \arrow[ddl] & \\
 & & \text{MaxRes} \\
\text{TreeRes} \arrow[uurr, dotted] & &
\end{tikzcd}$$

– $A \longrightarrow B$ denotes that A simulates B and B does not simulate A.
– $A \dashrightarrow B$ denotes that A simulates B.
– $A \cdots\!\!\rightarrow B$ denotes that A does not simulate B.

Related work

One reason why studying MaxRes is interesting is that it displays unexpected power after some preprocessing. As described in [10] (see also [15]), the PHP formulas that are hard for Resolution can be encoded into MaxHornSAT, and then polynomially many MaxRes steps suffice to expose the contradiction. The underlying proof system, DRMaxSAT, has been studied further in [4], where it is shown to p-simulate general Resolution. While DRMaxSAT gains power from the encoding, the basic steps are MaxRes steps. Thus, to understand how DRMaxSAT operates, a better understanding of MaxRes could be quite useful.

A very recent paper in [13] studies a proof system where MaxRes is augmented with an extension rule. The extension rule generalises a weighted version of MaxRes; as defined, it eliminates the non-negativity constraint inherent in MaxResW and SubCubeSums. This system too gains power over MaxRes, while the basic steps remain MaxRes steps and a more generalised weakening.

In the setting of communication complexity and of extension complexity of polytopes, non-negative rank is an important and useful measure. As discussed in [9], the query-complexity analogue is *conical juntas*; these are non-negative combinations of subcubes. Our SubCubeSums refutations are a restriction of conical juntas to non-negative *integral* combinations. Not surprisingly, our lower bound for Tseitin contradictions is similar to the conical junta degree lower bound established in [8].

Organisation of the paper

We define the proof systems MaxRes, MaxResW, SubCubeSums in Section 2. In Section 3 we relate them to TreeRes. In Section 4, we focus on the SubCubeSums proof system, showing the separation from Res (Section 4.1), the lower bound for SubCubeSums (Section 4.2), and the lifting technique (Section 4.3). Some examples / illustrative details appear in an Appendix.

2 Defining the Proof Systems

For set X of variables, let $\langle X \rangle$ denote the set of all total assignments to variables in X. For a (multi-) set of F clauses, $\mathrm{viol}_F : \langle X \rangle \rightarrow \{0\} \cup \mathbb{N}$ is the function mapping α to the number of clauses in F (counted with multiplicity) falsified by α. A (sub)cube is the set of assignments falsifying a clause, or equivalently, the set of assignments satisfying a conjunction of literals.

The proof system Res has the resolution rule inferring $C \vee D$ from $C \vee x$ and $D \vee \overline{x}$, and optionally the weakening rule inferring $C \vee x$ from C if $\overline{x} \notin C$. A refutation of a CNF formula F is a sequence of clauses $C_1, \ldots, C_t$ where each C_i is either in F or is obtained from some $j, k < i$ using resolution or weakening,, and where C_t is the empty clause. The underlying graph of such a refutation has the clauses as nodes, and directed edge from C to D if C is used in the step deriving D. The proof system TreeRes is the fragment of Res where only refutations in which the underlying graph is a tree are permitted. A proof system P simulates (p-simulates) another proof system P' if proofs in P can be transformed into proofs in P' with polynomial blow-up (in time polynomial in the size of the proof). See, for instance, [1], for more details.

The MaxRes and MaxResW proof systems

The MaxRes proof system operates on sets of clauses, and uses the MaxSAT resolution rule [5], defined as follows:

$$\begin{array}{ll}
x \vee a_1 \vee \ldots \vee a_s & (x \vee A) \\
\overline{x} \vee b_1 \vee \ldots \vee b_t & (\overline{x} \vee B) \\
\hline
a_1 \vee \ldots \vee a_s \vee b_1 \vee \ldots \vee b_t & (\text{the ``standard resolvent''}) \\
\\
(\text{weakenings of } x \vee A) & (\text{weakenings of } \overline{x} \vee B) \\
x \vee A \vee \overline{b}_1 & \overline{x} \vee B \vee \overline{a}_1 \\
x \vee A \vee b_1 \vee \overline{b}_2 & \overline{x} \vee B \vee a_1 \vee \overline{a}_2 \\
\vdots & \vdots \\
x \vee A \vee b_1 \vee \ldots \vee b_{t-1} \vee \overline{b}_t & \overline{x} \vee B \vee a_1 \vee \ldots \vee a_{s-1} \vee \overline{a}_t
\end{array}$$

The weakening rule for MaxSAT resolution replaces a clause A by the two clauses $A \vee x$ and $A \vee \overline{x}$. While applying either of these rules, the antecedents are removed from the multi-set and the non-tautologous consequents are added. If F'

is obtained from F by applying these rules, then viol_F and $\text{viol}_{F'}$ are the same function.

In the proof system MaxRes, a refutation of F is a sequence $F = F_0, F_1, \ldots, F_s$ where each F_i is a multi-set of clauses, each F_i is obtained from F_{i-1} by an application of the MaxSAT resolution rule, and F_s contains the empty clause $\Box$. In the proof system MaxResW, F_i may also be obtained from F_i by using the weakening rule. The size of the proof is the number of steps, s. In [5,12], MaxRes is shown to be complete for MaxSAT, hence also for unsatisfiability.

The SubCubeSums proof system

The SubCubeSums proof system is a static proof system. For an unsatisfiable CNF formula F, a SubCubeSums proof is a multi-set G of sub-cubes (or terms, or conjunctions of literals) satisfying $\text{viol}_F \equiv 1 + \text{viol}_G$.

We can view SubCubeSums as a subsystem of the semialgebraic Sherali–Adams proof system as follows. Let F be a CNF formula with m clauses in variables $x_1, \ldots, x_n$. Each clause C_i, $i \in [m]$, is translated into a polynomial equation $f_i = 0$; a Boolean assignment satisfies clause C_i iff it satisfies equation $f_i = 0$. Boolean values are forced through the axioms $x_j^2 - x_j = 0$ for $j \in [n]$. A Sherali–Adams proof is a sequence of polynomials g_i, $i \in [m]$; q_j, $j \in [n]$; and a polynomial p_0 of the form

$$p_0 = \sum_{A,B \subseteq [n]} \alpha_{A,B} \prod_{j \in A} x_j \prod_{j \in B} (1 - x_j) = \sum_{A,B \subseteq [n]} \alpha_{A,B} X_A (1 - X)_B$$

where each $\alpha_{A,B} \geq 0$, such that

$$\left(\sum_{i \in [m]} g_i f_i \right) + \left(\sum_{j \in [n]} q_j (x_j^2 - x_j) \right) + p_0 + 1 = 0$$

The degree or rank of the proof is the maximum degree of $g_i f_i, q_j(x_j^2 - x_j), p_0$.

The polynomials f_i corresponding to the clauses, as well as the terms in p_0, are conjunctions of literals, thus special kinds of d-juntas (Boolean functions depending on at most d variables). So p_0 is a non-negative linear combination of non-negative juntas, that is, in the nomenclature of [9], a *conical junta*.

The Sherali–Adams system is sound and complete, and verifiable in randomized polynomial time; see for instance [7].

Consider the following restriction of Sherali–Adams:

1. Each $g_i = -1$.
2. Each $\alpha_{A,B} \in \mathbb{Z}^{\geq 0}$, (non-negative integers), and $\alpha_{A,B} > 0 \implies A \cap B = \emptyset$.

This implies each q_j must be 0, since the rest of the expression is multilinear. Hence, for some non-negative integral $\alpha_{A,B}$,

$$\sum_{A,B \subseteq [n] : A \cap B = \emptyset} \alpha_{A,B} \prod_{j \in A} x_j \prod_{j \in B} (1 - x_j) + 1 = \left(\sum_{i \in [m]} f_i \right)$$

This is exactly the SubCubeSums proof system: the terms in p_0 are subcubes, and the right-hand-side is viol_F. For each disjoint pair A, B, the SubCubeSums proof has $\alpha_{A,B}$ copies of the corresponding sub-cube. The degree of a SubCubeSums proof is the maximum number of literals appearing in a conjunction. The size of a SubCubeSums proof is the number of subcubes, that is, $\sum_{A,B} \alpha_{A,B}$. The constraint $g_i = 1$ means that for bounded CNF formulas, the degree of a SubCubeSums proof is essentially the degree of p_0, i.e. the degree of the juntas.

The following proposition shows why this restriction remains complete.

Proposition 1. *SubCubeSums p-simulates MaxResW.*

Proof. If an unsatisfiable CNF formula F with m clauses and $n \geq 3$ variables has a MaxResW refutation with s steps, then this derivation produces $\{\Box\} \cup G$ where the number of clauses in G is at most $m + (n-2)s - 1$. (A weakening step increases the number of clauses by 1. A MaxRes step increases it by at most $n-2$.) The subcubes falsifying the clauses in G give a SubCubeSums proof. $\Box$

In fact, SubCubeSums is also implicationally complete in the following sense. We say that $f \geq g$ if for every Boolean x, $f(x) \geq g(x)$.

Proposition 2. *If f and g are polynomials with $f \geq g$, then there are subcubes h_j and non-negative numbers c_j such that on the Boolean hypercube, $f - g = \sum_j c_j h_j$. Further, if f, g are integral on the Boolean hypercube, so are the c_j.*

Proof. A brute-force way to see this is to consider subcubes of degree n, i.e. a single point/assignment. For each $\beta \in \{0,1\}^n$, define $c_\beta = (f-g)(\beta) \in \mathbb{R}^{\geq 0}$. $\Box$

3 MaxRes, MaxResW, and TreeRes

Since TreeRes allows reuse only of input clauses, while MaxRes does not allow any reuse of clauses but produces multiple clauses at each step, the relative power of these fragments of Res is intriguing. In this section, we show that MaxRes with the weakening rule, MaxResW, p-simulates TreeRes, is exponentially separated from it, and even MaxRes (without weakening) is not simulated by TreeRes.

Lemma 3. *For every unsatisfiable CNF F,*
$\mathit{size}(F \vdash_{\mathit{MaxResW}} \Box) \leq 2\mathit{size}(F \vdash_{\mathit{TreeRes}} \Box)$.

Proof. Let T be a tree-like derivation of $\Box$ from F of size s. Without loss of generality, we may assume that T is regular; no variable is used as pivot twice on the same path.

Since a MaxSAT resolution step always adds the standard resolvent, each step in a tree-like resolution proof can be performed in MaxResW as well, provided the antecedents are available. However, a tree-like proof may use an axiom (a clause in F) multiple times, whereas after it is used once in MaxResW it is no longer available. So we need to work with weaker antecedents.

For each axiom $A \in F$, consider the subtree T_A of T defined by retaining only the paths from leaves labeled A to the final empty clause. Start with A at the final node and walk up the tree T_A towards the leaves. If we reach a branching node v with clause A', and the pivot at v is x, weaken A' to $A' \vee x$ and $A' \vee \overline{x}$. Proceed along the edge contributing x with $A' \vee x$, and along the other edge with $A' \vee \overline{x}$. Since T is regular, no tautologies are created in this process.

After doing this for each axiom, we have as many clauses as leaves in T. Now we simply perform all the steps in T.

Since each weakening step increases the number of clauses by one, and since we finally produce at most s clauses for the leaves, the number of weakening steps required is at most s. □

(See the appendix for a simple illustration.)

We now show that even without weakening, MaxRes has short proofs of formulas exponentially hard for TreeRes. The formulas that exhibit the separation are *composed* formulas of the form $F \circ g$, where F is a CNF formula, $g\colon \{0,1\}^\ell \to \{0,1\}$ is a Boolean function, there are b new variables $x_1, \dots, x_\ell$ for each original variable x of F, and there is a block of clauses $C \circ g$, a CNF expansion of the expression $\bigvee_{x^b \in C} [\![g(x_1, \dots x_\ell) = b]\!]$, for each original clause $C \in F$.

We denote by $\mathrm{PebHint}(G)$ the standard pebbling formula with additional hints $u \vee v$ for each pair of siblings (u, v)—that is, two incomparable vertices with a common predecessor—, and we prove the separation for the composed formula $\mathrm{PebHint}(G) \circ \mathrm{OR}$. More formally, if G is a DAG with a single sink z, we define $\mathrm{PebHint}(G) \circ \mathrm{OR}$ as follows. For each vertex $v \in G$ there are variables v_1 and v_2. The clauses are

- For each source v, the clause $v_1 \vee v_2$.
- For each internal vertex w with predecessors u, v, the expression $((u_1 \vee u_2) \wedge (v_1 \vee v_2)) \to (w_1 \vee w_2)$, expanded into 4 clauses.
- The clauses $\overline{z_1}$ and $\overline{z_2}$ for the sink z.
- For each pair of siblings (u, v), the clause $u_1 \vee u_2 \vee v_1 \vee v_2$.

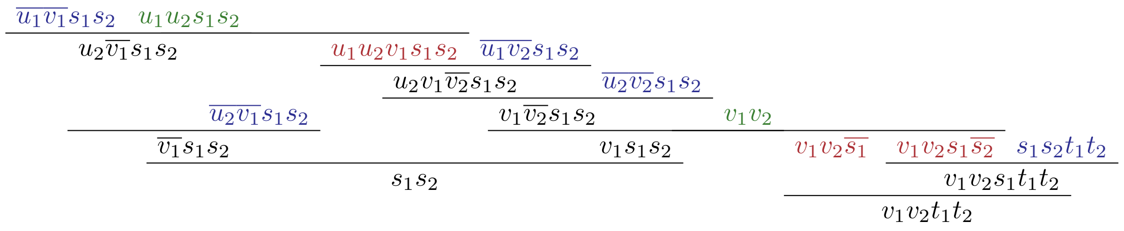
Note that the first three types of clauses are also present in standard composed pebbling formulas, while the last type are the hints.

We prove a MaxRes upper bound for the particular case of pyramid graphs. Let P_h be a pyramid graph of height h and $n = \Theta(h^2)$ vertices.

Lemma 4. *The* $\mathrm{PebHint}(P_h) \circ \mathrm{OR}$ *formulas have $\Theta(n)$ size MaxRes refutations.*

Proof. We derive the clause $s_1 \vee s_2$ for each vertex $s \in P_n$ in layered order, and left-to-right within one layer. If s is a source, then $s_1 \vee s_2$ is readily available as an axiom. Otherwise assume that for a vertex s with predecessors u and v and siblings r and t – in this order – we have clauses $u_1 \vee u_2 \vee s_1 \vee s_2$ and $v_1 \vee v_2$, and let us see how to derive $s_1 \vee s_2$. We also make sure that the clause $v_1 \vee v_2 \vee t_1 \vee t_2$ becomes available to be used in the next step.

In the following derivation we skip $\vee$ symbols, and we colour-code clauses so that green clauses are available by induction, axioms are blue, and red clauses, on the right side in steps with multiple consequents, are additional clauses that are obtained by the MaxRes rule but not with the usual resolution rule.


$\overline{u_1 v_1} s_1 s_2$
$u_1 u_2 s_1 s_2$
$u_2 \overline{v_1} s_1 s_2$
$u_1 u_2 v_1 s_1 s_2$
$\overline{u_1 v_2} s_1 s_2$
$u_2 v_1 \overline{v_2} s_1 s_2$
$\overline{u_2 v_2} s_1 s_2$
$\overline{u_2 v_1} s_1 s_2$
$v_1 \overline{v_2} s_1 s_2$
$v_1 v_2$
$\overline{v_1} s_1 s_2$
$v_1 s_1 s_2$
$v_1 v_2 \overline{s_1}$
$v_1 v_2 s_1 \overline{s_2}$
$s_1 s_2 t_1 t_2$
$s_1 s_2$
$v_1 v_2 s_1 t_1 t_2$
$v_1 v_2 t_1 t_2$

The case where some of the siblings are missing is similar: if r is missing then we use the axiom $u_1 \vee u_2$ instead of the clause $u_1 \vee u_2 \vee s_1 \vee s_2$ that would be available by induction, and if t is missing then we skip the steps that use $s_1 \vee s_2 \vee t_1 \vee t_2$ and lead to deriving $v_1 \vee v_2 \vee t_1 \vee t_2$.

Finally, once we derive the clause $z_1 \vee z_2$ for the sink, we resolve it with axiom clauses $\overline{z_1}$ and $\overline{z_2}$ to obtain a contradiction.

A constant number of steps suffice for each vertex, for a total of $\Theta(n)$. $\square$

We can prove a tree-like lower bound along the lines of [1], with some extra care to respect the hints. For that we use the *pebble game*, a game where the single player starts with a DAG and a set of pebbles, the allowed moves are to place a pebble on a vertex if all its predecessors have pebbles or to remove a pebble at any time, and the goal is to place a pebble on the sink using the minimum number of pebbles. Denote by $\mathrm{bpeb}(P \to w)$ the cost of placing a pebble on a vertex w assuming there are free pebbles on a set of vertices $P \subseteq V$ – in other words, the number of pebbles used outside of P when the starting position has pebbles in P. For a DAG G with a single sink z, $\mathrm{bpeb}(G)$ denotes $\mathrm{bpeb}(\emptyset \to z)$. For $U \subseteq V$ and $v \in V$, the subgraph of v modulo U is the set of vertices u such that there exists a path from u to v avoiding U.

Lemma 5 ([6]). $\mathrm{bpeb}(P_h) = h + 1$.

Lemma 6 ([1]). *For all P, v, w, we have*
$\mathrm{bpeb}(P \to v) \leq \max(\mathrm{bpeb}(P \to w), \mathrm{bpeb}(P \cup \{w\} \to v) + 1)$.

The canonical search problem of a formula F is the relation $\mathrm{Search}(F)$ where inputs are variable assignments $\alpha \in \{0,1\}^n$ and the valid outputs for α are the clauses $C \in F$ that α falsifies. Given a relation f, we denote by $\mathrm{DT}_1(f)$ the 1-query complexity of f [14], that is the minimum over all decision trees computing f of the maximum of 1-answers that the decision tree receives.

Lemma 7. *For all G we have* $\mathrm{DT}_1(\mathrm{Search}(\mathrm{PebHint}(G))) \geq \mathrm{bpeb}(G) - 1$.

Proof. We give an adversarial strategy. Let R_i be the set of variables that are assigned to 1 at round i. We initially set $w_0 = z$, and maintain the invariant that

1. there is a distinguished variable w_i and a path π_i from w_i to the sink z such that a queried variable v is 0 iff $v \in \pi_i$; and

2. after each query the number of 1 answers so far is at least $\mathrm{bpeb}(G) - \mathrm{bpeb}(R_i \to w_i)$.

Assume that a variable v is queried. If v is not in the subgraph of w_i modulo R_i then we answer 0 if $v \in \pi_i$ and 1 otherwise. Otherwise we consider $p_0 = \mathrm{bpeb}(R_i \to v)$ and $p_1 = \mathrm{bpeb}(R_i \cup \{v\} \to w_i)$. By Lemma 6, $\mathrm{bpeb}(R_i \to w_i) \leq \max(p_0, p_1 + 1)$. If $p_0 \geq p_1$ then we answer 0, set $w_{i+1} = v$, and extend π_i with a path from w_{i+1} to w_i that does not contain any 1 variables (which exists by definition of subgraph modulo R_i). This preserves item 1 of the invariant, and since $p_0 \geq \mathrm{bpeb}(R_i \to w_i)$, item 2 is also preserved. Otherwise we answer 1 and since $p_1 \geq \mathrm{bpeb}(R_i \to w_i) - 1$ the invariant is also preserved.

This strategy does not falsify any hint clause, because all 0 variables lie on a path, or the sink axiom, because the sink is assigned 0 if at all. Therefore the decision tree ends at a vertex w_t that is set to 0 and all its predecessors are set to 1, hence $\mathrm{bpeb}(R_t \to w_t) = 1$. By item 2 of the invariant the number of 1 answers is at least $\mathrm{bpeb}(G) - 1$. □

To complete the lower bound we use the Pudlák–Impagliazzo Prover–Delayer game [18] where Prover points to a variable, Delayer may answer 0, 1, or $*$, in which case Delayer obtains a point in exchange for letting Prover choose the answer, and the game ends when a clause is falsified.

Lemma 8 ([18]). *If Delayer can win p points, then all TreeRes proofs require size at least 2^p.*

Lemma 9. *$F \circ \mathrm{OR}$ requires size $\exp(\Omega(\mathrm{DT}_1(\mathrm{Search}(F))))$ in tree-like resolution.*

Proof. We use a strategy for the 1-query game of $\mathrm{Search}(F)$ to ensure that Delayer gets $\mathrm{DT}_1(F)$ points in the Prover–Delayer game. If Prover queries a variable x_i then

- If x is already queried we answer accordingly.
- Otherwise we query x. If the answer is 0 we answer 0, otherwise we answer $*$.

Our strategy ensures that if both x_1 and x_2 are assigned then $x_1 \vee x_2 = x$. Therefore the game only finishes at a leaf of the decision tree, at which point Delayer earns as many points as 1s are present in the path leading to the leaf. The lemma follows by Lemma 8. □

The formulas $\mathrm{PebHint}(P_n) \circ \mathrm{OR}$ are easy to refute in MaxRes (Lemma 4), but from Lemmas 5,7, and 9, they are exponentially hard for TreeRes. Hence,

Theorem 10. *TreeRes does not simulate MaxResW and MaxRes.*

4 The SubCubeSums Proof System

4.1 Res does not simulate SubCubeSums

We now show that Res does not simulate SubCubeSums.

Theorem 11. *There are formulas that have SubCubeSums proofs of size* $\mathrm{O}(n)$ *but require resolution length* $\exp(\Omega(n))$.

The separation is achieved using subset cardinality formulas [20,22,16]. These are defined as follows: we have a bipartite graph $G(U \cup V, E)$, with $|U| = |V| = n$. The degree of G is 4, except for two vertices that have degree 5. There is one variable for each edge. For each left vertex $u \in U$ we have a constraint $\sum_{e \ni u} x_e \geq \lceil d(u)/2 \rceil$, while for each right vertex $v \in V$ we have a constraint $\sum_{e \ni v} x_e \leq \lfloor d(v)/2 \rfloor$, both expressed as a CNF. In other words, for each vertex $u \in U$ we have the clauses $\bigvee_{i \in I} x_i$ for $I \in \binom{E(u)}{\lfloor d(u)/2 \rfloor + 1}$, while for each vertex $v \in V$ we have the clauses $\bigvee_{i \in I} \overline{x_i}$ for $I \in \binom{E(v)}{\lfloor d(v)/2 \rfloor + 1}$.

The lower bound requires G to be an expander, and is proven in [16, Theorem 6]. The upper bound is the following lemma.

Lemma 12. *Subset cardinality formulas have SubCubeSums proofs of size* $\mathrm{O}(n)$.

Proof. Our plan is to reconstruct each constraint independently, so that for each vertex we obtain the original constraints $\sum_{e \ni u} x_e \geq \lceil d(u)/2 \rceil$ and $\sum_{e \ni v} \overline{x_e} \geq \lceil d(v)/2 \rceil$, and then add all of these constraints together.

Formally, if F_u is the set of polynomials that encode the constraint corresponding to vertex u, we want to write

$$\sum_{f \in F_u} f - \left(\lceil d(u)/2 \rceil - \sum_{e \ni u} x_e \right) = \sum_j c_{u,j} h_j \tag{1}$$

and

$$\sum_{f \in F_v} f - \left(\lceil d(v)/2 \rceil - \sum_{e \ni v} \overline{x_e} \right) = \sum_j c_{v,j} h_j \tag{2}$$

with $c_{u,j}, c_{v,j} \geq 0$ and $\sum_j c_{u,j} = \mathrm{O}(1)$, so that

$$\begin{aligned}
\sum_{f \in F} f &= \sum_{u \in U} \sum_{f \in F_u} f + \sum_{v \in V} \sum_{f \in F_v} f \\
&= \sum_{u \in U} \left(\lceil d(u)/2 \rceil - \sum_{e \ni u} x_e + \sum_j c_{u,j} h_j \right) + \sum_{v \in V} \left(\lceil d(v)/2 \rceil - \sum_{e \ni v} \overline{x_e} + \sum_j c_{v,j} h_j \right) \\
&= \sum_{u \in U} \lceil d(u)/2 \rceil + \sum_{v \in V} \lceil d(v)/2 \rceil - \sum_{e \in E} (x_e + \overline{x_e}) + \sum_j c_j h_j \\
&= \left(1 + \sum_{u \in U} 2 \right) + \left(1 + \sum_{u \in U} 2 \right) - \sum_{e \in E} 1 + \sum_j c_j h_j \\
&= (2n+1) + (2n+1) - (4n+1) + \sum_j c_j h_j = 1 + \sum_j c_j h_j
\end{aligned}$$

where $c_j = \sum_{v \in U \cup V} c_{v,j} \geq 0$. Hence we can write $\sum_{f \in F} f - 1 = \sum_j c_j h_j$ with $\sum_j c_j = \mathrm{O}(n)$.

It remains to show how to derive equations (1) and (2). The easiest way is to appeal to the implicational completeness of SubCubeSums, Proposition 2. We continue deriving equation (1), assuming for simplicity a vertex of degree d and incident edges $[d]$. Let $\overline{x_I} = \prod_{i \in I} \overline{x_i}$, and let $\left\{\overline{x_I} : I \in \binom{[d]}{d-k+1}\right\}$ represent a constraint $\sum_{i \in [d]} x_i \geq k$. Let $f = \sum_{I \in \binom{[d]}{d-k+1}} \overline{x_I}$ and $g = k - \sum_{i \in [d]} x_i$. For each point $x \in \{0,1\}^d$ we have that either x satisfies the constraint, in which case $f(x) \geq 0 \geq g(x)$, or it falsifies it, in which case we have on the one hand $g(x) = s > 0$, and on the other hand $f(x) = \binom{d-k+s}{d-k+1} = \frac{(d-k+s)\cdot\dots\cdot s}{(d-k+1)\cdot\dots\cdot 1} \geq s$.

We proved that $f \geq g$, therefore by Proposition 2 we can write $f - g$ as a sum of subcubes of size at most $2^d = \mathrm{O}(1)$.

Equation (2) can be derived analogously, completing the proof.

(See the appendix for an explicit derivation.) □

4.2 A lower bound for SubCubeSums

Fix any graph G with n nodes and m edges, and let I be the node-edge incidence matrix. Assign a variable x_e for each edge e. Let b be a vector in $\{0,1\}^n$ with $\sum_i b_i \equiv 1 \bmod 2$. The Tseitin contradiction asserts that the system $IX = b$ has a solution over $\mathbb{F}_2$. The CNF formulation has, for each vertex u in G, with degree d_u, a set S_u of $2^{d_u - 1}$ clauses expressing that the parity of the set of variables $\{x_e \mid e \text{ is incident on } u\}$ equals b_u.

These formulas are exponentially hard for Res [21], and hence are also hard for MaxResW. We now show that they are also hard for SubCubeSums. By Theorem 11, this lower bound cannot be inferred from hardness for Res.

We will use some standard facts: For connected graph G, over $\mathbb{F}_2$, if $\sum_i b_i \equiv 1 \bmod 2$, then the equations $IX = b$ have no solution, and if $\sum_i b_i \equiv 0 \bmod 2$, then $IX = b$ has exactly 2^{m-n+1} solutions. Furthermore, for any assignment a, and any vertex u, a falsifies at most one clause in S_u.

A graph is a c-expander if for all $V' \subseteq V$ with $|V'| \leq |V|/2$, $|\delta(V')| \geq c|V'|$, where $\delta(V') = \{(u,v) \in E \mid u \in V', v \in V \setminus V'\}$.

Theorem 13. *Tseitin contradictions on odd-charged expanders require exponential size SubCubeSums refutations.*

Proof. Fix a graph G that is a d-regular c-expander on n vertices, where n is odd; $m = dn/2$. Let b be the all-1s vector. The Tseitin contradiction F has $n2^{d-1}$ clauses. By the facts mentioned above, for all $a \in \{0,1\}^m$, $\mathrm{viol}_F(a)$ is odd. So viol_F partitions $\{0,1\}^m$ into $X_1, X_3, \dots, X_{N-1}$, where $X_i = \mathrm{viol}_F^{-1}(i)$.

Let $\mathcal{C}$ be a SubCubeSums refutation of F, that is, $\mathrm{viol}_{\mathcal{C}} = \mathrm{viol}_F - 1 = g$, say. For a cube C, define $N_i(C) = |C \cap X_i|$. Then for all $C \in \mathcal{C}$, $N_1(C) = 0$, and so C is partitioned by X_i, $i \geq 3$. Let $\mathcal{C}'$ be those cubes of $\mathcal{C}$ that have a non-empty part in X_3. We will show that $\mathcal{C}'$ is large. In fact, we will show that for a suitable S, the set $\mathcal{C}'' \subseteq \mathcal{C}'$ of cubes with $|C \cap X_5| \leq S|C \cap X_3|$ is large.

Defining the probability distribution μ on $\mathcal{C}'$ as

$$\mu(C) = \frac{|C \cap X_3|}{\sum_{D \in \mathcal{C}'} |D \cap X_3|} = \frac{N_3(C)}{\sum_{D \in \mathcal{C}'} N_3(D)} \quad ,$$

$$|\mathcal{C}| \geq |\mathcal{C}'| = \sum_{C \in \mathcal{C}'} 1 = \mathop{\mathbb{E}}_{C\sim\mu}\left[\frac{1}{\mu(C)}\right] \geq \underbrace{\mathop{\mathbb{E}}_{C\sim\mu}\left[\frac{1}{\mu(C)}\middle|\frac{|C\cap X_5|}{|C\cap X_3|} \leq S\right]}_{A} \cdot \underbrace{\Pr_{\mu}\left[\frac{|C\cap X_5|}{|C\cap X_3|} \leq S\right]}_{B} \tag{3}$$

We want to choose a good value for S so that A is very large and B is sufficiently large, $\Theta(1)$. To see what will be a good value for S, we estimate the expected value of $\frac{|C\cap X_5|}{|C\cap X_3|}$ and then use Markov's inequality. For this, we should understand the sets X_3, X_5 better. These set sizes are known precisely: for each odd i, $|X_i| = \binom{n}{i}2^{m-n+1}$. (An assignment lies in i cubes of f, each cube corresponds to a distinct vertex because the 2^{d-1} cubes corresponding to a single vertex are disjoint, once the i vertices are fixed and b flipped in those coordinates to get b', there are 2^{m-n+1} 0-1 solutions to $Ix = b'$.)

Now let us consider $C \cap X_3$ and $C \cap X_5$ for a $C \in \mathcal{C}'$ (that is, we know $C \cap X_3 \neq \emptyset$ and $C \cap X_1 = \emptyset$). We rewrite the system $IX = b$ as $I'X' + I_C X_C = b$, where X_C are the variables fixed in cube C (to a_C, say). So $I'X' = b + I_C X_C$. An assignment a is in $C \cap X_r$ iff it is of the form $a'a_C$, and a' falsifies exactly r equations in $I'X' = b'$ where $b' = b + I_C a_C$. This is a system for the subgraph G_C where the edges in X_C have been deleted. This subgraph may not be connected, so we cannot use our size expressions directly. Consider the vertex sets $V_1, V_2, \ldots$ of the components of G_C. The system $I'X' = b'$ can be broken up into independent systems; $I'(i)X'(i) = b'(i)$ for the ith connected component. Say a component is odd if $\sum_{j\in V_i} b'(i)_j \equiv 1 \bmod 2$, even otherwise. Let $|V_i| = n_i$ and $|E_i| = m_i$. Any a' falsifies an odd/even number of equations in an odd/even component.

For $a' \in C \cap X_3$, it must falsify three equations overall, so G_C must have either one or three odd components. If it has only one odd component, then there is another assignment in C falsifying just one equation (from this odd component), so $C \cap X_1 \neq \emptyset$, a contradiction. Hence G_C has exactly three odd components, with vertex sets V_1, V_2, V_3, and overall $k \geq 3$ components. An $a \in C \cap X_3$ falsifies exactly one equation in $I(1), I(2), I(3)$. We thus arrive at the expression

$$|C \cap X_3| = \left(\prod_{i=1}^{3} n_i 2^{m_i - n_i + 1}\right)\left(\prod_{i\geq 4} 2^{m_i - n_i + 1}\right) = n_1 n_2 n_3 2^{m - w(C) - n + k}.$$

An $a' \in C \cap X_5$ must falsify five equations overall. One each must be from V_1, V_2, V_3. The remaining 2 must be from the same component. Hence

$$\begin{aligned}|C \cap X_5| &= \left(\binom{n_1}{3}n_2 n_3 + n_1\binom{n_2}{3}n_3 + n_1 n_2\binom{n_3}{3}\right)2^{m-w(C)-n+k} \\ &\quad + n_1 n_2 n_3 \sum_{i=4}^{k}\binom{n_i}{2}2^{m-w(C)-n+k} \\ &\geq n_1 n_2 n_3 2^{m-w(C)-n+k}\left(\frac{1}{3}\sum_{i=1}^{k}\binom{n_i - 1}{2}\right)\end{aligned}$$

Hence we have, for $C \in \mathcal{C}'$,

$$\frac{|C \cap X_5|}{|C \cap X_3|} \geq \frac{1}{3} \sum_{i=1}^{k} \binom{n_i - 1}{2}$$

This alone does not tell us enough unless we can say something about the n_i's. But we can deduce more by using the definition of μ, and the following fact: Since $g = \mathrm{viol}_F - 1$, an assignment in X_3 belongs to exactly two cubes in $\mathcal{C}$, and by definition these cubes are in $\mathcal{C}'$. Similarly, an assignment in X_5 belongs to exactly four cubes in $\mathcal{C}$, not all of which may be in $\mathcal{C}'$. Hence

$$\sum_{C \in \mathcal{C}'} |C \cap X_3| = 2|X_3| = 2\binom{n}{3} 2^{m-n+1}$$

$$\sum_{C \in \mathcal{C}'} |C \cap X_5| \leq 4|X_5| = 4\binom{n}{5} 2^{m-n+1}$$

$$\mu(C) = \frac{|C \cap X_3|}{2|X_3|}$$

Now we can estimate the average:

$$\mathop{\mathbb{E}}_{\mu}\left[\frac{|C \cap X_5|}{|C \cap X_3|}\right] = \sum_{C \in \mathcal{C}'} \mu(C) \frac{|C \cap X_5|}{|C \cap X_3|} = \sum_{C \in \mathcal{C}'} \frac{|C \cap X_5|}{2|X_3|} \leq \frac{4|X_5|}{2|X_3|} \leq \frac{n^2}{10}$$

Choosing $S = n^2/9$, and using Markov's inequality, we get

$$B = \Pr_{\mu}\left[\frac{|C \cap X_5|}{|C \cap X_3|} \leq S = \frac{n^2}{9}\right] \geq 1/10$$

Now we show that conditioned on $\frac{|C \cap X_5|}{|C \cap X_3|} \leq S$, the average value of $\frac{1}{\mu(C)}$ is large.

$$\frac{1}{\mu(C)} = \frac{2|X_3|}{|C \cap X_3|} = \frac{2\binom{n}{3}2^{m-n+1}}{n_1 n_2 n_3 2^{m-w(C)-n+k}} = \frac{2\binom{n}{3}2^{w(C)+1-k}}{n_1 n_2 n_3} \geq \frac{2^{w(C)+1-n}}{3}$$

So we must show that $w(C)$ must be large. Each literal in C removes one edge from G while constructing G_C. Counting the sizes of the cuts that isolate components of G_C, we count each deleted edge twice. So

$$2w(C) = \sum_{i=1}^{k} |\delta(V_i, V \setminus V_i)| = \sum_{i:n_i \leq n/2} \underbrace{|\delta(V_i, V \setminus V_i)|}_{Q1} + \sum_{i:n_i > n/2} \underbrace{|\delta(V_i, V \setminus V_i)|}_{Q2}$$

By the c-expansion property of G, $Q1 \geq cn_i$.
If $n_i > n/2$, it still cannot be too large because of the promise. Recall

$$S = \frac{n^2}{9} \geq \frac{|C \cap X_5|}{|C \cap X_3|} \geq \frac{1}{3} \sum_{i=1}^{k} \binom{n_i - 1}{2}$$

If any n_i is very large, say larger than $5n/6$, then the contribution from that component alone, $\frac{1}{3}\binom{n_i-1}{2}$, will exceed our chosen $S = \frac{n^2}{9}$. So each $n_i \leq 5n/6$. Thus even when $n_i > n/2$, we can conclude that $n_i/5 \leq n/6 \leq n - n_i < n/2$. By expansion of $V \setminus V_i$, we have $Q2 \geq c(n - n_i) \geq cn_i/5$.

$$\begin{aligned}2w(C) &= \sum_{i:n_i \leq n/2} \underbrace{|\delta(V_i, V \setminus V_i)|}_{Q1} + \sum_{i:n_i > n/2} \underbrace{|\delta(V_i, V \setminus V_i)|}_{Q2} \\ &\geq \sum_{i:n_i \leq n/2} cn_i + \sum_{i:n_i > n/2} \frac{cn_i}{5} \geq cn/5\end{aligned}$$

Choose c-expanders where c ensures $w(C) + 1 - n = \Omega(n)$. (Any constant $c > 10$.) Going back to our calculation of A from Equation 3),

$$A = \mathop{\mathbb{E}}_{C\sim\mu}\left[\frac{1}{\mu(C)}\middle|\frac{|C \cap X_5|}{|C \cap X_3|} \leq S\right] \geq \mathop{\mathbb{E}}_{C\sim\mu}\left[\frac{2^{w(C)+1-n}}{3}\middle|\frac{|C \cap X_5|}{|C \cap X_3|} \leq S\right] \geq \frac{1}{3}2^{cn/10+1-n} = 2^{\Omega(n)}$$

for suitable $c > 10$. Thus $|\mathcal{C}| \geq |\mathcal{C}'| \geq A \times B \geq 2^{\Omega(n)} \times (1/10)$. □

4.3 Lifting degree lower bounds to size

We describe a general technique to lift lower bounds on conical junta degree to size lower bounds for SubCubeSums.

Theorem 14. *Let d be the minimum degree of a SubCubeSums refutation of an unsatisfiable CNF formula F. Then every SubCubeSums refutation of $F \circ \oplus$ has size* $\exp(\Omega(d))$.

Before proving this theorem, we establish two lemmas. For a function $h : \{0,1\}^n \to \mathbb{R}$, define the function $h \circ \oplus : \{0,1\}^{2n} \to \mathbb{R}$ as $(h \circ \oplus)(\alpha_1, \alpha_2) = h(\alpha_1 \oplus \alpha_2)$, where $\alpha_1, \alpha_2 \in \{0,1\}^n$ and the $\oplus$ in $\alpha_1 \oplus \alpha_2$ is taken bitwise.

Lemma 15. $\mathrm{viol}_F(\alpha_1 \oplus \alpha_2) = \mathrm{viol}_{F\circ\oplus}(\alpha_1, \alpha_2)$.

Proof. Fix assignments α_1, α_2 and let $\alpha = \alpha_1 \oplus \alpha_2$. We claim that for each clause $C \in F$ falsified by α there is exactly one clause $D \in F \circ \oplus$ that is falsified by $\alpha_1\alpha_2$. Indeed, by the definition of composed formula the assignment $\alpha_1\alpha_2$ falsifies $C \circ \oplus$, hence the assignment falsifies some clause $D \in C \circ \oplus$. However, the clauses in the CNF expansion of $C \circ \oplus$ have disjoint subcubes, hence $\alpha_1\alpha_2$ falsifies at most one clause from the same block. Observing that if α does not falsify C, then $\alpha_1\alpha_2$ does not falsify any clause in $C \circ \oplus$ completes the proof. □

Note that Lemma 15 may not be true for gadgets other than $\oplus$.

Corollary 16. $\mathrm{viol}_{F\circ\oplus} - 1 = ((\mathrm{viol}_F) \circ \oplus) - 1 = (\mathrm{viol}_F - 1) \circ \oplus$.

Proof. $((\mathrm{viol}_F - 1) \circ \oplus)(\alpha_1, \alpha_2) = (\mathrm{viol}_F - 1)(\alpha_1 \oplus \alpha_2) = (\mathrm{viol}_F)(\alpha_1 \oplus \alpha_2) - 1 = (\mathrm{viol}_{F\circ\oplus})(\alpha_1, \alpha_2) - 1$. □

Lemma 17. *If $f \circ \oplus_2$ has a (integral) conical junta of size s, then f has a (integral) conical junta of degree $d = \mathrm{O}(\log s)$.*

Proof. Let J be a conical junta of size s that computes $f \circ \oplus_2$. Let ρ be the following random restriction: for each original variable x of f, pick $i \in \{0, 1\}$ and $b \in \{0, 1\}$ uniformly and set $x_i = b$. Consider a term C of J of degree at least $d > \log_{4/3} s$. The probability that C is not zeroed out by ρ is at most $(3/4)^d < 1/s$, hence by a union bound the probability that the junta $J\restriction_\rho$ has degree larger than d is at most $s \cdot (3/4)^d < 1$. Hence there is a restriction ρ such that $J\restriction_\rho$ is a junta of degree at most d, although not one that computes f. Since for each original variable x, ρ sets exactly one of the variables x_0, x_1, flipping the appropriate surviving variables—those where x_i is set to 1—gives a junta of degree at most d for f. □

Now we can prove Theorem 14.

Proof. We prove the contrapositive: if $F \circ \oplus$ has a SubCubeSums proof of size s, then there is an integral conical junta for $g = \mathrm{viol}_F - 1$ of degree $\mathrm{O}(\log s)$.

Let H be the collection of cubes in the SubCubeSums proof for $F \circ \oplus$. So $\mathrm{viol}_{F\circ\oplus} - 1 = \mathrm{viol}_H$. By Corollary 16, there is an integral conical junta for $(\mathrm{viol}_F - 1) \circ \oplus$ of size s. By Lemma 17 there is an integral conical junta for $\mathrm{viol}_F - 1$ of degree $\mathrm{O}(\log s)$. □

Recovering the Tseitin lower bound: This theorem, along with the $\Omega(n)$ conical junta degree lower bound of [8], yields an exponential lower bound for the SubCubeSums and MaxResW refutation size for Tseitin contradictions.

A candidate for separating Res from SubCubeSums: We conjecture that the SubCubeSums degree of the pebbling contradiction on the pyramid graph, or on a minor modification of it (a stack of butterfly networks, say, at the base of a pyramid), is $n^{\Omega(1)}$. This, along with Theorem 14 would imply that $F \circ \oplus$ is hard for SubCubeSums, thereby separating it from Res. However we have not yet been able to prove the desired degree lower bound. We do know that SubCubeSums degree is not exactly the same as Res width – for small examples, a brute-force computation has shown SubCubeSums degree to be strictly larger than Res width.

5 Discussion

We placed MaxRes(W) in a propositional proof complexity frame and compared it to more standard proof systems, showing that MaxResW is between tree-like resolution (strictly) and resolution. With the goal of also separating MaxRes and resolution we devised a new lower bound technique, captured by SubCubeSums, and proved lower bounds for MaxRes without relying on Res lower bounds.

Perhaps the most conspicuous open problem is whether our conjecture that pebbling contradictions composed with XOR separate Res and SubCubeSums holds. It also remains open to show that MaxRes simulates TreeRes – or even MaxResW – or that they are incomparable instead.

Acknowledgments

Part of this work was done when the last author was at TIFR, Mumbai, India. Some of this work was done in the conducive academic environs of the Chennai Mathematical Institute (during the CAALM workshop of CNRS UMI ReLaX, 2019), Banff International Research Station BIRS (seminar 20w5144) and Schloss Dagstuhl Leibniz Centre for Informatics (seminar 20061). The authors also thank Susanna de Rezende, Tuomas Hakoniemi, and Aaron Potechin for useful discussions.

References

1. Ben-Sasson, E., Impagliazzo, R., Wigderson, A.: Near optimal separation of tree-like and general resolution. Combinatorica **24**(4), 585–603 (Sep 2004)
2. Ben-Sasson, E., Wigderson, A.: Short proofs are narrow—resolution made simple. Journal of the ACM **48**(2), 149–169 (Mar 2001), preliminary version in *STOC '99*
3. Blake, A.: Canonical expressions in Boolean algebra. Ph.D. thesis, University of Chicago (1937)
4. Bonet, M.L., Buss, S., Ignatiev, A., Marques-Silva, J., Morgado, A.: MaxSAT resolution with the dual rail encoding. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, (AAAI-18). pp. 6565–6572 (2018)
5. Bonet, M.L., Levy, J., Manyà, F.: Resolution for Max-SAT. Artificial Intelligence **171**(8), 606 – 618 (2007)
6. Cook, S.A.: An observation on time-storage trade off. Journal of Computer and System Sciences **9**(3), 308–316 (1974), preliminary version in *STOC '73*
7. Fleming, N., Kothari, P., Pitassi, T.: Semialgebraic proofs and efficient algorithm design. Foundations and Trends in Theoretical Computer Science **14**(1-2), 1–221 (2019)
8. Göös, M., Jain, R., Watson, T.: Extension complexity of independent set polytopes. SIAM Journal on Computing **47**(1), 241–269 (Feb 2018)
9. Göös, M., Lovett, S., Meka, R., Watson, T., Zuckerman, D.: Rectangles are nonnegative juntas. SIAM Journal on Computing **45**(5), 1835–1869 (Oct 2016), preliminary version in *STOC '15*
10. Ignatiev, A., Morgado, A., Marques-Silva, J.: On tackling the limits of resolution in SAT solving. In: Theory and Applications of Satisfiability Testing - SAT. pp. 164–183 (2017)
11. Iwama, K., Miyano, E.: Intractability of read-once resolution. In: Structure in Complexity Theory Conference. pp. 29–36. IEEE Computer Society (1995)
12. Larrosa, J., Heras, F., de Givry, S.: A logical approach to efficient max-sat solving. Artif. Intell. **172**(2-3), 204–233 (2008)
13. Larrosa, J., Rollon, E.: Augmenting the power of (partial) maxsat resolution with extension. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence (2020)
14. Loff, B., Mukhopadhyay, S.: Lifting theorems for equality. In: Proceedings of the 36th Symposium on Theoretical Aspects of Computer Science (STACS '19). Leibniz International Proceedings in Informatics (LIPIcs), vol. 126, pp. 50:1–50:19 (Mar 2019)
15. Marques-Silva, J., Ignatiev, A., Morgado, A.: Horn maximum satisfiability: Reductions, algorithms and applications. In: 18th EPIA Conference on Artificial Intelligence. pp. 681–694 (2017)

16. Mikša, M., Nordström, J.: Long proofs of (seemingly) simple formulas. In: Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14). Lecture Notes in Computer Science, vol. 8561, pp. 121–137. Springer (Jul 2014)
17. Narodytska, N., Bacchus, F.: Maximum satisfiability using core-guided maxsat resolution. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence. pp. 2717–2723 (2014)
18. Pudlák, P., Impagliazzo, R.: A lower bound for DLL algorithms for k-SAT (preliminary version). In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '00). pp. 128–136 (Jan 2000)
19. Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal of the ACM **12**, 23–41 (1965)
20. Spence, I.: sgen1: A generator of small but difficult satisfiability benchmarks. Journal of Experimental Algorithmics **15**, 1.2:1–1.2:15 (Mar 2010)
21. Urquhart, A.: Hard examples for resolution. Journal of the ACM **34**(1), 209–219 (Jan 1987)
22. Van Gelder, A., Spence, I.: Zero-one designs produce small hard SAT instances. In: Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT '10). Lecture Notes in Computer Science, vol. 6175, pp. 388–397. Springer (Jul 2010)

Appendix

An example for Lemma 3

Consider the tree-like resolution proof in Figure 1 Following the procedure in the

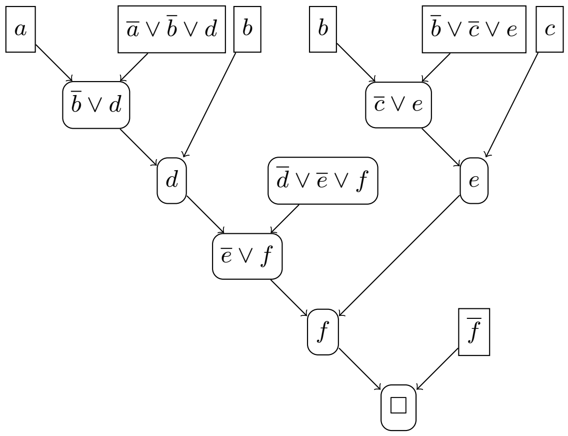

a
$\overline{a} \vee \overline{b} \vee d$
b
b
$\overline{b} \vee \overline{c} \vee e$
c
$\overline{b} \vee d$
$\overline{c} \vee e$
d
$\overline{d} \vee \overline{e} \vee f$
e
$\overline{e} \vee f$
f
$\overline{f}$
$\Box$

Fig. 1. A tree-like resolution proof

proof of the Lemma, the axiom b is weakened to $b \vee e$ and $b \vee \neg e$, since e is the pivot variable at the branching point where b is used in both sub-derivations.

Short proofs in SubCubeSums for subset cardinality formulas

In proving the upper bound in Lemma 12, we invoked implicational completeness from Proposition 2. However, in our case the numbers are small enough that we can show how to derive equation (1) explicitly, by solving the appropriate LP, and without relying on Proposition 2. As a curiosity we display them next. We have

$$\begin{aligned}
&\overline{x_{1,2,3}} + \overline{x_{1,2,4}} + \overline{x_{1,3,4}} + \overline{x_{2,3,4}} - (2 - x_1 - x_2 - x_3 - x_4) = \\
&2x_1x_2x_3x_4 + x_1x_2x_3\overline{x_4} + x_1x_2\overline{x_3}x_4 + x_1\overline{x_2}x_3x_4 + \overline{x_1}x_2x_3x_4 + 2\overline{x_1x_2x_3x_4}
\end{aligned}$$

and

$$\begin{aligned}
&\overline{x_{1,2,3}} + \overline{x_{1,2,4}} + \overline{x_{1,2,5}} + \overline{x_{1,3,4}} + \overline{x_{1,3,5}} + \overline{x_{1,4,5}} + \overline{x_{2,3,4}} + \overline{x_{2,3,5}} + \overline{x_{2,4,5}} \\
&+ \overline{x_{3,4,5}} - (3 - x_1 - x_2 - x_3 - x_4 - x_5) = \\
&2x_1x_2x_3x_4x_5 + x_1x_2x_3x_4\overline{x_5} + x_1x_2x_3\overline{x_4}x_5 + x_1x_2\overline{x_3}x_4x_5 + x_1\overline{x_2}x_3x_4x_5 \\
&+ \overline{x_1}x_2x_3x_4x_5 + 2\overline{x_1x_2x_3x_4}x_5 + 2\overline{x_1}x_2\overline{x_3}x_4\overline{x_5} \\
&+ 2\overline{x_1x_2}x_3\overline{x_4x_5} + 2\overline{x_1}x_2\overline{x_3x_4x_5} + 2x_1\overline{x_2x_3x_4x_5} + 7\overline{x_1x_2x_3x_4x_5}
\end{aligned}$$