The Power of Negative Reasoning

de Rezende, Susanna F.; Lauria, Massimo; Nordström, Jakob; Sokolov, Dmitry

Published in:
36th Computational Complexity Conference, CCC 2021

DOI:
10.4230/LIPIcs.CCC.2021.40

Publication date:
2021

Document version
Publisher's PDF, also known as Version of record

Document license:
CC BY

Citation for published version (APA):
https://doi.org/10.4230/LIPIcs.CCC.2021.40

Download date: 24. maj. 2022
The Power of Negative Reasoning

Susanna F. de Rezende
Institute of Mathematics of the Czech Academy of Sciences, Prague, Czech Republic

Massimo Lauria
Sapienza Università di Roma, Italy

Jakob Nordström
University of Copenhagen, Denmark
Lund University, Sweden

Dmitry Sokolov
St. Petersburg State University, Russia
PDMI RAS, St. Petersburg, Russia

Abstract

Semialgebraic proof systems have been studied extensively in proof complexity since the late 1990s to understand the power of Gröbner basis computations, linear and semidefinite programming hierarchies, and other methods. Such proof systems are defined alternately with only the original variables of the problem and with special formal variables for positive and negative literals, but there seems to have been no study how these different definitions affect the power of the proof systems. We show for Nullstellensatz, polynomial calculus, Sherali-Adams, and sums-of-squares that adding formal variables for negative literals makes the proof systems exponentially stronger, with respect to the number of terms in the proofs. These separations are witnessed by CNF formulas that are easy for resolution, which establishes that polynomial calculus, Sherali-Adams, and sums-of-squares cannot efficiently simulate resolution without having access to variables for negative literals.

2012 ACM Subject Classification Theory of computation → Proof complexity; Computing methodologies → Representation of polynomials

Keywords and phrases Proof complexity, Polynomial calculus, Nullstellensatz, Sums-of-squares, Sherali-Adams

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.40

Funding Susanna F. de Rezende was supported by Knut and Alice Wallenberg grant KAW 2018.0371. Jakob Nordström received funding from the Swedish Research Council grant 2016-00782 and the Independent Research Fund Denmark grant 9040-00389B. Part of this work was carried out while visiting the Simons Institute for the Theory of Computing.

Acknowledgements We thank Or Meir for fruitful discussions and the anonymous reviewers for comments on the presentation.

1 Introduction

Given a set of polynomial equalities

\[ p_j = 0 \quad j \in [m] \]  

(1)

and/or inequalities

\[ r_j \geq 0 \quad j \in [\ell] \]  

(2)

in some field \( \mathbb{F} \) (which should be ordered if \( \ell > 0 \)), the problem of determining whether there exists solutions satisfying all constraints is a natural and well-known \( \text{NP} \)-hard problem. If one includes among the equalities (1) also equations \( x_i^2 - x_i = 0 \) for all variables \( x_i \), then this setting can also be used to decide satisfiability of formulas in conjunctive normal
form (CNF). This is done by identifying 1 with \textit{true} and 0 with \textit{false} and then translating disjunctive clauses like \( x_1 \lor x_2 \lor x_k \) into equalities \( (1 - x_1)x_2(1 - x_3) = 0 \) or inequalities \( x_1 + (1 - x_2) + x_3 \geq 1 \) (using the \textit{multiplicative} or \textit{additive} translation, respectively).

For polynomial equalities as in (1), it follows from (a mild extension of) Hilbert’s Nullstellensatz that there is no solution if and only if there are polynomials \( q_j \) such that the syntactic equality

\[
\sum_{j \in [m]} q_j p_j = 1
\]

holds. Such \textit{Nullstellensatz certificates} can be viewed as proof system in the sense of Cook and Reckhow [15], and the study of this \textit{Nullstellensatz} proof system was initiated in [7]. In the \textit{polynomial calculus} proof system introduced in [14] such certificates can be constructed step by step by explicitly deriving polynomials in the ideal generated by \{ \( p_j \mid j \in [m] \} \). This can be seen to correspond to \textit{Gröbner basis} computations, which can potentially yield more concise certificates of unsatisfiability. When there are also inequalities (2), linear combinations of polynomial products

\[
\sum_{j \in [m]} q_j p_j + \sum_{j \in [\ell]} s_j r_j = -1
\]

for \( s_j \geq 0 \) with different syntactic restrictions yield proof systems such as \textit{Sherali-Adams} [38] and \textit{sums-of-squares (SOS)} [30, 25], corresponding to linear and semidefinite programming hierarchies.

By now there is a rich literature on upper and lower bounds for these proof systems. An excellent general reference on proof complexity is [28]. For more details on Nullstellensatz and polynomial calculus the reader can consult [13] and the references therein, and a recent survey covering Sherali-Adams and sums-of-squares is [21].

1.1 Encoding of Variables and Literals

One slightly annoying aspect when translating CNF formulas to the algebraic setting described above is that the translation is quite sensitive to the signs of the literals in clauses. Normally, polynomials are represented as linear combinations of monomials, which means that a clause

\[
x_1 \lor x_2 \lor \cdots \lor x_3
\]

with \( k \) positive literals turns into a polynomial equation

\[
\prod_{i=1}^{k} (1 - x_i) = 0
\]

with \( 2^k \) monomials if we use the multiplicative translation. This problem does not immediately arise for the additive translation, but it is still conceivable that it could be helpful to encode polynomials of the form (5b) more concisely.

This problem was perhaps first addressed in [1], where a version of polynomial calculus was defined with separate formal variables \( \overline{x}_i \) for negative literals, together with equations \( x_i + \overline{x}_i - 1 = 0 \) enforcing the intended meaning of negation. This proof system was called \textit{polynomial calculus with resolution}, or \textit{PCR} for short, in [1], since the introduction of negative literals can be seen to allow polynomial calculus to simulate the \textit{resolution} proof system efficiently, but in this paper we will refer to this flavour of the proof system as \textit{polynomial}
calculus with negative literals (as opposed to polynomial calculus without negative literals). When the proof system has access to separate variables for positive and negative literals, this ensures that lower bounds do not depend on the choice of signs for literals encoding the input, but reflect more intrinsic properties of the problem under study. As far as we are aware, essentially all lower bounds for polynomial calculus holds even when negative literals are allowed (with the exception of some of the lower bounds in [20]), and to the best of our knowledge there are no polynomial calculus upper bounds that are known to hold only for polynomial calculus with negative literals and not for polynomial calculus without them. Papers such as [5, 31, 11, 4] have studied the Sherali-Adams and sums-of-squares proof systems both with and without variables for negative literals, but again without really distinguishing between the two versions of the proof system thus obtained.

The purpose of this work is to understand if and how the introduction of formal variables for negative literals affect the power of reasoning of (semi)algebraic proof systems. This is arguably quite a natural question, and we find it somewhat surprising that nothing seems to be known regarding how the two variants of these (semi)algebraic proof systems are related.

Somewhat intriguingly, this does not seem to be just a theoretical concern. For, e.g., Gröbner basis computations, one could expect that this whole question should be irrelevant, since the basis reduction algorithm will immediately remove whichever literal over a given variable that comes later in the order. This appears not to be the case, however, and papers such as [36, 27] use “bit-flipping” (i.e., the introduction of formal variables for negated literals) to try to avoid blow-ups in polynomial size during hardware circuit verification.

1.2 Our Results

We show that for all of the proof systems Nullstellensatz, polynomial calculus, Sherali-Adams, and sums-of-squares, adding separate formal variables for negative literals results in an exponential increase in power. Our main results can be summarized as follows (where we refer to Section 2 for the missing formal definitions).

\textbf{Theorem 1.} Let $\mathcal{P}$ be any of the proof systems Nullstellensatz or polynomial calculus (over any field), or Sherali-Adams or sums-of-squares. Then there is a family of CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size polynomial in $n$ such that the proof system $\mathcal{P}$ has polynomial size refutations of $F_n$ that use formal variables for negative literals, whereas $\mathcal{P}$ refutations of $F_n$ requires exponential size when such formal variables are not allowed.

We remark that, except for sums-of-squares, the separating formulas above are CNFs of constant width. It is known from [1, 5] that polynomial calculus, Sherali-Adams, and sums-of-squares over literals can simulate the resolution proof system efficiently. Since the formulas in Theorem 1 are all easy for resolution, it follows that negative literals are necessary for the simulation.

\textbf{Corollary 2.} None of the proof systems polynomial calculus, Sherali-Adams, or sums-of-squares can polynomially simulate resolution, unless formal variables for negative literals are allowed.

For Nullstellensatz and polynomial calculus we also give some more refined separation results involving size-degree trade-offs and space-degree trade-offs.

1.3 Outline of This Paper

In Section 2 we review the relevant preliminaries. In Section 3 we establish our separation results for polynomial calculus. Analogous results for Sherali-Adams and sums-of-squares are obtained in Section 4, and the separation for Sherali-Adams is sharpened somewhat in Section 5. Our results for Nullstellensatz are presented in Section 6.
2 Preliminaries

We encode propositional variables as algebraic variables with \{0, 1\} values, with the intended meaning that 1 represents true and 0 represents false. For each variable \(x\) we consider a corresponding variable \(\overline{x}\) that represents the logical negation of \(x\), i.e., it holds that \(\overline{1} = 1 - x\). We say that \(x\) is a positive literal and \(\overline{x}\) is a negative literal. A (partial) boolean assignment \(\rho\) is a mapping from some algebraic variables to \{0, 1\}, with the constraint that \(x \in \text{dom}(\rho)\) if and only if \(\overline{x} \in \text{dom}(\rho)\) and that in such case \(\rho(x) = 1 - \rho(\overline{x})\). Given a polynomial \(p\) the restriction of \(p\) by \(\rho\), denoted as \(p|_{\rho}\), is the polynomial obtained from \(p\) by substituting in it all variables \(x \in \text{dom}(\rho)\) with the corresponding value \(\rho(x)\). Given a set of polynomials \(S\) we denote as \(S|_{\rho}\) the set of restricted polynomials. A random restriction is a distribution over partial boolean assignments. A polynomial is multilinear if no variable appears with degree larger than one and no monomial contains two opposite literals.

For a set \(S = \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\tau \geq 0\}\) of polynomial equations and inequalities we say that a boolean assignment satisfies \(S\) if it satisfies all the equations and inequalities in it. We say that \(S\) implies an equation \(p = 0\) when every boolean assignment which satisfies \(S\) also satisfies \(p = 0\). In the same way \(S\) implies an inequality \(r \geq 0\) when every boolean assignment which satisfies \(S\) also satisfies \(r \geq 0\). We now discuss encodings of a clause

\[
x_1 \lor \cdots \lor x_j \lor \neg x_{j+1} \lor \cdots \lor \neg x_k
\]

into polynomial constraints

\[
(1 - x_1) \cdots (1 - x_j) \cdot x_{j+1} \cdots x_k = 0,
\]

\[
\overline{x}_1 \cdots \overline{x}_j \cdot x_{j+1} \cdots x_k = 0,
\]

and

\[
x_1 + \cdots + x_j + (1 - x_{j+1}) + \cdots + (1 - x_k) \geq 1.
\]

A clause (6a) is naturally encoded as the polynomial equation (6b), which has \(2^k\) monomials of degree up to \(k\). Using negative literals we get the more efficient encoding (6c) which has a single monomial. We would like to stress that (6b) and (6c) are algebraic representations of the same boolean function, even though they are syntactically different. For semi-algebraic proofs, clauses are naturally represented as inequalities (6d).

We now define all proof systems discussed in this paper.

Resolution. We first introduce some basic notation. We denote the negation of a variable \(x\) by \(\neg x\) or \(\overline{x}\). The width of a clause \(C\) is the number of literals in \(C\). A CNF formula is a conjunction of clauses and a width-\(k\) CNF formula, or simply a \(k\)-CNF formula, is a CNF formula where every clause has width at most \(k\). A resolution proof from a CNF formula \(F\) of a clause \(C\) is a sequence of clauses \((C_1, \ldots, C_\tau)\) such that \(C_\tau = C\) and, for each \(i \in [\tau]\), \(C_i\) is either a clause of \(F\), or is some clause \(C_j \lor D\) obtained by weakening a clause \(C_j\), for some \(j < i\), or is derived from \(C_j\) and \(C_{j'}\), for some \(j, j' < i\) by applying the resolution rule

\[
\frac{B \lor x}{B \lor D \lor \neg x},
\]

where \(C_j = B \lor x\), \(C_{j'} = D \lor \neg x\), and \(C_i = B \lor D\). When applying rule (7), we say that we resolve on \(x\). The size/length of a resolution proof \((C_1, \ldots, C_\tau)\) is \(\tau\) and its width is the maximum width of any clause in the proof. A resolution refutation (i.e., proof of unsatisfiability) of \(F\) is a proof of the empty clause \(\bot\) from it.
A resolution proof \((C_1, \ldots, C_\tau)\) can also be viewed as a DAG, with nodes \([\tau]\) and, for all \(i, j \in [\tau]\), a directed edge from \(j\) to \(i\) if \(C_j\) was used to derive \(C_i\). The depth of a proof is the length of the longest directed path in the underlying DAG. If the DAG is a tree the proof is tree-like.

The following (semi-)algebraic proof systems reason about polynomial equations and/or inequalities over \([0,1]\), expressed in term of variables representing positive and negative literals. To deal with CNF formulas we use the encodings (6), plus appropriate axioms enforcing boolean values.

**Nullstellensatz.** Consider an initial set of polynomial equations \(S = \{p_1 = 0, \ldots, p_m = 0\}\) over a field \(F\) and over variables \(x_1, \ldots, x_n, \tau_1, \ldots, \tau_s\), where we require that the set \(S\) to include variable axioms \(x_i^2 - x_i = 0\), \(\tau_i^2 - \tau_i = 0\) and \(x_i + \tau_i - 1 = 0\) for each \(i \in [n]\). A Nullstellensatz (NS) proof of \(p = 0\) from \(S\) is a set of polynomials \(\{q_1, \ldots, q_m\}\) in \(F[x_1, \ldots, x_n, \tau_1, \ldots, \tau_s]\) such that

\[
\sum_{j \in [m]} q_j p_j = 0,
\]  

(8)

where we stress that the equality is syntactical. Since all polynomials in \(S\) are zero by hypothesis, the proof is sound. The \((\text{monomial})\) size of any such proof is the sum over \(j \in [m]\) of the number of monomials occurring in each polynomial \(q_j p_j\), when expanded out as a linear combination of monomials. The degree of any such proof is the maximum degree among all \(q_j p_j\) for \(j \in [m]\). A refutation of \(S\) is a proof of the equation \(1 = 0\). A refutation of a CNF formula \(F\) in NS is a refutation of a set \(S\) of polynomials containing the variable axioms specified above plus the clauses of \(F\) encoded as in (6b), unless a different encoding is specified.

**Proposition 3** (NS with negative literals simulates tree-like resolution). Let \(F\) be an unsatisfiable CNF formula that has a tree-like resolution refutation of \(F\) in size \(s\) and depth \(d\). Then the set of polynomial equations obtained by encoding each clause of \(F\) as in (6c) has an NS refutation with negative literals in size \(2s - 1\) and degree \(d\).

**Polynomial calculus.** As was the case for Nullstellensatz, we consider an initial set of polynomial equations \(S = \{p_1 = 0, \ldots, p_m = 0\}\) over a field \(F\) and over variables \(x_1, \ldots, x_n, \tau_1, \ldots, \tau_s\), and we require that \(\tau\) include variable axioms \(x_i^2 - x_i = 0\), \(\tau_i^2 - \tau_i = 0\) and \(x_i + \tau_i - 1 = 0\) for each \(i \in [n]\). A polynomial calculus (PC) proof of \(p = 0\) from \(S\) is a sequence of polynomials \(\{q_1, q_2, \ldots, q_\tau\}\) in \(F[x_1, \ldots, x_n, \tau_1, \ldots, \tau_s]\) such that \(q_\tau = p\) and each \(q_i\) for \(1 \leq t \leq \tau\) is either

- some polynomial \(p_j\) with \(p_j = 0 \in S\);
- a linear combination \(\alpha q_1 + \beta q_2\) for some \(\alpha, \beta \in F\) and \(1 \leq t_1, t_2 < \tau\);
- a multiplication \(x \cdot q_\nu\) for some \(t' < t\) and variable \(x = x_i\) or \(x = \tau_i\).

When the equations in \(S\) are satisfied, all derived polynomials, \(p\) in particular, are zero. The \((\text{monomial})\) size of such a proof is the sum over \(1 \leq t \leq \tau\) of the number of monomials occurring in each polynomial \(q_t\), when written as a sum of monomials. The degree of such a proof is the maximum degree among all \(q_t\) for \(1 \leq t \leq \tau\). A refutation of \(S\) is a proof of \(1 = 0\). A refutation of a CNF formula \(F\) in PC is a refutation of a set \(S\) of polynomials containing the variable axioms specified above plus the clauses of \(F\) encoded as in (6b), unless a different encoding is specified. It is a simple observation that when dealing with CNF formulas of constant width, it is possible to efficiently deduce the representation (6b) from the representation (6c) and vice versa.

We stress that all results proved here for NS and PC hold independently of the field \(F\).
Sherali-Adams. We consider an initial set of polynomial equations and inequalities \( \mathcal{S} = \{ p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_t \geq 0 \} \) over the real field and over variables \( x_1, \ldots, x_n \), and we require that the set \( \mathcal{S} \) include, for each \( i \in [n] \), variable axioms \( x_i^2 - x_i = 0, \pi_i^2 - \pi_i = 0, x_i + \pi_i - 1 = 0, x_i \geq 0, \pi_i \geq 0, 1 - x_i \geq 0 \), and \( 1 - \pi_i \geq 0 \). We also assume that \( \mathcal{S} \) includes the axiom \( 1 \geq 0 \). We refer to an arbitrary product of factors of the form \( x_1, \pi_i, 1 - x_1, 1 - \pi_i \) as a \textit{generalized monomial}.\(^1\) A Sherali-Adams (SA) proof/derivation of \( r \geq 0 \) from \( \mathcal{S} \) is a set of polynomials \( \{ q_1, \ldots, q_m; s_1, \ldots, s_r \} \) such that

\[
\sum_{j \in [m]} q_j p_j + \sum_{j \in [r]} s_j r_j = r ,
\]

where each \( s_j \) is a positive linear combination of generalized monomials. That is, \( s_j \) can be written as \( s_j = \sum_i \alpha_j,i h_{j,i} \) for some \( \alpha_j,i \)'s that are positive real numbers and \( h_{j,i} \)'s that are generalized monomials. Under the assumption that all polynomial equations and inequalities in \( \mathcal{S} \) are satisfied, the addends \( q_j p_j \) are equal to zero and the addends \( s_j r_j \) are nonnegative; hence \( r \geq 0 \).

The \textit{(monomial) size} of an SA proof is the sum over \( j \in [m] \) and \( j \in [r] \) of the number of monomials occurring in each summand in \( (9) \), when written as a sum of monomials. The \textit{degree} of an SA proof is the maximum degree among all \( q_j p_j \) for \( j \in [m] \) and all \( s_j r_j \) for \( j \in [r] \). An SA refutation of \( \mathcal{S} \) is an SA proof of \(-1 \geq 0 \). A refutation of a CNF formula \( F \) in \( \text{SA} \) is a refutation of a set \( \mathcal{S} \) of polynomials containing the variable axioms specified above plus the clauses of \( F \) encoded as in \((6d)\), unless a different encoding is specified.

Sums-of-squares. As was the case for Sherali-Adams, we consider an initial set of polynomial equations and inequalities \( \mathcal{S} = \{ p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_t \geq 0 \} \) over the real field and over variables \( x_1, \ldots, x_n \), and we require that \( \mathcal{S} \) include, for each \( i \in [n] \), \textit{variable axioms} \( x_i^2 - x_i = 0, \pi_i^2 - \pi_i = 0, x_i + \pi_i - 1 = 0, x_i \geq 0, \pi_i \geq 0, 1 - x_i \geq 0 \), and \( 1 - \pi_i \geq 0 \), and also the axiom \( 1 \geq 0 \). A sum of squares (SOS) proof of \( r \geq 0 \) from \( \mathcal{S} \) is a set of polynomials \( \{ q_1, \ldots, q_m; s_1, \ldots, s_r \} \) in \( \mathbb{R}[x_1, \ldots, x_n, \pi_1, \ldots, \pi_n] \) such that

\[
\sum_{j \in [m]} q_j p_j + \sum_{j \in [r]} s_j r_j = r ,
\]

where each \( s_j \) is a positive linear combination of squared polynomials, that is, \( s_j \) can be written as \( s_j = \sum_i \alpha_j,i h_{j,i}^2 \) for some \( \alpha_j,i \)'s that are positive real numbers and \( h_{j,i}^2 \)'s that are polynomials. Under the assumption that all polynomial equations and inequalities in \( \mathcal{S} \) are satisfied, the summands \( q_j p_j \) are equal to zero and the summands \( s_j r_j \) are nonnegative; hence, \( r \geq 0 \).

The \textit{(monomial) size} of an SOS proof is the sum over \( j \in [m] \) and \( j \in [r] \) of the number of monomials occurring in each summand in \( (10) \), when written as a sum of monomials. The \textit{degree} of an SOS proof is the maximum degree among all \( q_j p_j \) for \( j \in [m] \) and all \( s_j r_j \) for \( j \in [r] \). An SOS refutation of \( \mathcal{S} \) is an SOS proof of \(-1 \geq 0 \). A refutation of a CNF formula \( F \) in \( \text{SOS} \) is a refutation of a set \( \mathcal{S} \) of polynomials containing the variable axioms specified above plus the clauses of \( F \) encoded as in \((6d)\), unless a different encoding is specified.

For the rest of the paper we say a proof in either \text{NS}, \text{PC}, \text{SA}, or \text{SOS} is \textit{without negative literals} if none of the variables \( \pi_1, \ldots, \pi_n \) occur in any of the polynomials occurring in the proof. Otherwise we say that the proof is \textit{with negative literals}.

\(^1\) For instance \((1 - x_2)x_3\pi_4x_5(1 - \pi_9)\) is a generalized monomial. It is positive under the assumption that all variables are between 0 and 1.
Proposition 4. Consider a CNF formula $F$ with a resolution refutation of length $L$ and width $w$. It holds that

- the clauses of $F$, encoded as in (6c), have a PC refutation with negative literals of size $O(L)$ and degree $w + 1$;
- when $F$ is a $k$-CNF formula with $m$ clauses, its representation using encoding (6b) has a PC refutation with negative literals of size $O(2^k m + L)$ and degree $w + 1$;
- the clauses of $F$, represented using encoding (6b), have a PC refutation without negative literals of size $O(2^n L)$ and degree $w + 1$.

Proposition 5 ([11]). Let $S := \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ be a set of polynomial equations and inequalities. If $S$ has a Sherali-Adams refutation of degree $d$ and size $N$, then it has a sums-of-squares refutation of degree $d + 1$ and size $N^c$ for some $c > 0$.

The next lemma is a fundamental tool for the results in the next section.

Lemma 6. Let $S$ be a set of monomials over (positive) variables $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$. There is a restriction $\rho$ that for all $i \in [n]$ sets exactly one of $\{y_i, z_i\}$ to 0 and is such that $S|_\rho$ has degree at most $\log|S|$.

Proof. We consider a random restriction $\rho$ that for each $i$, chooses either $y_i$ or $z_i$ with probability $1/2$ and sets it to 0. Note that a monomial of degree $d$ is set to 0 by $\rho$ with probability at least $1 - (1/2)^d$. Indeed, if the monomial contains both $y_i$ and $z_i$ for some $i \in [n]$, then it is set to 0 with probability 1; otherwise every variable is set to 0 independently with probability $1/2$ and thus the monomials is not set to 0 with probability $(1/2)^d$. Therefore, by union bound over all monomials in $S$ we have that

$$\Pr[S|_\rho \text{ has a monomial of degree } > \log|S|] \leq |S| \cdot (1/2)^{\log|S|+1} < 1. \quad (11)$$

We conclude that there is some restriction $\rho$ such that $S|_\rho$ has degree at most $\log|S|$.

3 Negative literals and polynomial calculus

The main goal of this section is to exhibit a formula that has short refutations in resolution but requires exponential size refutations in PC without negative literals. In particular, this implies that not using negative literals can lead to an exponential blow-up in the size of refutations. The starting point is the graph ordering principle, a formula introduced in [37] that falsly claims that it is possible to partially order vertices of some finite graph such that each vertex has at least one neighbour that is smaller (according to the ordering) than itself.

Consider a finite undirected graph $G = (V, E)$. The graph ordering principle on $G$, denoted as $\text{GOP}(G)$, is a CNF formula defined on propositional variables $x_{u,v}$ for every two distinct $u,v \in V$, with the intended meaning that $x_{u,v}$ is true when $u$ is smaller than $v$ in the partial order. The clauses of $\text{GOP}(G)$ are

$$\overline{x}_{u,v} \lor \overline{x}_{v,w} \lor x_{u,w} \quad \text{for every three distinct } u,v,w \in V, \quad (12a)$$
$$\overline{x}_{u,v} \lor \overline{x}_{v,u} \quad \text{for every two distinct } u,v \in V, \quad (12b)$$
$$\bigvee_{u : \{u,v\} \in E} x_{u,v} \quad \text{for every } v \in V. \quad (12c)$$

The graph ordering principle is a generalization of the ordering principle, considered for the first time in [29]. The latter principle falsely claims that it is possible to partially order a set of $n$ element so that no element is minimal. The ordering principle, expressed as a
The Power of Negative Reasoning

CNF formula, is often denoted by $\text{OP}_n$, and is exactly the formula $\text{GOP}(K_n)$, where $K_n$ is the complete graph over $n$ vertices. Proposition 7 claims an upper bound that holds for any graph, even the complete one. The degree lower bound in Proposition 8, however, holds only for specific families of expander graphs.

**Proposition 7 ([39]).** Given any graph $G$ with $n$ vertices and maximum degree $d$, the formula $\text{GOP}(G)$ is a $d$-CNF formula with $\Theta(n^2)$ variables and $\Theta(n^3)$ clauses. Furthermore, $\text{GOP}(G)$ has a resolution refutation of length $\Theta(n^3)$ where every clause in the refutation contains at most two negative literals.

**Proposition 8 ([23]).** There exists a sequence of graphs $\{G_n\}_n$ such that each $G_n$ has $\Theta(n)$ vertices and constant maximum degree $d$, and any PC refutation of $\text{GOP}(G_n)$ requires polynomials of degree $\Omega(n)$.

The degree lower bound implies, in particular, that any resolution refutation of $\text{GOP}(G_n)$ must have width $\Omega(n)$ (due to Proposition 4). Given the resolution upper bound in Proposition 7, the simulation of resolution in Proposition 4 gives a small PC refutation of $\text{GOP}(G_n)$ only when using with negative literals. This suggests that negative literals are essential to obtain small refutations of $\text{GOP}(G_n)$. Is this really the case? A positive answer would give us the separation we are looking for, but unfortunately we are not able to prove a size lower bound for refuting $\text{GOP}(G_n)$ in PC without negative literals. Instead, we compose $\text{GOP}(G_n)$ with the 2-bit OR function, thus obtaining a new formula that will remain easy for resolution but will be provably hard for PC without negative literals.

Let us make this construction explicit. We denote by $\text{GOP}^{\text{OR}}(G)$ the CNF formula obtained from $\text{GOP}(G)$ by substituting each variable $x_{u,v}$ in $\text{GOP}(G)$ by the disjunction of two fresh variables, $y_{u,v} \lor z_{u,v}$. In order to obtain a CNF formula, after the substitution we must apply distributivity. This process transforms a clause of width $k$, with $j$ negative literals and $k - j$ positive literals, into a set of $2^j$ clauses with $j$ negative literals and $2(k - j)$ positive literals. For example, see how the substitution transforms this clause with 2 negative literals

$$
\Pi_{u_1, v_1} \lor \Pi_{u_2, v_2} \lor x_{u_3, v_3} \lor x_{u_4, v_4} \lor \ldots \lor x_{u_k, v_k},
$$

into four clauses

$$
\Pi_{u_1, v_1} \lor \Pi_{u_2, v_2} \lor y_{u_3, v_3} \lor y_{u_4, v_4} \lor \ldots \lor y_{u_k, v_k} \lor z_{u_k, v_k},
$$

$$
\Pi_{u_1, v_1} \lor \Pi_{u_2, v_2} \lor y_{u_3, v_3} \lor y_{u_4, v_4} \lor \ldots \lor y_{u_k, v_k} \lor z_{u_k, v_k},
$$

$$
\Pi_{u_1, v_1} \lor \Pi_{u_2, v_2} \lor z_{u_3, v_3} \lor z_{u_4, v_4} \lor \ldots \lor z_{u_k, v_k} \lor y_{u_k, v_k},
$$

$$
\Pi_{u_1, v_1} \lor \Pi_{u_2, v_2} \lor z_{u_3, v_3} \lor z_{u_4, v_4} \lor \ldots \lor z_{u_k, v_k} \lor y_{u_k, v_k}.
$$

This transformation has not increased the size of the formula by much: $\text{GOP}^{\text{OR}}(G)$ has $\Theta(n^2)$ variables, $\Theta(n^3)$ clauses, and the maximum width of its clauses is at most 2 times the maximum width of a clause in $\text{GOP}(G)$. Moreover, $\text{GOP}^{\text{OR}}(G)$ still admits short resolution refutations.

**Lemma 9.** For every graph $G$ with $n$ vertices, the formula $\text{GOP}^{\text{OR}}(G)$ has a resolution refutation of length $\Theta(n^3)$ where every clause in the refutations contains at most two negative literals.
Proof. The idea is to use the resolution refutation of GOP\((G)\) from Proposition 7 as a scheme for the refutation of GOP\(^{OR}\)(\(G\)). Let \(C_1, C_2, \ldots, C_r\) be the sequence of clauses in this refutation. For each \(C_i\) we consider the set \(C_i\) of at most four clauses that we get by applying substitution (14) to it. Every clause in \(C_i\) has the same number of negative literals as \(C_i\), and that is at most two.

We show how to derive each \(C_i\) from GOP\(^{OR}\)(\(G\)) by induction on \(i\), assuming all previous set \(C_j\) for \(j < i\) have already been derived. Furthermore, we show that each such derivation takes a constant number of resolution steps.

If \(C_i\) is an initial clause of GOP\((G_n)\) then all clauses of \(C_i\) are in GOP\(^{OR}\)(\(G\)) by construction.

If \(C_i\) follows from \(C_j\) for some \(j < i\) by weakening, then each clause of \(C_i\) is a superset of some clause in \(C_j\) and thus follows from it by weakening. The remaining case is when \(C_i\) is derived by a resolution step from two previous clauses \(C_j\) and \(C_k\). Without loss of generality, we rewrite clause \(C_j\) as \(A \lor x_{u,v}\), clause \(C_k\) as \(B \lor \overline{x}_{u,v}\), and clause \(C_i\) as \(A \lor B\). The structure of sets \(C_j\) and \(C_k\) as follows,

\[
\begin{align*}
C_j & : \\
A_1 \lor y_{u,v} \lor z_{u,v} & & C_k : \\
A_2 \lor y_{u,v} \lor z_{u,v} & \lor \overline{y}_{u,v} \lor B_1 \\
A_3 \lor y_{u,v} \lor z_{u,v} & \lor \overline{z}_{u,v} \lor B_1 \\
A_4 \lor y_{u,v} \lor z_{u,v} & \lor \overline{z}_{u,v} \lor \overline{B}_2 ,
\end{align*}
\]

where \(A_1, A_2, A_3, A_4\) and \(B_1, B_2\) are the result of applying the substitution to \(A\) and \(B\) respectively. These clauses may contain repetitions: if \(A\) does not contain negative literals then \(A_1, \ldots, A_4\) are all the same. If \(A\) contains one negative literal then we get two clauses repeated twice each. If \(A\) contains two negative literals then they are all different. Similarly for \(B\): if it contains no negative literals then \(B_1\) is equal to \(B_2\), otherwise it contains one negative literal and \(B_1\) is different from \(B_2\). \(B\) cannot contain two negative literals.

By resolving on both variables \(y_{u,v}\) and \(z_{u,v}\) we obtain clauses \(A_\mu \lor \overline{B}_\nu\) for \(\mu \in \{1, 2, 3, 4\}\) and \(\nu \in \{1, 2\}\). We can exclude the possibility that \(A\) contains two negative literals and simultaneously \(B\) contains one, because otherwise \(A \lor B\) would have three negative literals. Therefore, the set of newly derived clauses has size at most four and is indeed the sequence of clauses obtained by applying the substitution to \(A \lor B\). This concludes the induction and gives us a refutation of GOP\(^{OR}\)(\(G\)) since \(C_r\) is the empty clause and, therefore, \(C_r\) is the set containing only the empty clause.

Lemma 10. There exists a sequence of graphs \(\{G_n\}_n\) such that each \(G_n\) has \(\Theta(n)\) vertices and constant degree \(d\), and any PC refutation of GOP\(^{OR}\)(\(G_n\)) without negative literals requires monomial size \(2^\Omega(n)\).

Proof. Let \(\{G_n\}_n\) be the sequence of graphs given by Proposition 8. Let \(P\) be a refutation of GOP\(^{OR}\)(\(G_n\)) in monomial size \(s\). By Lemma 6, there is a restriction \(\rho\) that sets exactly one of \(\{y_{u,v}, \overline{z}_{u,v}\}\) to \(0\) and is such that all monomials in \(P\) have degree at most \(s\). Note that the formula GOP\(^{OR}\)(\(G_n\))\(\rho\) is an isomorphic copy GOP\((G_n)\), where each variable \(x_{u,v}\) has been renamed either to \(y_{u,v}\) or to \(\overline{z}_{u,v}\), and thus, by Proposition 8, it requires refutations of degree \(\Omega(n)\). Since \(P\)\(\rho\) is a PC refutation of GOP\(^{OR}\)(\(G_n\))\(\rho\), we conclude that \(\log s \geq \Omega(n)\) and the lemma follows.

We collect the two lemmas in the following theorem.
Theorem 11. There is a family of constant width CNF formulas \( \{ F_n \}_{n \in \mathbb{N}} \) of size \( \Theta(n^3) \) such that \( F_n \) has a resolution refutation of length \( \Theta(n^3) \), but any PC refutation of \( F_n \) with no negative literals must contain \( 2^{\Omega(n)} \) monomials.

Remark 12. It is legitimate to ask whether the result holds when we reverse the encoding of true and false and adopt the classic standard for PC literature, where 0 is true and 1 is false. In this case, \( \text{GOP}^\text{OR}(G_n^\text{PC}) \) becomes easy for PC, but nevertheless we can get the same separation by simply flipping the polarity of all literals in \( \text{GOP}^\text{OR}(G_n) \), i.e., by substituting each \( x_{u,v} \) with \( \overline{y}_{u,v} \lor \overline{z}_{u,v} \) instead of \( y_{u,v} \lor z_{u,v} \), and then changing the random restriction to assign to true the variable chosen from each pair. Since in this case true is 0, monomials of large degree will be set to zero with overwhelmingly high probability.

We end this section by presenting a family of formulas that have small size, small space refutations in resolution – and, therefore, also in PC with negative literals – but exhibit a strong size-space trade-off for PC without negative literals. To define the space of a refutation, we think of it as a proof being presented on a blackboard. At each step we can either write down an axiom of the formula being refuted or a new clause obtained by one of the derivation rules of the proof system applied to what is already on the blackboard, or we can erase a line from the blackboard. The resolution space of the refutation is then the maximum number of clauses on the blackboard at any given moment, and the PC space of the refutation is the maximum number of monomials on the blackboard at any given moment.

Theorem 13. There exists a family of constant-width CNF formulas \( \{ F_n \}_{n \in \mathbb{N}} \) of size \( \Theta(n) \) such that:
1. there is a resolution refutation of \( F_n \) in size \( O(n) \) and space \( O(1) \); but
2. any PC refutation without negative literals of \( F_n \) in monomial size \( t \) and space \( s \) must satisfy \( s \log t = \Omega(n/\log n) \).

The CNF formulas we consider are lifted pebbling formulas as defined next. Let \( G = (V,E) \) be a DAG. If \( (u,v) \in E \) we say that \( u \) is a predecessor of \( v \) and \( v \) a successor of \( u \). We write \( \text{pred}(v) \) to denote the set of all predecessors of \( v \). A vertex with no predecessor (resp. successor) is called a source (resp. sink).

The pebbling formula [9] over a DAG \( G = (V,E) \) with a single sink \( z \), denoted \( \text{Peb}_G \), consists of the clauses \( x_u \lor \bigvee_{v \in \text{pred}(u)} \neg x_u \) for all \( v \in V \) (note that if \( v \) is a source, then \( \text{pred}(v) = \emptyset \) encoding that sources are true and truth propagates upwards, and the clause \( \neg x_v \) encoding that the sink is false). We encode this formula by a set of polynomials in the standard way. Given a set \( U \subseteq V \), we denote by \( x_U \) the monomial \( \prod_{u \in U} x_u \) (in particular, \( x_\emptyset = 1 \)). For every vertex \( v \in V \), we have the polynomial equation
\[ x_{\text{pred}(v)} \cdot (1 - x_v) = 0, \tag{15} \]
and for the sink \( z \) we also have the polynomial equation
\[ x_z = 0. \tag{16} \]
The formulas that witness the trade-off separation of Theorem 13 are based on the family of graphs defined by Gilbert and Tarjan [24]. These graphs have large pebbling cost \( \Omega(n/\log n) \), even in the stronger, so-called black-white pebbling model and were used in [8] to obtain a space-degree trade-off for PC.

Lemma 14 ([24, 8]). There is a family of graphs \( \{ G_n \}_{n \in \mathbb{N}} \) with indegree 2 of size \( \Theta(n) \) such that any PC refutation, even with negative literals, of \( \text{Peb}_{G_n} \) in space \( s \) and degree \( d \) must satisfy \( sd = \Omega(n/\log n) \).
With this result, we are now ready to prove Theorem 13.

Proof of Theorem 13. Let \( \{G_n\}_{n \in \mathbb{N}} \) be the family of graphs given by Lemma 14 and let \( N \) be the number of vertices of \( G_n \). Let \( x_1, \ldots, x_N \) be the variables of \( \text{Peb}_{G_n} \) in inverse topological order. We define \( F_n = \text{Peb}_{G_n}^\alpha \), that is, we substitute each variable \( x_i \) by \( \neg(y_i \lor z_i) \) and rewrite the formula in CNF.

The linear size resolution refutation of \( \text{Peb}_{G_n}^\alpha \) in space \( O(1) \) can be described in rounds. We start with the clause \( y_1 \lor z_1 \). At the end of round \( i \), we will have derived a clause \( \bigvee_{j \in S_i}(y_j \lor z_j) \) for some set \( S_i \subseteq [i] \) such that \( S_i \) forms a cut in \( G_n \), that is, the sink of \( G_n \) and the sources of \( G_n \) are not connected in \( G_n \setminus S_i \); and moreover every vertex in \( S_i \) has at least one predecessor not in \( S_i \). Furthermore, at each round, the cut \( S_i \) moves towards the sources, i.e., the set of vertices connected to the sink in \( G_n \setminus S_i \) increases when \( i \) increases.

For round \( i+1 \), we first weaken \( \bigvee_{j \in S_i}(y_j \lor z_j) \) to \( \bigvee_{j \in S_i \cup \{i+1\}}(y_j \lor z_j) \). Now, for all \( v \in S_i \) such that both predecessors of \( v \), say \( u \) and \( w \), are in \( S_i \cup \{i+1\} \) we resolve \( \bigvee_{j \in S_i \cup \{i+1\}}(y_j \lor z_j) \) with \( y_u \lor z_u \lor y_w \lor z_w \lor \bar{y}_v \) and then with \( y_u \lor z_u \lor y_w \lor z_w \lor \bar{z}_v \), thus obtaining a clause \( \bigvee_{j \in S_{i+1}}(y_j \lor z_j) \) for some set \( S_{i+1} \) that satisfies the invariant. Finally, after round \( N \), we have derived \( \bigvee_{j \in S_N}(y_j \lor z_j) \) where \( S_N \) only contains sources. Thus, we can easily derive contraction by resolving this with \( \bar{y}_j \) and \( \bar{z}_j \) for all \( j \in S_N \). Note that this refutation has space 3 and size \( O(N) = O(n) \).

Now for proving item 2, let \( \mathcal{P} \) be a PC refutation without negative literals of \( \text{Peb}_{G_n}^\alpha \) in monomial size \( t \) and space \( s \). By Lemma 6, there is a restriction \( \rho \) that for all \( i \in [N] \) sets exactly one of \( \{y_i, z_i\} \) to 0 and such that all monomials in \( \mathcal{P} \) when restricted by \( \rho \) have degree at most \( \log t \). Since space does not increase with restriction, we have that \( \mathcal{P}|_\rho \) is a refutation of \( \text{Peb}_{G_n}^{\rho, \alpha} \) in space at most \( O(s) \) and degree at most \( \log t \).

We now argue that there is a PC refutation with negative literals of \( \text{Peb}_{G_n}^{\rho, \alpha} \) in space \( O(s) \) and degree \( O(\log t) \) and, by Lemma 14, this will imply that \( s \log t = \Omega(n/\log n) \). Let \( H \) be the formula \( \text{Peb}_{G_n}^{\rho, \alpha} \) with any \( y_i \) substituted by \( (1 - \bar{y}_i) \) and any \( z_i \) by \( (1 - \bar{z}_i) \). Since \( H \) is an isomorphic copy of \( \text{Peb}_{G_n} \), where each variable \( x_i \) has been substituted by either \( \bar{y}_i \) or \( \bar{z}_i \), it is enough to show that there is a PC refutation with negative literals of \( H \) in space \( O(s) \) and degree \( O(\log t) \). Indeed, this follows since we can derive each axiom of \( \text{Peb}_{G_n}^{\rho, \alpha} \) from an axiom of \( H \) and variable axioms in constant space and degree.

4 Negative Literals and Semialgebraic Proofs

We show that allowing negative literals makes Sherali-Adams and sums-of-squares exponentially stronger, too. The main result of this section is that there is a family of formulas that have short resolution refutations but require exponential size \( \text{SA} \) and \( \text{SOS} \) refutations without negative literals. This implies, in both systems, an exponential separation between the power of proofs with and without negative literals.

The following auxiliary lemma states the well-known semantic completeness of \( \text{SA} \).

Lemma 15 (Folklore). If some multilinear inequalities \( S = \{r_1 \geq 0, \ldots, r_t \geq 0\} \) on variables \( \bar{x} = (x_1, \ldots, x_n) \) semantically imply a multilinear inequality \( r \geq 0 \) then there is an \( \text{SA} \) derivation of \( r \) from \( S \) in degree \( 2n \) and size \( 2^{\Theta(n)} \).

Proof. For a multilinear polynomial \( p \), we define the sets \( S^-_p := \{a \in \{0,1\}^n \mid p(a) < 0\} \) and \( S^+_p := \{a \in \{0,1\}^n \mid p(a) \geq 0\} \). The fact that inequality \( r \geq 0 \) is semantically implied by \( S \) means that \( S^-_r \subseteq \bigcup_i S^-_{r_i} \).
Let \( Q_i := S_{r_i}^- \setminus \bigcup_{j=1}^{i-1} S_{r_j}^- \). Consider the polynomial

\[
\sum_{i \in [\ell]} \left( \sum_{\alpha \in Q_i \cap \mathcal{P}_{r_i}^-} \frac{|r(\alpha)|}{|r_i(\alpha)|} r_i(\bar{x}) \chi_{\alpha}(\bar{x}) \right) + \sum_{\alpha \in \mathcal{P}_{r_i}^+} r(\alpha) \chi_{\alpha}(\bar{x}) ,
\]

where \( \chi_{\alpha}(\bar{x}) \) is the characteristic function of a point \( \alpha \). The polynomial (17) is pointwise equivalent to \( r \) on the boolean cube because of the definition of the characteristic functions. Moreover, (17) is a legal \( \text{SA} \) derivation from \( \mathcal{S} \) because \( \mathcal{S}_r^- \subseteq \bigcup \mathcal{S}_{r_j}^- \) implies that coefficients \( \frac{|r(\alpha)|}{|r_i(\alpha)|} \) in (17) are all positives.

The degree of the polynomial (17) is at most \( 2n \) by definition and size is at most \( 2^{3n} \). Since it is pointwise equivalent to \( r \) on the boolean it is enough to multilinearize to transform it into \( r \).

For multilinearization we apply the following procedure. Denote by \( h(\bar{x}) \) the polynomial (17) after expanding brackets. While polynomial \( h(\bar{x}) \) has a term of the form \( x_i^n t \) we subtract a polynomial \( (x_i^{d-i} - t) x_i^d \) from polynomial (17) where \( i \in [n] \) and \( d \geq 2 \) is an integer. In one step we reduce the individual degree of one variable in one term in the polynomial \( h(x) \) and increase the size of polynomial (17) by 2. At the end of the process (17) is a multilinear polynomial of degree at most \( 2n \) and size at most \( 2n^{3n} \), pointwise equivalent to \( r \). After expanding brackets it will be a multilinear polynomial that is pointwise equivalent to \( r \) on the boolean cube.

**Lemma 16.** Consider two sets \( \mathcal{S}_1 := \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_{\ell} \geq 0\} \) and \( \mathcal{S}_2 := \{f_1 = 0, \ldots, f_{m'} = 0; g_1 \geq 0, \ldots, g_{\ell'} \geq 0\} \). If there is an \( \text{SA} \) (resp. \( \text{SOS} \)) refutation of \( \mathcal{S}_2 \) in size \( N_2 \) and degree \( d_2 \) and each element \( f_1 \geq 0, -f_1 \geq 0, \) and \( g_1 \geq 0 \) can be derived in \( \text{SA} \) (resp. \( \text{SOS} \)) from \( \mathcal{S}_1 \) in size \( N_1 \) and degree \( d_1 \), then there is an \( \text{SA} \) (resp. \( \text{SOS} \)) refutation of \( \mathcal{S}_1 \) in size \( N_1 N_2 \) and degree \( d_1 d_2 \) (resp. in size \( N_1 N_2^{O(1)} \) and degree \( O(d_1 d_2) \)).

**Proof.** First consider a set \( \mathcal{S}_2 \) without equations (i.e., \( m' = 0 \)). Let \( \{h_1, \ldots, h_d\} \) be an \( \text{SA} \) (or \( \text{SOS} \)) refutation of \( \mathcal{S}_2 \) in size \( N_2 \) and degree \( d_2 \), so that we have

\[
\sum_{i \in [\ell']} h_i g_i = -1 .
\]

For \( i \in [\ell'] \), let \( \{q_{i,1}, \ldots, q_{i,m_i}; s_{1,i}, \ldots, s_{\ell,i}\} \) be an \( \text{SA} \) (or \( \text{SOS} \)) derivation of \( g_i \geq 0 \) from \( \mathcal{S}_1 \) in size \( N_1 \) and degree \( d_1 \), so that

\[
\sum_{j \in [m_i]} q_{i,j} p_j + \sum_{j \in [\ell]} s_{j,i} r_j = g_i .
\]

The composition of these derivations

\[
-1 = \sum_{i \in [\ell']} h_i g_i = \sum_{i \in [\ell']} h_i \left( \sum_{j \in [m_i]} q_{i,j} p_j + \sum_{j \in [\ell]} s_{j,i} r_j \right)
\]

\[
= \sum_{j \in [m_i]} \left( \sum_{i \in [\ell']} h_i q_{i,j} \right) p_j + \sum_{j \in [\ell]} \left( \sum_{i \in [\ell']} h_i s_{j,i} \right) r_j
\]

gives us the desired refutation of \( \mathcal{S}_1 \) in size \( N_1 N_2 \) and degree \( d_1 d_2 \). Notice that (21) is a valid \( \text{SA} \) (or \( \text{SOS} \)) refutation because polynomials \( h_i \) and \( s_{j,i} \) are valid multipliers for inequalities and thus so are their products and sums of products.
When $S_2$ contains equations, we reduce to the case where $m' = 0$, using the observation that the set
\[ S_2' := \{ -f_1 \geq 0, \ldots, -f_{m'} \geq 0; f_1 \geq 0, \ldots, f_{m'} \geq 0; g_1 \geq 0, \ldots, g_e \geq 0 \} \tag{22} \]
has an SA refutation of size $N_2$ and degree $d_2$ (or an SOS refutation of size $N_2^{O(1)}$ and degree $O(d_2)$). To see this, start from a refutation \{\(e_1, \ldots, e_m; h_1, \ldots, h_\ell\)\} of $S_2$ in size $N_2$ and degree $d_2$, so that we have
\[ \sum_{i \in [m']} e_i f_i + \sum_{i \in [\ell']} h_i g_i = -1. \tag{23} \]
To make it a valid SA refutation of $S_2'$, rewrite each $e_i f_i$ as $e_i^+ f_i + e_i^- (-f_i)$ where $e_i = e_i^+ - e_i^-$ and both $e^+$ and $e^-$ are positive sums of monomials. Note that this operation does not change neither size nor degree. To make it a valid SOS refutation of $S_2'$, rewrite each $e_i f_i$ as $(e_i^- + e_i^+)/2 \cdot f_i + (e_i^- - e_i^+)/2 \cdot (-f_i)$. Note that this refutation has degree at most $2d_2$ and size at most $N_2^2$. The result follows. \(\blacksquare\)

Recall the ordering principle formula $\text{OP}_n$, which is the graph ordering principle formula (12) over the complete graph $K_n$. As mentioned in Section 2, for SA and SOS the default encoding of CNF formulas is (6d). For $\text{OP}_n$ this encoding consists of inequalities:
\begin{align*}
(1 - x_{u,v}) + (1 - x_{v,w}) + x_{u,w} - 1 & \geq 0 \quad \text{for any three distinct } u, v, w \in [n], \tag{24a} \\
(1 - x_{u,v}) + (1 - x_{v,u}) - 1 & \geq 0 \quad \text{for any two distinct } u, v \in [n], \tag{24b} \\
\sum_{u \in [n]} x_{u,v} - 1 & \geq 0 \quad \text{for any } u, v \in [n]. \tag{24c}
\end{align*}
The reason we cannot use the graph ordering principle as we did in Section 3 is that we do not know how to prove strong SA degree lower bounds for GOP. Instead we use $\text{OP}_n$ which can be still encoded in low degree using inequalities, and for which we have degree lower bounds.

For the separation we use the $\text{OP}^{\text{OR}}_n$ formula. We have already showed in Lemma 9 that $\text{OP}^{\text{OR}}_n$ is easy for resolution. In the presence of negative literals, this transfers to SA by the following known simulation result.

\textbf{Lemma 17 ([5])}. If a CNF formula $F$ has a resolution refutation of width $w$ and length $L$, then it has an SA refutation with negative literals of degree $w+1$ and size $O(w^2L)$.

Since SOS can simulate SA we obtain the following upper bound.

\textbf{Lemma 18}. The formula $\text{OP}^{\text{OR}}_n$ has SA and SOS refutations with negative literals of size $n^{O(1)}$.

\textbf{Proof}. By Lemma 9 the formula $\text{OP}^{\text{OR}}_n$ has a resolution refutation of size $O(n^3)$. The width of any resolution refutation cannot exceed the number of variables that appear in the formula, hence the considered refutation has width at most $O(n^2)$. Together with Lemma 17, this implies the desired result for SA. To conclude the proof it is enough to recall that, by Proposition 5, SOS can simulate any SA proof with at most a polynomial blowup in size. \(\blacksquare\)

We now proceed to prove the lower bounds for SA and SOS without negative literals. The main idea is analogous to that of Lemma 10: we show that we can reduce any small SA or SOS refutation without negative literals of $\text{OP}^{\text{OR}}_n$ to a low degree refutation of $\text{OP}_n$. To conclude the proof we then apply the following degree lower bounds.
The Power of Negative Reasoning

Lemma 19 ([16]). Any SA refutation of $\text{OP}_n$ has degree at least $n - 2$.

For SOS the lower bound we know holds for the following, slightly different encoding:

\[
\begin{align*}
x_{u,v}x_{v,w}(1 - x_{u,w}) &= 0 & \text{for any three distinct } u, v, w & \in [n], \\
x_{u,v}x_{v,u} &= 0 & \text{for any two distinct } u, v & \in [n], \\
\sum_{u \in [n]} x_{u,v} &= 1 + z_v^2 & \text{for any } u, v & \in [n],
\end{align*}
\]

where $z_v$ are real valued extension variables.

Lemma 20 ([35]). For any $\varepsilon > 0$, there is a constant $c_\varepsilon > 0$ such that any SOS proof of the system of equations (25) has degree at least $c_\varepsilon n^{1/2 - \varepsilon}$.

We show that this result implies a degree lower bound for the standard encoding of $\text{OP}_n$ as in (24).

Corollary 21. For any $\varepsilon > 0$, there is a constant $c_\varepsilon > 0$ such that any SOS proof of the $\text{OP}_n$ has degree at least $c_\varepsilon n^{1/2 - \varepsilon}$.

Proof. For the sake of completeness, let us argue a well known fact. If $p = 0$ is the product encoding, as per (6b), of a clause $C$ of width $w$, and $r \geq 0$ is the additive encoding of $C$, as per (6d), then there is an SA (and hence also SOS) derivation of $r$ from $p$ and boolean axioms in degree $w + 1$. Indeed, this follows from Lemma 15 by noting that the product encoding $p = 0$ is equivalent to the two inequalities $p \geq 0$ and $-p \geq 0$ that semantically imply the inequality $r \geq 0$.

By using the above fact we can derive inequalities (24a) and (24b) from the constraints (25a) and (25b) in degree 4. Finally, the inequality

\[
\sum_{u \in [n]} x_{u,v} \geq 1 \geq 0
\]

(26)

can be derived in SOS from (25c) by adding the square $z_v^2$ and thus obtaining

\[
\left(\sum_{u \in [n]} x_{u,v} - 1 - z_v^2\right) + z_v^2 = \sum_{u \in [n]} x_{u,v} - 1.
\]

Therefore, if there an SOS refutation of (24) in degree $d$, then by Lemma 16 there is an SOS refutation of (25) in degree $O(d)$. Together with Lemma 20, this implies the desired lower bound.

We are now ready to prove the size lower bounds for SA and SOS.

Lemma 22. Any SA refutation of $\text{OP}_n^{\text{OR}}$ without negative literals requires monomial size $2^\Omega(n)$. For any $\varepsilon > 0$ there is a constant $c_\varepsilon > 0$ such that any SOS refutation of $\text{OP}_n^{\text{OR}}$ without negative literals requires monomial size $2^\varepsilon n^{1/2 - \varepsilon}$.

Proof. The proof is very similar to that of Lemma 10. Let $y_{u,v}, z_{u,v}$ for $u, v \in [n]$ be the variables of $\text{OP}_n^{\text{OR}}$, that is, $\text{OP}_n^{\text{OR}}$ is obtained by substituting in $\text{OP}_n$ each variable $x_{u,v}$ by $y_{u,v} + z_{u,v}$. Let $\{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_f \geq 0\}$ is the encoding of $\text{OP}_n^{\text{OR}}$ and let $\{q_1, \ldots, q_m; s_1, \ldots, s_f\}$ be an SA refutation of $\text{OP}_n^{\text{OR}}$ without negative literals, so that

\[
\sum_{j \in [n]} q_j p_j + \sum_{j \in [f]} s_j r_j = -1.
\]
Let $S$ be the monomial size of this refutation. By Lemma 6, there is a restriction $\rho$ that sets exactly one of $\{y_{u,v}, z_{u,v}\}$ to 0 and is such that all monomials appearing in (28) when restricted by $\rho$ have degree at most $\log S$. Note that the formula $\text{OP}_n^\text{OR,\rho}$ is an isomorphic copy $\text{OP}_n$, where each variable $x_{u,v}$ has been renamed either to $y_{u,v}$ or to $z_{u,v}$, and thus, by Lemma 19, it requires refutation of degree $O(n)$. Since $\mathcal{P}|_\rho$ is an SA refutation of $\text{OP}_n^\text{OR,\rho}$ in degree at most $\log S$, we conclude that $\log S \geq \Omega(n)$ and the size lower bound for SA follows.

The proof of the size lower bound for SOS is analogous, except that we use Corollary 21 for the degree lower bound instead of Lemma 19.

We collect Lemmas 9,18 and 22 in the following theorem.

**Theorem 23.** There is a family of CNF formulas $\{F_n\}_n$ of size $\Theta(n^3)$ such that $F_n$ has a resolution refutation and SA and SOS refutations with negative literals in monomial size $n^{O(1)}$. But any SA refutation of $F_n$ without negative literals requires monomial size $2^{\Omega(n)}$, and for any $\varepsilon > 0$ there is a constant $c_\varepsilon > 0$ such that any SOS refutation of $F_n$ without negative literals requires monomial size $2^{c_\varepsilon n^{1/2-\varepsilon}}$.

## 5 Pigeonhole and Sherali-Adams

In this section we improve the previous result for Sherali-Adams and show a separation between SA with and without negative literals, using constant width formulas, and hence independent of the encoding of the clauses. Note that, in contrast to the previous section, this result does not give a corresponding separation for SOS.

We start with the formula that encodes the (negation of the) pigeonhole principle (PHP). The formula is defined on propositional variables $x_{i,j}$ for $i \in [n+1]$ and $j \in [n]$, with the intended meaning that $x_{i,j}$ is true if and only if the $i$-th pigeon goes into hole $j$. The clauses of PHP are:

$$P_i := \bigvee_{j \in [n]} x_{i,j} \quad \text{for every } i \in [n+1],$$

$$H_{i,j} := \overline{x}_{i,j} \vee \overline{\overline{x}}_{i,j} \quad \text{for every two distinct } i,k \in [n+1] \text{ and every } j \in [n].$$

In order to reduce the width of the formula we introduce extension variables $e_{i,j}$ for $i \in [n+1]$ and $j \in [n]$ and replace the clauses (29a) by

$$\text{EP}_{i,j} := e_{i,j-1} \vee x_{i,j} \vee \overline{x}_{i,j} \quad \text{for every } i \in [n+1] \text{ and } j \in [n],$$

$$\text{EP}_{i,0} := e_{i,0}, \quad \text{EP}_{i,n+1} := e_{i,n} \quad \text{for every } i \in [n+1].$$

Intuitively, the variable $e_{i,j}$ represents the disjunction of the variables $x_{i,\ell}$ for $\ell \leq j$. We denote this 3-CNF formula with extension variables by $\text{EPHP}$.

Similarly to previous cases, we substitute the variables in the formula by a 2-bit function. In this case, however, we use $\text{NOR}(y, z) := \neg(y \vee z)$ which is equivalent to $\overline{y} \land \overline{z}$. We apply this substitution to the formula $\text{EPHP}$, to obtain the formula $\text{EPHP}^\text{NOR}$, by replacing each variable $x_{i,j}$ with $\overline{x}_{i,j} \land \overline{\overline{x}}_{i,j}$ and each $e_{i,j}$ with $\overline{x}_{i,j} \land \overline{e}_{i,j}$ and rewriting it in CNF.

It was shown in [16] that Sherali-Adams without negative literals can refute PHP in polynomial size. We use this result to obtain a size upper bound for Sherali-Adams refutations with negative literals of $\text{EPHP}^\text{NOR}$.

**Lemma 24 ([16]).** There is an SA refutation without negative literals of PHP of size $O(n^2)$.

**Lemma 25.** There is an SA refutation with negative literals of $\text{EPHP}^\text{NOR}$ of size $O(n^5)$.
The Power of Negative Reasoning

**Proof.** Let \( S \) be the set of polynomial inequalities encoding PHP as per (6d) plus the variable axioms for each \( x_{i,j} \) and let \( S' = \{ p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0 \} \) be the set of polynomial inequalities obtained from \( S \) by replacing each variable \( x_{i,j} \) by the product \( \overline{y}_{i,j} \overline{z}_{i,j} \).

We want to show that there is a small, namely size \( O(n) \), SA derivation with negative literals from \( \text{EPHP}^{\text{NOR}} \) of each of the inequalities \( p_i \geq 0, \neg p_i \geq 0 \) for \( i \in [m] \) and \( r_i \geq 0 \) for \( i \in [\ell] \). Suppose this is true. Then given a size \( O(n^4) \) refutation of PHP, which is guaranteed to exist by Lemma 24, we can replace each occurrence of \( x_{i,j} \) by the product \( \overline{y}_{i,j} \overline{z}_{i,j} \) and obtain a refutation of \( S' \) of exactly the same size. Composing this refutation with the derivation of \( S' \) from \( \text{EPHP}^{\text{NOR}} \), we obtain, by Lemma 16, a refutation of \( \text{EPHP}^{\text{NOR}} \) of size \( O(n^5) \).

We start by considering the hole axioms (29b) of PHP, that is, \( \overline{x}_{i,j} \lor \overline{y}_{i,j} \), which is encoded as \( (1 - x_{i,j}) + (1 - x_{k,j}) - 1 \geq 0 \). After replacing the \( x \) variables in the polynomial inequality, we obtain

\[
(1 - \overline{y}_{i,j} \overline{z}_{i,j}) + (1 - \overline{y}_{k,j} \overline{z}_{k,j}) - 1 \geq 0 , \tag{31}
\]

which is in \( S' \). Now, the formula \( \text{EPHP} \) also contains hole axioms (29b), and thus the substituted formula \( \text{EPHP}^{\text{NOR}} \) contains a set of inequalities encoding the formula \( \overline{y}_{i,j} \overline{z}_{i,j} \lor \overline{y}_{k,j} \overline{z}_{k,j} \). Since this formula, and therefore also the inequalities encoding it, semantically implies inequality (31), by Lemma 15 there is an SA derivation of (31) from \( \text{EPHP}^{\text{NOR}} \) in constant size.

We have a similar situation for the pigeon axioms (29a) of PHP, i.e., \( \lor_{j \in [n]} x_{i,j} \), which is encoded as \( \sum_{j=1}^{n} x_{i,j} - 1 \geq 0 \). Our goal is to derive the polynomial inequality

\[
\sum_{j=1}^{n} \overline{y}_{i,j} \overline{z}_{i,j} - 1 \geq 0 \tag{32}
\]

from \( \text{EPHP}^{\text{NOR}} \). Again, by Lemma 15, each of the inequalities

\[
\begin{align*}
&\sum_{j=1}^{n} (1 - \overline{y}_{i,j} \overline{z}_{i,j}) - 1 \geq 0 ; \\
&\sum_{j=1}^{n} (1 - \overline{y}_{i,j} \overline{z}_{i,j}) - 1 \geq 0 ,
\end{align*}
\]

has an SA derivation from \( \text{EPHP}^{\text{NOR}} \) of the constant size, since they are semantically implied by the clauses \( \text{EP}_{i,j} \) with variables substitute by \( \text{NOR} \). Note that the sum of these inequalities

\[
(1 - \overline{y}_{i,0} \overline{z}_{i,0}) + \sum_{j=1}^{n} (1 - \overline{y}_{i,j} \overline{z}_{i,j}) - 1 + \overline{y}_{i,n} \overline{z}_{i,n} - 1 = \sum_{j=1}^{n} \overline{y}_{i,j} \overline{z}_{i,j} - 1 \tag{33}
\]

is a valid SA derivation of (32) in size \( O(n) \).

Finally, we note that the substituted variable axioms \( \overline{y}_{i,j} \overline{z}_{i,j} \geq 0, 1 - \overline{y}_{i,j} \overline{z}_{i,j} \geq 0, (\overline{y}_{i,j} \overline{z}_{i,j})^2 - \overline{y}_{i,j} \overline{z}_{i,j} \geq 0 \) and \(- (\overline{y}_{i,j} \overline{z}_{i,j})^2 + \overline{y}_{i,j} \overline{z}_{i,j} \geq 0 \) can be easily derived in constant size from the variable axioms for \( y_{i,j} \) and \( z_{i,j} \).

We now show that any Sherali-Adams refutation of \( \text{EPHP}^{\text{NOR}} \) without negative literals has exponential size. For this, we use the following degree lower bound.

\begin{itemize}
  \item **Lemma 26 ([5])**. Any SA refutation of \( \text{EPHP} \) has a degree at least \( n - 2 \).
  \item **Lemma 27**. Any SA refutation of \( \text{EPHP}^{\text{NOR}} \) without negative literals requires monomial size \( 2^{\Omega(n)} \).
\end{itemize}
Proof. The proof is very similar to that of Lemma 22. Consider an SA refutation of $\text{EPHP}^{\text{NOR}}$ without negative literals, that is, a set of polynomials $P = \{q_1, \ldots, q_m; s_1, \ldots, s_\ell\}$ such that
\begin{equation}
\sum_{j \in [m]} q_j p_j + \sum_{j \in \ell} s_j r_j = -1 ,
\end{equation}
where $\{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ is the polynomial encoding of $\text{EPHP}^{\text{NOR}}$ and each $s_j$ is a positive linear combination of generalized monomials. Let $S$ be the monomial size of this refutation. By Lemma 6, there is a restriction $\rho$ that sets exactly one of $\{y_{i,j}, z_{i,j}\}$ and exactly one of $\{a_{i,j}, b_{i,j}\}$ to 0 and is such that all monomials appearing in (34) when restricted by $\rho$ have degree at most $\log S$. Note that the formula $\text{EPHP}^{\text{NOR}}|_\rho$ is almost an isomorphic copy of $\text{EPHP}$, except that:
- each variable $x_{i,j}$ has been substituted by either $(1 - y_{i,j})$ or by $(1 - z_{i,j})$;
- each variable $e_{i,j}$ has been substituted by either $(1 - a_{i,j})$ or by $(1 - b_{i,j})$.
It is not hard to see that this formula $\text{EPHP}^{\text{NOR}}|_\rho$ also requires degree $n - 2$ to be refuted in SA, since otherwise we could obtain, by substituting each variable $y_{i,j}$ and $z_{i,j}$ by $(1 - x_{i,j})$ and each variable $a_{i,j}$ and $b_{i,j}$ by $(1 - e_{i,j})$, a refutation of $\text{EPHP}$ in degree less than $n - 2$ contradicting Lemma 19. Therefore, since $P|_\rho$ is an SA refutation of $\text{EPHP}^{\text{NOR}}|_\rho$ in degree at most $\log S$, we conclude that $\log S \geq \Omega(n)$ and the size lower bound for SA follows. ▶

We collect Lemmas 25 and 27 in the following theorem.

Theorem 28. There is a family of constant width CNF formulas $\{F_n\}_n$ of size $\Theta(n^3)$ such that $F_n$ has an SA refutation with negative literals of monomial size $O(n^5)$, but any SA refutation of $F_n$ without negative literals must contain $2^{\Omega(n)}$ monomials.

6 Separating Nullstellensatz with and without negative literals

In this section we show that there are formulas that have linear size tree-like resolution refutations—and, therefore, also linear size Nullstellensatz refutations if variables for negative literals are allowed—but require nearly exponential size Nullstellensatz refutations if such variables are not allowed.

Theorem 29. There exists a family of constant width CNF formulas $\{F_n\}_{n \in \mathbb{N}}$ of size $\Theta(n)$ such that there are tree-like resolution refutations, and therefore also NS refutations with negative literals, of $F_n$ in size $O(n)$, but any NS refutation without negative literals of $F_n$ must have size $2^{\Omega(n/\log n)}$.

A formula that witnesses a size separation of $2^{\tilde{\Omega}(n)}$ must necessarily require NS degree $\tilde{\Omega}(n)$ since if there is a degree-$d$ NS refutation, then there is an NS refutation without negative literals in simultaneous degree $d$ and size $n^{O(d)}$. In this sense, the separation in Theorem 29 is nearly optimal. For smaller values of $d$, we can show a similar separation with the additional property that NS with negative literals presents a smooth trade-off between degree and size of refutations.

Theorem 30. For any $0 < \epsilon \leq 1/4$, any large enough $n \in \mathbb{N}$ and any $2 \leq k \leq n^{\epsilon/2}$, there exists a constant width CNF formula $F_{k,n}$ of size $\Theta(kn)$ such that:
1. there is an NS refutation with negative literals of $F_{k,n}$ in linear size $O(kn)$;
2. for any $d$ satisfying $2^{1+1/k}k^4 \log n \leq d \leq \sqrt{n}$, there is an NS refutation with negative literals of $F_{k,n}$ in degree $d$ and size $n^{k(1+5\epsilon)/d^{2k-3}}$; and
3. any NS refutation without negative literals of $F_{k,n}$ must have size $2^k$. 
The CNF formulas we consider are lifted pebbling formulas. We will also use the relation between the formulas and pebble games as defined next.

The reversible pebble game [10] is a single-player game that is played with a set of pebbles on a DAG \( G \). The goal of the game is to pebble (i.e., place a pebble on) each vertex of \( G \) at least once. Initially, the graph contains no pebbles. At each round, the player is allowed to place a pebble on any vertex of \( G \) such that all its predecessors are pebbled. In particular, the player is always allowed to place a pebble on any source of \( G \). Moreover, at any given round, a pebble on a vertex \( v \) can be removed from \( G \) if all the predecessors of \( v \) are pebbled. Again, this implies that it is always possible to remove a pebble from a source of \( G \). A sequence of pebbling moves that pebbles each vertex of \( G \) at least once according to these rules and ends with the empty graph is called a reversible pebbling of \( G \). The time of a reversible pebbling is the number of rounds and the space is the maximum number of pebbles on \( G \) at any given moment. The reversible pebbling cost of \( G \) is the minimum space required for any reversible pebbling of \( G \) (independent of time). We sometime refer to the standard pebble game where the rule for removing pebbles is relaxed so that any pebble can be removed at any point.

For our purpose, we note that pebbling formulas always have linear size NS refutations (even without negative literals), while for some “hard” graphs the NS degree is necessarily large. In order to prove the separations in this section, we use the following characterization of NS degree and size, when negative literals are not allowed, in terms of reversible pebbling space and time [19]. We would like to point out that for Theorem 29 the degree characterization of [18], or even the not-so-tight bound of [12], would have be enough.

▶ Lemma 31 ([19]). Let \( G \) be a single-sink DAG. There is a Nullstellensatz degree \( d \) and size \( t \) refutation without negative literals of \( \text{Peb}_G \) if and only if there is a reversible pebbling of \( G \) in space \( d \) and time \( t − 1 \).

By this characterisation, it is easy to see that pebbling formulas always have linear size NS refutations without negative literals. In order to obtain NS size lower bounds when negative literals are not allowed we compose pebbling formulas with the not-or function NOR, that is, we substitute each variable \( x_i \) by \( \neg(y_i \lor z_i) \). This is useful for proving NS lower bounds since formulas lifted with NOR satisfy the following property.

▶ Lemma 32. Let \( F \) be an unsatisfiable CNF formula. If NS requires degree \( d \) to refute \( F \), then NS without negative literals requires size \( 2^d \) to refute \( F^{\text{NOR}} \).

Proof. Let \( n \) be the number of variables of \( F \), and let \( y_1, \ldots, y_n \) and \( z_1, \ldots, z_n \) be the variables of \( F^{\text{NOR}} \). Let \( S = \{ p_1 = 0, \ldots, p_m = 0 \} \) be the set of polynomial equations encoding \( F^{\text{NOR}} \) (plus the variable axioms). Let \( \{ q_1, \ldots, q_m \} \) be an NS refutation without negative literals of \( S \), that is,

\[
\sum_{j \in [m]} q_j p_j = 1 ,
\]

and let \( s \) be its monomial size. By Lemma 6, there is a restriction \( \rho \) that for all \( i \in [n] \) sets exactly one of \( \{ y_i, z_i \} \) to 0 and such that all monomials in \( q_j p_j \) for \( p_j = 0 \in S \) when restricted by \( \rho \) have degree less than \( \log s \). Note that \( F^{\text{NOR}}|_{\rho} \) is almost an isomorphic copy of \( F \), except that variables \( x_i \) have been substituted by either \( (1−y_i) \) or \( (1−z_i) \). It is not hard to see that \( F^{\text{NOR}}|_{\rho} \) also requires degree \( d \) refutations, since otherwise substituting every \( y_i \) or \( z_i \) appearing in the refutation by \((1−x_i)\) would give a refutation of \( F \) in degree less than \( d \). This implies that \( \log s \geq d \).
While substituting variables in a formula with \( \text{NOR} \) can give \( \text{NS} \) size lower bounds if negative literals are not allowed, for pebbling formulas this substitution does not make the formula harder for \( \text{NS} \) if negative literals are allowed, and not even for tree-like resolution.

**Lemma 33.** Let \( G \) be a DAG with \( n \) vertices. There is a tree-like resolution refutation of \( \text{Peb}_G^{\text{NOR}} \) in size \( 4n + 1 \).

**Proof.** We describe a decision tree that solves the falsified clause search problem of \( \text{Peb}_G^{\text{NOR}} \). The idea is to query the variables in topological order, from the sources to the sink. Let \( x_1, \ldots, x_n \) be the variables of \( \text{Peb}_G \), ordered topologically according to \( G \) from the sources to the sink, and for \( i \in [n] \), let \( y_i, z_i \) be the lifted variables so that \( x_i = \neg(y_i \lor z_i) \). The decision tree queries \( y_i \) and \( z_i \), from \( i = 1 \) to \( n \): if the result of the query is 0 it proceeds to the next query, if it is 1 it has found a falsified axiom (since this implies there is a false variable whose predecessors are true). Finally, if all vertices are 0, then the sink clause of \( \text{Peb}_G^{\text{NOR}} \), which states the sink is false, is falsified. This gives a decision tree of size \( 4n + 1 \) (and depth \( 2n \)).

We also observe that if a CNF formula has small \( \text{NS} \) refutations without negative literals in degree \( d \), then the formula composed with \( \text{NOR} \) has small \( \text{NS} \) refutations with negative literals in degree \( 2d \).

**Lemma 34.** Let \( F \) be a constant-width unsatisfiable CNF formula. If there is an \( \text{NS} \) refutation without negative literals of \( F \) in size \( s \) and degree \( d \) then there is an \( \text{NS} \) refutation with negative literals of \( F^{\text{NOR}} \) in size \( O(s) \) and degree \( 2d \).

**Proof.** Let \( x_1, \ldots, x_n \) be the variables of \( F \), and for \( i \in [n] \), let \( y_i, z_i \) be the lifted variables so that \( x_i = \neg(y_i \lor z_i) \). For a clause (or a CNF) \( C \), let \( C^* \) be the polynomial translation of \( C \) without negative literals, as per (6b). Moreover, for a polynomial \( p \) over \( x \) variables, let \( p[yz] \) be the polynomial obtained by substituting in \( p \) each variable \( x_i \) by the product \( y_i z_i \).

Consider a clause \( C \) of \( F \) and denote by \( C^{\text{NOR}} \) the CNF that is obtained by substituting variables \( x_i \) by \( \neg(y_i \lor z_i) \). Since \( C \) has constant width, there is an \( \text{NS} \) derivation (with negative literals) of \( C^*[yz] \) from the set of polynomials \( (C^{\text{NOR}})^* \) in constant size and without increasing the degree.

Let \( S = \{ p_1 = 0, \ldots, p_m = 0 \} \) be the set of polynomial equations encoding \( F \) (plus the variable axioms). Let \( \{ q_1, \ldots, q_m \} \) be an \( \text{NS} \) refutation without negative literals of \( S \), that is,

\[
\sum_{j \in [m]} q_j p_j = 1 ,
\]

in degree \( d \) and monomial size \( s \). If we substitute every variable \( x_i \) in \( \sum_{j \in [m]} q_j p_j \) by \( y_i z_i \), we have a polynomial that is syntactically equal to 1, has degree \( 2d \) and has monomials size \( s \). The lemma follows by the observation above that implies that there is an \( \text{NS} \) derivation (with negative literals) of \( p[yz] \) from \( F^{\text{NOR}} \) in constant size and without increasing the degree.

The family of graphs we consider for the proof of Theorem 29 is the one defined by Paul et al. [34] and also used by Gilbert and Tarjan [24]. It was shown in [34] that these graphs have large standard pebbling cost, and thus also have large reversible pebbling cost.

**Theorem 35 ([34]).** For every \( N \in \mathbb{N} \), there is a DAG of size \( \Theta(N) \) that has reversible pebbling cost \( \Omega(N / \log N) \).
We are now ready to prove the first theorem of this section.

**Proof of Theorem 29.** Let $G_N$ be the DAG of size $\Theta(N)$ and pebbling cost $\Omega(N/\log N)$ given by Theorem 35. We define $F_N = \text{Peb}_{\text{G}_N}^{\text{NOR}}$. The upper bound follows directly from Lemma 33. For the lower bound, note that Theorem 35 together with Lemma 31 imply that $\text{NS}$ requires degree $\Omega(N/\log N)$ to refute $\text{Peb}_{\text{G}_N}$ and, therefore, by Lemma 32, any $\text{NS}$ refutation of $F_N$ must have size $2^{\Omega(N/\log N)}$. □

To prove Theorem 30 we consider another family of graphs, based on the so-called **bit-reversal permutation graphs**. Let $n$ be an integer. Given $j \in \{0, 1, \ldots, 2^n - 1\}$ and $\ell \in [n]$, we denote by $j_{\ell}$ the $\ell$th bit of $j$. Now let $\text{reverse}(j) = \sum_{\ell \in [n]} 2^n - \ell _{\ell}$ be the integer in $\{0, 1, \ldots, 2^n - 1\}$ obtained by reversing the bit representation of $j$.

The $k$-layer $n$-bit-reversal permutation graph consists of $k$ directed path graphs of length $2^n$, where we consider vertices in each path to be numbered from 0 to $2^n - 1$, and in between consecutive layers $i$ and $i + 1$, for $i \in [k - 1]$, there are edges from vertex $j$ in layer $i$ to vertex $\text{reverse}(j)$ in layer $i + 1$, for all $j \in \{0, 1, \ldots, 2^n - 1\}$. See Figure 1 for an illustration.

It was shown in [3] that these graphs exhibit a certain smooth time-space trade-off for standard pebbling.

**Proposition 36 ([3]).** Let $G$ be a $k$-layer $n$-bit-reversal permutation graph, and let $N = 2^n$. For any $s$ such that $k + 1 \leq s \leq \sqrt{N}/4$ there exists a standard pebbling of $G$ in space $2k^2s + 2$ and time $2^{k/2}(N^k/s^{2k-3})$. Furthermore, every standard pebbling of $G$ in space $s$ requires time $2^{-3k}(N^k/s^{2k-3})$.

By a classical result of [10], which is analysed precisely in [32], we can translate, with some loss both in time and in space, the upper bound in this trade-off to the reversible pebble game.

**Proposition 37 ([10, 32]).** Let $G$ be an arbitrary DAG. If $G$ has a standard pebbling in space $s$ and time $t \geq 2s$, then for any $\epsilon > 0$, $G$ can be reversibly pebbled in simultaneous time $t^{1+\epsilon}/s^{1}$ and space $\epsilon(2^{1/\epsilon} - 1)s \log(t/s)$.

**Corollary 38.** Let $0 < \epsilon \leq 1/4$, let $G$ be a $k$-layer $n$-bit-reversal permutation graph and let $N = 2^n$. For any $s$ such that $k + 1 \leq s \leq \sqrt{N}/4$ there exists a reversible pebbling of $G$ in space $\frac{n}{k+1}2^{1/\epsilon}k4 \log N$ and time $2^k(N^{k(1+\epsilon)}/s^{2k-3})$.

We are now ready to prove Theorem 30.

**Proof of Theorem 30.** Let $0 < \epsilon \leq 1/4$, let $G$ be a $k$-layer $n$-bit-reversal permutation graph for $k < 2^{n/2}$, and let $N = 2^n$. We define $F_{k,N}$ to be $\text{Peb}_{\text{G}_N}^{\text{NOR}}$. Item 1 follows from Lemma 34 and the fact that any pebbling formula has linear size $\text{NS}$ refutations.
We argue that from Corollary 38 it follows that for any \( d \) such that \( 2^{1/\epsilon} k^4 \log N \leq d \leq \sqrt{N} \),
the graph \( G \) can be reversibly pebbled in space \( d \) and time \( n^{k(1+5\epsilon)/d^{2k-3}} \). Item 2 then
follows from the correspondence between reversible pebbling and NS refutations (Lemma 31) and Lemma 34. To see why the claim above holds, let

\[
s := \frac{d(k+1)}{2^{1/\epsilon} k^4 \log N},
\]

which is at least \( k + 1 \) and at most \( \sqrt{N}/4 \) by the bounds of \( d \). Note, moreover, that

\[
s \geq \frac{d}{2^{1/\epsilon} k^4 \log N} \geq \frac{2d}{N^{2\epsilon}},
\]

where the last inequality holds for \( N \) large enough since \( k \leq N^{\epsilon/2} \). By Corollary 38 it then
follows that there is a reversible pebbling of \( G \) in space \( d \) and time

\[
2^k \cdot \frac{N^{k(1+\epsilon)}}{2^{k-3}} \leq 2^k \cdot \frac{N^{2\epsilon}}{2} \cdot \frac{N^{k(1+\epsilon)}}{d^{2k-3}} \leq \frac{n^{k(1+5\epsilon)}}{d^{2k-3}},
\]

as claimed.

Item 3 follows by applying Lemma 32 with \( d = k \). ▶

7 Concluding Remarks

Algebraic and semi-algebraic proof systems become more powerful when they can succinctly
represent negation of variables using additional formal variables. In some cases this advantage
results in exponentially smaller proofs. To witness these separations we built rather artificial
formulas. It would be interesting to understand whether this phenomenon occurs for formulas
encoding natural problems as well.

More importantly, is this just a theoretical advantage? Practical approaches based on
the naive computation of a Gröbner basis nullify any additional expressive power. Since the
polynomials \( \pi_i = 1 - x_i \) are in the ideal, any such computation eliminates one variable in
each pair, potentially causing an exponential blow-up in size along the way. In algebraic
circuit verification this is a concrete problem. Some works indeed use new variables for
negated literals and have either to avoid or to mitigate such blow-up [36, 27]. Any algorithm
that tests ideal membership and wants to make good use of negative literals should be more
adaptive than, say, the standard Buchberger’s algorithm. It should figure out when to reduce
between \( x_i \) and \( \pi_i \), depending on the context.

Back to the theoretical aspects of this work, the separation formula for sums-of-squares
has unbounded width. Since we manage to get formulas of constant width for the others
proof systems, we would like to do the same for sums-of-squares. Is this possible? The issue
here is not so much our proof techniques, which has been more than enough for all the other
proof systems discussed in this paper, but the not so surprising fact that the lower bound
technology for sums-of-squares is quite behind the one for NS, PC and SA. It seems fair
to say that due to research progress that has happened during the last few years we now
have a situation where many of the open problems regarding algebraic proof system and how
they relate to one another have been resolved (see for example [11]). We know how different
complexity measures relate [26, 2, 23, 33, 22, 4] and whether these systems admit efficient
proof search [6, 17]. Yet the situation for sums-of-squares is far from being so positive. We
still do not understand the complexity of many important formulas in this proof systems.
References


