Circular (Yet Sound) Proofs

Albert Atserias[1] and Massimo Lauria[2]

[1] Universitat Politècnica de Catalunya
[2] Sapienza - Università di Roma

Abstract. We introduce a new way of composing proofs in rule-based proof systems that generalizes tree-like and dag-like proofs. In the new definition, proofs are directed graphs of derived formulas, in which cycles are allowed as long as every formula is derived at least as many times as it is required as a premise. We call such proofs circular. We show that, for all sets of standard inference rules, circular proofs are sound. We first focus on the circular version of Resolution, and see that it is stronger than Resolution since, as we show, the pigeonhole principle has circular Resolution proofs of polynomial size. Surprisingly, as proof systems for deriving clauses from clauses, Circular Resolution turns out to be equivalent to Sherali-Adams, a proof system for reasoning through polynomial inequalities that has linear programming at its base. As corollaries we get: 1) polynomial-time (LP-based) algorithms that find circular Resolution proofs of constant width, 2) examples that separate circular from dag-like Resolution, such as the pigeonhole principle and its variants, and 3) exponentially hard cases for circular Resolution. Contrary to the case of circular resolution, for Frege we show that circular proofs can be converted into tree-like ones with at most polynomial overhead.

1 Introduction

In rule-based proof systems, proofs are traditionally presented as sequences of formulas, where each formula is either a hypothesis, or follows from some previous formulas in the sequence by one of the inference rules. Equivalently, such a proof can be represented by a directed acyclic graph, or *dag*, with one vertex for each formula in the sequence, and edges pointing forward from the premises to the conclusions. In this paper we introduce a new way of composing proofs: we allow cycles in this graph as long as every formula is derived at least as many times as it is required as a premise, and show that this structural condition is enough to guarantee soundness. Such proofs we call *circular*.

More formally, our definition is phrased in terms of *flow assignments*: each rule application must carry a positive integer, its *flow* or *multiplicity*, which intuitively means that in order to produce that many copies of the conclusion of the rule we must have produced at least that many copies of each of the premises first. Flow assignments induce a notion of *balance* of a formula in the proof, which is the difference between the number of times that the formula is produced as a conclusion and the number of times that it is required as a premise. Given these definitions, a proof-graph will be a valid circular proof if it admits a flow assignment that satisfies the following *flow-balance* condition: the only formulas of strictly negative balance are the hypotheses, and the formula that needs to be proved displays strictly positive balance. While proof-graphs with unrestricted cycles are, in general, unsound, we show that circular proofs *are* sound.

Atserias, A.; Lauria, M. Circular (yet sound) proofs. A: International Conference on Theory and Applications of Satisfiability Testing. "Theory and Applications of Satisfiability Testing–SAT 2019, 22nd International Conference: SAT 2019, Lisbon, Portugal, July 9–12, 2019: proceedings". Berlín: Springer, 2019, p. 1-18.
The final authenticated version is available online at https://doi.org/10.1007/978-3-030-24258-9_1

Proof complexity of circular proofs With all the definitions in place, we study the power of circular proofs from the perspective of propositional proof complexity.

For Resolution, we show that circularity *does* make a real difference. First we show that the standard propositional formulation of the pigeonhole principle has circular Resolution proofs of polynomial size. This is in sharp contrast with the well-known fact that Resolution *cannot count*, and that the pigeonhole principle is exponentially hard for (tree-like and dag-like) Resolution [16]. Second we observe that the LP-based proof of soundness of circular Resolution can be formalized in the Sherali-Adams proof system (with twin variables), which is a proof system for reasoning with polynomial inequalities that has linear programming at its base. Sherali-Adams was originally conceived as a hierarchy of linear programming relaxations for integer programs, but it has also been studied from the perspective of proof complexity in recent years.

Surprisingly, the converse simulation holds too! For deriving clauses from clauses, Sherali-Adams proofs translate efficiently into circular Resolution proofs. Moreover, both translations, the one from circular Resolution into Sherali-Adams and its converse, are efficient in terms of their natural parameters: length/size and width/degree. As corollaries we obtain for Circular Resolution all the proof complexity-theoretic properties that are known to hold for Sherali-Adams: 1) a polynomial-time (LP-based) proof search algorithm for proofs of bounded width, 2) length-width relationships, 3) separations from dag-like length and width, and 4) explicit exponentially hard examples.

Going beyond resolution we address the question of how circularity affects more powerful proof systems. For Frege systems, which operate with arbitrary propositional formulas through the standard textbook inference rules, we show that circularity adds no power: the circular, dag-like and tree-like variants of Frege polynomially simulate one another. The equivalence between the dag-like and tree-like variants of Frege is well-known [18]; here we add the circular variant to the list.

Earlier work While the idea of allowing cycles in proofs is not new, all the instances from the literature that we are aware of are designed for reasoning about inductive definitions, and not for propositional logic, nor for arbitrary inference-based proofs.

Shoesmith and Smiley [23] initiate the study of inference based proofs with multiple conclusions. In order to do so they introduce a graphical representation of proofs where nodes represents either formulas or inference steps, in a way similar to our definition in Section 2. While most of that book does not consider proof with cycles, in Section 10.5 they do mention briefly this possibility but they do not analyze it any further.

Niwińksi and Walukiewicz [19] introduced an infinitary tableau method for the modal μ-calculus. The proofs are regular infinite trees that are represented by finite graphs with cycles, along with a decidable *progress condition* on the cycles to guarantees their soundness. A sequent calculus version of this tableau method was proposed in [13], and explored further in [24]. In his PhD thesis, Brotherston [8] introduced a *cyclic* proof system for the extension of first-order logic with inductive definitions; see also [9] for a journal article presentation of the results. The proofs in [9] are ordinary proofs of the first-order sequent calculus extended with the rules that define the inductive predicates, along with a set of *backedges* that link equal formulas in the proof. The soundness is guaranteed by an additional *infinite descent condition* along the cycles that is very much inspired by the progress condition in Niwiński-Walukiewicz' tableau method. We refer the reader to Section 8 from [9] for a careful overview of the various flavours of proofs with cycles for logics with inductive definitions. From tracing the references in this body of the literature, and as far as we know, it seems that our flow-based definition of circular proofs had not been considered before.

The Sherali-Adams hierarchy of linear programming relaxations has received considerable attention in recent years for its relevance to combinatorial optimization and approximation algorithms; see the original [21], and [4] for a recent survey. In its original presentation, the Sherali-Adams hierarchy can already be thought of as a proof system for reasoning with polynomial inequalities, with the levels of the hierarchy corresponding to the degrees of the polynomials. For propositional logic, the system was studied in [11], and developed further in [20,3]. Those works consider the version of the proof system in which each propositional variable X comes with a formal *twin variable* $\bar{X}$, that is to be interpreted by the negation of X. This is the version of Sherali-Adams that we use. It was already known from [12] that this version of the Sherali-Adams proof system polynomially simulates standard Resolution, and has polynomial-size proofs of the pigeonhole principle.

2 Preliminaries

Formulas and Resolution proofs A *literal* is a variable X or the negation of a variable $\overline{X}$. A *clause* is a disjunction (or set) of literals, and a formula in conjunctive normal form, a *CNF formula*, is a conjunction (or set) of clauses. We use 0 to denote the empty clause. A *truth-assignment* is a mapping that assigns a truth-value *true* (1) or *false* (0) to each variable. A clause is true if one of its literal is true, and false otherwise. A CNF is true if all its clauses are true and false otherwise.

A *resolution* proof of a clause A from a CNF formula $C_1 \wedge \ldots \wedge C_m$ is a sequence of clauses $A_1, A_2, \ldots, A_r$ where $A_r = A$ and each A_i is either contained in $C_1, \ldots, C_m$ or is obtained by one of the following inference rules from earlier clauses in the sequence:

$$\frac{}{X \vee \overline{X}} \qquad \frac{C \vee X \qquad D \vee \overline{X}}{C \vee D} \qquad \frac{C}{C \vee D}. \tag{1}$$

Here C and D are clauses, and X must be some variable occurring in $C_1 \wedge \ldots \wedge C_m$. The inference rules in (1) are called *axiom*, *cut*, and *weakening*, respectively.

Resolution with Symmetric Rules Most standard inference rules in the literature are defined to derive a single consequent formula from one or more antecedents. For standard, non-circular proofs, this is no big loss in generality. However, for the proof complexity of circular proofs a particular rule with two consequent formulas will play an important role. Consider the variant of Resolution defined through the axiom rule and the following two nicely symmetric-looking inference rules:

$$\frac{C \vee X \qquad C \vee \overline{X}}{C} \qquad\qquad \frac{C}{C \vee X \qquad C \vee \overline{X}}. \tag{2}$$

These rules are called *symmetric cut* and *symmetric weakening*, or *split*, respectively. Note the subtle difference between the symmetric cut rule and the standard cut rule in (1):

in the symmetric cut rule, both antecedent formulas have the same *side formula* C. This difference is minor: an application of the non-symmetric cut rule that derives $C \vee D$ from $C \vee X$ and $D \vee \overline{X}$ may be efficiently simulated as follows: derive $C \vee X \vee D$ and $D \vee \overline{X} \vee C$ by sequences of $|D|$ and $|C|$ splits on $C \vee X$ and $D \vee \overline{X}$, respectively, and then derive $C \vee D$ by symmetric cut. Here $|C|$ and $|D|$ denote the widths of C and D. The standard weakening rule may be simulated also by a sequence of splits. Thus, Resolution may well be defined this way with little conceptual change. The choice of this form is not just due to elegance and symmetry: it makes it easier to deal with the concept of flow assignments that we introduce later, and to highlight the connection with Sherali-Adams proofs.

Resolution Complexity Measures A *resolution refutation of* $C_1 \wedge \ldots \wedge C_m$ is a resolution proof of the empty formula 0 from $C_1 \wedge \ldots \wedge C_m$. Its *length*, also called *size*, is the length of the sequence of clauses that constitutes it. The *width* of a resolution proof is the number of literals of its largest clause. If only the cut inference rule is allowed, resolution is sound and complete as a refutation system, meaning that a CNF is unsatisfiable if and only if it has a resolution refutation. Adding the axiom rule and weakening rules, resolution is sound and complete for deriving clauses from clauses.

Proofs are defined as sequences but are naturally represented through directed acyclic graphs, a.k.a. *dags*; see Figure 1. The graph has one *formula-vertex* for each formula in the sequence (the boxes), and one *inference-vertex* for each inference step that produces a formula in the sequence (the circles). Each formula-vertex is labelled by the corresponding formula, and each inference-vertex is labelled by the corresponding instance of the corresponding inference rule. Each inference-vertex has an incoming edge from any formula-vertex that corresponds to one of its premises, and at least one outgoing edge towards the corresponding consequent formula-vertices. The *proof-graph* of a proof Π is its associated dag and is denoted $G(\Pi)$. A proof Π is *tree-like* if $G(\Pi)$ is a tree.

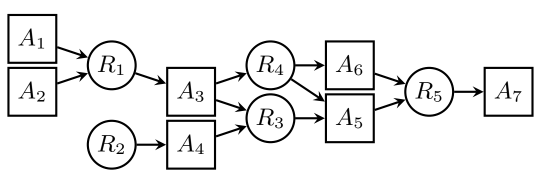

Fig. 1: A proof graph. All rules except R_4 have exactly one consequent formula; R_4 has two. All rules except R_2 have at least one antecedent formula; R_2 has none.

Sherali-Adams Proof System Let $A_1, \ldots, A_m$ and A be polynomials on $X_1, \ldots, X_n$ and $\bar{X}_1, \ldots, \bar{X}_n$; variables X_i and $\bar{X}_i$ are twins with the intended meaning that $\bar{X}_i = 1 - X_i$. A *Sherali-Adams proof of* $A \geq 0$ *from* $A_1 \geq 0, \ldots, A_m \geq 0$ is an identity

$$\sum_{j=1}^{t} Q_j P_j = A, \tag{3}$$

where each Q_j is a non-negative linear combination of monomials on the variables $X_1, \dots, X_n$ and $\bar{X}_1, \dots, \bar{X}_n$, and each P_j is a polynomial among $A_1, \dots, A_m$ or one among the following set of *basic* polynomials: $X_i - X_i^2$, $X_i^2 - X_i$, $1 - X_i - \bar{X}_i$, $X_i + \bar{X}_i - 1$, and 1. The *degree* of the proof is the maximum of the degrees of the polynomials $Q_j P_j$ in (3). The *monomial size* of the proof is the sum of the monomial sizes of the polynomials $Q_j P_j$ in (3), where the monomial size of a polynomial is the number of monomials with non-zero coefficient in its unique representation as a linear combination of monomials.

Simulation A proof system P *polynomially simulates* another proof system P' if there is a polynomial-time algorithm that, given a proof Π' in P' as input, computes a proof Π in P, such that Π has the same goal formula and the same hypothesis formulas as Π'.

3 Circular Proofs

Circular Pre-Proofs A *circular pre-proof* is just an ordinary proof with *backedges* that match equal formulas. More formally, a *circular pre-proof* from a set $\mathscr{H}$ of hypothesis formulas is a proof $A_1, \dots, A_\ell$ from an augmented set of hypothesis formulas $\mathscr{H} \cup \mathscr{B}$, together with a set of *backedges* that is represented by a set $M \subseteq [\ell] \times [\ell]$ of pairs (i, j), with $j < i$, such that $A_j = A_i$ and $A_j \in \mathscr{B}$. The formulas in the set $\mathscr{B}$ of additional hypotheses are called *bud formulas*.

Just like ordinary proofs are natually represented by directed acyclic graphs, circular pre-proofs are natually represented by directed graphs; see Figure 2. For each pair (i, j) in M there is a *backedge* from the formula-vertex of A_i to the formula-vertex of the bud formula A_j; note that $A_j = A_i$ by definition. By contracting the backedges of a circular pre-proof we get an ordinary directed graph with cycles. If Π is a circular pre-proof, we use $G(\Pi)$ to denote this graph, which we call the *compact graph representation of* Π. Note that $G(\Pi)$ is a bipartite graph with all its edges pointing from a formula-vertex to an inference-vertex, or vice-versa. When Π is clear from the context, we write I and J for the sets of inference- and formula-vertices of $G(\Pi)$, respectively, and $N^-(u)$ and $N^+(u)$ for the sets of in- and out-neighbours of a vertex u of $G(\Pi)$, respectively.

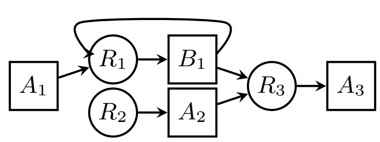

Fig. 2: The compact graph representation of a circular pre-proof.

In general, circular pre-proofs need not be sound; for an example we refer to the full version of the paper. In order to ensure soundness we need to require a global condition as defined next.

Circular Proofs Let Π be a circular pre-proof. A *flow assignment* for Π is an assignment $F : I \to \mathbb{R}^+$ of positive real weights, or *flows*, where I is the set of inference-vertices of the compact graph representation $G(\Pi)$ of Π. The flow-extended graph that labels each inference-vertex w of $G(\Pi)$ by its flow $F(w)$ is denoted $G(\Pi, F)$. The *inflow* of a formula-vertex in $G(\Pi, F)$ is the sum of the flows of its in-neighbours. Similarly, the *outflow* of a formula-vertex in $G(\Pi, F)$ is the sum of the flows of its out-neighbours. The *balance* of a formula-vertex u of $G(\Pi, F)$ is the inflow minus the outflow of u, and is denoted $B(u)$. In symbols,

$$B(u) := \sum_{w \in N^-(u)} F(w) - \sum_{w \in N^+(u)} F(w). \tag{4}$$

The formula-vertices of strictly negative balance are the *sources* of $G(\Pi, F)$, and those of strictly positive balance are the *sinks* of $G(\Pi, F)$. We think of flow assignments as witnessing a proof of a formula that labels a sink, from the set of formulas that label the sources. Concretely, for a given set of hypothesis formulas $\mathscr{H}$ and a given goal formula A, we say that the flow assignment *witnesses a proof of A from $\mathscr{H}$* if every source of $G(\Pi, F)$ is labelled by a formula in $\mathscr{H}$, and some sink of $G(\Pi, F)$ is labelled by the formula A.

Finally, a *circular proof of A from $\mathscr{H}$* is a circular pre-proof for which there exists a flow assignment that witnesses a proof of A from $\mathscr{H}$. The *length* of a circular proof Π is the number of vertices of $G(\Pi)$, and the *size* of Π is the sum of the sizes of the formulas in the sequence. Note that this definition of size does not depend on the weights that witness the proof. The next lemma states that such weights can be found efficiently, may be assumed to be integral, and have small bit-complexity. For the proof of this lemma we refer to the full version of the paper.

Lemma 1. *There is a polynomial-time algorithm that, given a circular pre-proof Π, a finite set of hypothesis formulas $\mathscr{H}$, and a goal formula A as input, returns a flow assignment for Π that witnesses a proof of A from $\mathscr{H}$, if it exists. Moreover, if the length of the pre-proof is ℓ, then the flows returned by the algorithm are positive integers bounded by $\ell!$.*

Soundness of Circular Resolution Proofs We still need to argue that the existence of a witnessing flow assignment guarantees soundness. In this section we develop the soundness proof for resolution as defined in (2). See Section 2 for a discussion on this choice of rules. The proof of the following theorem generalizes to circular proof systems based on more powerful inference rules with essentially no changes, but here we keep the discussion focused on resolution. In the full paper we develop the general case.

Theorem 1. *Let $\mathscr{H}$ be a set of hypothesis formulas and let A be a goal formula. If there is a circular resolution proof of A from $\mathscr{H}$ then every truth assignment that satisfies every formula in $\mathscr{H}$ also satisfies A.*

Proof. Fix a truth assignment α. We prove the stronger claim that, for every circular pre-proof Π from an unspecified set of hypothesis formulas, every integral flow assignment F for Π, and every sink s of $G(\Pi, F)$, if α falsifies the formula that labels s, then α also falsifies the formula that labels some source of $G(\Pi, F)$. The restriction to integral flow assignments is no loss of generality by Lemma 1, and allows a proof by induction on the sum of the flow among all inference-vertices, which we will call the total flow-sum of F.

If the total flow-sum is zero, then there are no inference-vertices, hence there are no sinks, and the statement holds vacuously. Assume then that the total flow-sum is positive, and let s be some sink of $G(\Pi, F)$ with balance $B(s) > 0$ so that the formula labels it is falsified by α. Since its balance is positive, s must have at least one in-neighbour r. Since the consequent formula of the rule at r is falsified by α, some antecedent formula of the rule at r must exist that is also falsified by α. Let u be the corresponding in-neighbour of r, and let $B(u)$ be its balance. If $B(u)$ is negative, then u is a source of $G(\Pi, F)$, and we are done. Assume then that $B(u)$ is non-negative.

Let $\delta := \min\{B(s), F(r)\}$ and note that $\delta > 0$ since $B(s) > 0$ and $F(r) > 0$. We define a new circular pre-proof Π' and an integral flow assignment F' for Π' to which we will apply the induction hypothesis. The construction will guarantee the following properties:

1. $G(\Pi')$ is a subgraph of $G(\Pi)$ with the same set of formula-vertices,
2. the total flow-sum of F' is smaller than the total flow-sum of F.
3. u is a sink of $G(\Pi', F')$ and s is not a source of $G(\Pi', F')$,
4. if t is a source of $G(\Pi', F')$, then t is a source of $G(\Pi, F)$ or an out-neighbour of r in $G(\Pi)$.

From this the claim will follow by applying the induction hypothesis to Π', F' and u. Indeed the induction hypothesis applies to them by Properties 1, 2 and the first half of 3, and it will give a source t of $G(\Pi', F')$ whose labelling formula is falsified by α. We argue that t must also be a source of $G(\Pi, F)$, in which case we are done. To argue for this, assume otherwise and apply Property 4 to conclude that t is an out-neighbour of r in $G(\Pi)$, which by the second half of Property 3 must be different from s because t is a source of $G(\Pi', F')$. Recall now that s is a second out-neighbour of r. This can be the case only if r is a split inference, in which case the formulas that label s and t must be of the form $C \vee X$ and $C \vee \overline{X}$, respectively, for appropriate formula C and variable X. But, by assumption, α falsifies the formula that labels s, let us say $C \vee X$, which means that α satisfies the formula $C \vee \overline{X}$ that labels t. This is the contradiction we were after.

It remains to construct Π' and F' that satisfy properties 1, 2, 3, and 4. We define them by cases according to whether $F(r) > B(s)$ or $F(r) \leq B(s)$, and then argue for the correctness of the construction. In case $F(r) > B(s)$, and hence $\delta = B(s)$, let Π' be defined as Π without change, and let F' be defined by $F'(r) := F(r) - \delta$ and $F'(w) := F(w)$ for every other $w \in I \setminus \{r\}$. Obviously Π' is still a valid pre-proof and F' is a valid flow assignment for Π' by the assumption that $F(r) > B(s) = \delta$. In case $F(r) \leq B(s)$, and hence $\delta = F(r)$, let Π' be defined as Π with the inference-step that labels r removed, and let F' be defined by $F'(w) := F(w)$ for every $w \in I \setminus \{r\}$. Note that in this case Π' is still a valid pre-proof but perhaps from a larger set of hypothesis formulas.

In both cases the proof of the claim that Π' and F' satisfy Properties 1, 2, 3, and 4 is the same. Property 1 is obvious in both cases. Property 2 follows from the fact that the total flow-sum of F' is the total flow-sum of F minus δ, and $\delta > 0$. The first half of Property 3 follows from the fact that the balance of u in $G(\Pi', F')$ is $B(u) + \delta$, while $B(u) \geq 0$ by assumption and $\delta > 0$. The second half of Property 3 follows from the fact that the balance of s in $G(\Pi', F')$ is $B(s) - \delta$, while $B(s) \geq \delta$ by choice of δ. Property 4 follows from the fact that the only formula-vertices of $G(\Pi', F')$ of balance smaller than that in $G(\Pi, F)$ are the out-neighbours of r. This completes the proof of the claim, and of the theorem. □

4 Circular Resolution

In this section we investigate the power of Circular Resolution. Recall from the discussion in Section 2 that Resolution is traditionally defined to have cut as its only rule, but that an essentially equivalent version of it is obtained if we define it through symmetric cut, split, and axiom, still all restricted to clauses. This more liberal definition of Resolution, while staying equivalent vis-a-vis the tree-like and dag-like versions of Resolution, will play an important role for the circular version of Resolution.

In this section we show that circular Resolution can be exponentially stronger than dag-like Resolution. Indeed, we show that Circular Resolution is polynomially equivalent with the Sherali-Adams proof system, which is already known to be stronger than dag-like Resolution:

Theorem 2. *Sherali-Adams and Circular Resolution polynomially simulate each other. Moreover, the simulation one way converts degree into width (exactly), and the simulation in the reverse way converts width into degree (also exactly).*

For the statement of Theorem 2 to even make sense, Sherali-Adams is to be understood as a proof system for deriving clauses from clauses, under an appropriate encoding.

Pigeonhole Principles Let G be a bipartite graph with vertex bipartition (U, V), and set of edges $E \subseteq U \times V$. For a vertex $w \in U \cup V$, we write $N_G(w)$ to denote the set of neighbours of w in G, and $\deg_G(w)$ to denote its degree. The Graph Pigeonhole Principle of G, denoted G-PHP, is a CNF formula that has one variable $X_{u,v}$ for each edge (u, v) in E and the following set clauses:

$$\begin{array}{ll} X_{u,v_1} \vee \cdots \vee X_{u,v_d} & \text{for } u \in U \text{ with } N_G(u) = \{v_1, \ldots, v_d\}, \\ \overline{X_{u_1,v}} \vee \overline{X_{u_2,v}} & \text{for } u_1, u_2 \in U,\ u_1 \neq u_2, \text{ and } v \in N_G(u_1) \cap N_G(u_2). \end{array}$$

If $|U| > |V|$, and in particular if $|U| = n + 1$ and $|V| = n$, then G-PHP is unsatisfiable by the pigeonhole principle. For $G = K_{n+1,n}$, the complete bipartite graph with sides of sizes $n + 1$ and n, the formula G-PHP is the standard CNF encoding PHP_n^{n+1} of the pigeonhole principle.

Even for certain constant degree bipartite graphs with $|U| = n+1$ and $|V| = n$, the formulas are hard for Resolution.

Theorem 3 **([5,16]).** *There are families of bipartite graphs $(G_n)_{n\geq 1}$, where G_n has maximum degree bounded by a constant and vertex bipartition (U, V) of G_n that satisfies $|U| = n + 1$ and $|V| = n$, such that every Resolution refutation of G_n-PHP has width $\Omega(n)$ and length $2^{\Omega(n)}$. Moreover, this implies that every Resolution refutation of PHP_n^{n+1} has length $2^{\Omega(n)}$.*

In contrast, we show that these formulas have Circular Resolution refutations of polynomial length and, simultaneously, constant width. This result already follows from Theorem 2 plus the fact that Sherali-Adams has short refutations for G-PHP, of degree proportional to the maximum degree of G. Here we show a self-contained separation of Resolution from Circular Resolution that does not rely of Sherali-Adams.

Theorem 4. *For every bipartite graph G of maximum degree d with bipartition (U, V) such that $|U| > |V|$, there is a Circular Resolution refutation of G-PHP of length polynomial in $|U| + |V|$ and width d. In particular, PHP_n^{n+1} has a Circular Resolution refutation of polynomial length.*

Proof. We are going to build the refutation in pieces. Concretely, for every $u \in U$ and $v \in V$, we describe a Circular Resolution proof $\Pi_{u\to}$ and $\Pi_{\to v}$, with their associated flow assignments. These proofs will have width bounded by $\deg_G(u)$ and $\deg_G(v)$, respectively, and size polynomial in $\deg_G(u)$ and $\deg_G(v)$, respectively. Moreover, the following properties will be ensured:

1. The proof-graph of $\Pi_{u\to}$ contains a formula-vertex labelled by the empty clause 0 with balance $+1$ and a formula-vertex labelled $\overline{X_{u,v}}$ with balance -1 for every $v \in N_G(u)$; any other formula-vertex that has negative balance is labelled by a clause of G-PHP.
2. The proof-graph $\Pi_{\to v}$ contains a formula-vertex labelled by the empty clause 0 with balance -1 and a formula-vertex labelled by $\overline{X_{u,v}}$ with balance $+1$ for every $u \in N_G(v)$; any other formula-vertex that has negative balance is labelled by a clause of G-PHP.

We join these pieces by identifying the two formula-vertices labeled $\overline{X_{u,v}}$, for each $\{u, v\} \in E(G)$. Each such vertex gets balance -1 from $\Pi_{u\to}$ and $+1$ from $\Pi_{\to v}$, thus its balance is zero. The empty clause occurs $|U|$ times on formula vertices with balance $+1$, and $|V|$ times on formula vertices with balance -1. Now we add back-edges from $|V|$ of the $|U|$ formula-vertices with balance $+1$ to all the formula-vertices with balance -1, forming a matching. The only remaining formula-vertices labeled by 0 have positive balance. Now in the proof all the formula-vertices that have negative balance are clauses of G-PHP, and the empty clause 0 has positive balance. This is indeed a Circular Resolution refutation of G-PHP. See Figure 3 for a diagram of the proof for PHP_3^4.

For the construction of $\Pi_{u\to}$, rename the neighbours of u as $1, 2, \dots, \ell$. Let C_j denote the clause $X_{u,1} \lor \cdots \lor X_{u,j}$ and note that C_ℓ is a clause of G-PHP. Split $\overline{X_{u,\ell}}$ on $X_{u,1}$, then on $X_{u,2}$, and so on up to $X_{u,\ell-1}$ until we produce $\overline{X_{u,\ell}} \lor C_{\ell-1}$. Then resolve this clause with C_ℓ to produce $C_{\ell-1}$. The same construction starting at $\overline{X_{u,\ell-1}}$ and $C_{\ell-1}$ produces $C_{\ell-2}$. Repeating ℓ times we get down to the empty clause.

For the construction of $\Pi_{\to v}$ we need some more work. Again rename the neighbours of v as $1, 2, \dots, \ell$. We define a sequence of proofs $\Pi_1, \dots, \Pi_\ell$ inductively. The base case Π_1 is just one application of the split rule to the empty clause to derive $X_{1,v}$ and $\overline{X_{1,v}}$, with flow 1. Proof Π_{i+1} is built using Π_i as a component. Let $X_{i+1,v} \lor \Pi_i$ denote the proof that is obtained from adding the literal $X_{i+1,v}$ to every clause in Π_i. First we observe that $X_{i+1,v} \lor \Pi_i$ has balance -1 on $X_{i+1,v}$ and balance $+1$ on, among other

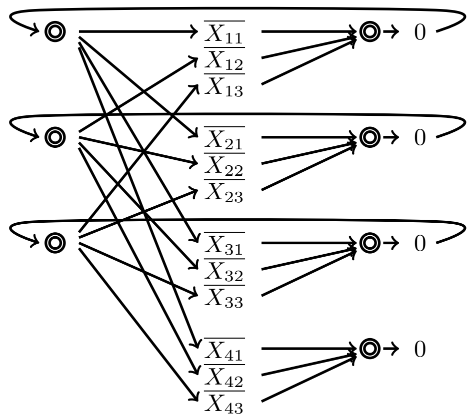

Fig. 3: The diagram of the circular proof of PHP^4_3. The double circles indicate multiple inferences. The empty clause 0 is derived four times and used only three times.

clauses, $X_{i+1,v} \vee \overline{X_{j,v}}$ for $j = 1, \ldots, i$. Each such clause can be resolved with clause $\overline{X_{i+1,v}} \vee \overline{X_{j,v}}$ to produce the desired clauses $\overline{X_{j,v}}$ with balance $+1$. Splitting the empty clause on variable $X_{i+1,v}$ would even out the balance of the formula-vertex labelled by $X_{i+1,v}$ and produce a vertex labelled by $\overline{X_{i+1,v}}$ of balance $+1$. Take $\Pi_\ell = \Pi_{\to v}$. $\square$

Equivalence with Sherali-Adams In this section we prove each half of Theorem 2 in a separate lemma. We need some preparation. Fix a set of variables $X_1, \ldots, X_n$ and their twins $\bar{X}_1, \ldots, \bar{X}_n$. For a clause $C = \bigvee_{j \in Y} X_j \vee \bigvee_{j \in Z} \overline{X_j}$ with $Y \cap Z = \emptyset$, define

$$T(C) := -\prod_{j \in Y} \bar{X}_j \prod_{j \in Z} X_j, \tag{5}$$

Observe that a truth assignment satisfies C if and only if the corresponding 0-1 assignment for the variables of $T(C)$ makes the inequality $T(C) \geq 0$ true. There is an alternative encoding of clauses into inequalities that is sometimes used. Define $L(C) := \sum_{j \in Y} X_j + \sum_{j \in Z} \bar{X}_j - 1$, and observe that a truth assignment satisfies C if and only if the corresponding 0-1 assignment makes the inequality $L(C) \geq 0$ true. We state the results of this section for the T-encoding of clauses, but the same result would hold for the L-encoding because there are efficient SA proofs of $T(C) \geq 0$ from $L(C) \geq 0$, and vice-versa. We will use the following lemma, which is a variant of Lemma 4.4 in [3]:

Lemma 2. *Let $w \geq 2$ be an integer, let C be a clause with at most w literals, let D be a clause with at most $w - 1$ literals, and let X be a variable that does not appear in D. Then the following four inequalities have Sherali-Adams proofs (from nothing) of constant monomial size and degree w:*

1. $T(X \vee \overline{X}) \geq 0$,
2. $-T(D \vee \overline{X}) - T(D \vee X) + T(D) \geq 0$,

3. $-T(D) + T(D \vee \overline{X}) + T(D \vee X) \geq 0$,
4. $-T(C) \geq 0$.

Proof. Let $D = \bigvee_{i \in Y} X_i \vee \bigvee_{j \in Z} X_j$ and $C = \bigvee_{i \in Y'} X_i \vee \bigvee_{j \in Z'} X_j$. Then

1. $T(X \vee \overline{X}) = (1 - X - \bar{X}) \cdot X + (X^2 - X)$,
2. $-T(D \vee \overline{X}) - T(D \vee X) + T(D) = (X + \bar{X} - 1) \cdot \prod_{i \in Y} \bar{X}_i \prod_{j \in Z} X_j$,
3. $-T(D) + T(D \vee \overline{X}) + T(D \vee X) = (1 - X - \bar{X}) \cdot \prod_{i \in Y} \bar{X}_i \prod_{j \in Z} X_j$,
4. $-T(C) = 1 \cdot \prod_{i \in Y'} \bar{X}_i \prod_{j \in Z'} X_j$.

The claim on the monomial size and the degree follows. □

Lemma 3. *Let $A_1, \ldots, A_m$ and A be non-tautological clauses. If there is a Circular Resolution proof of A from $A_1, \ldots, A_m$ of length s and width w, then there is a Sherali-Adams proof of $T(A) \geq 0$ from $T(A_1) \geq 0, \ldots, T(A_m) \geq 0$ of monomial size $O(s)$ and degree w.*

Proof. Let Π be a Circular Resolution proof of A from $A_1, \ldots, A_m$, and let F be the corresponding flow assignment. Let I and J be the sets of inference- and formula-vertices of $G(\Pi)$, and let $B(u)$ denote the balance of formula-vertex $u \in J$ in $G(\Pi, F)$. For each formula-vertex $u \in J$ labelled by formula A_u, define the polynomial $P_u := T(A_u)$. For each inference-vertex $w \in I$ labelled by rule R, with sets of in- and out-neighbours N^- and N^+, respectively, define the polynomial

$P_w := T(A_a)$	if $R =$ axiom with $N^+ = \{a\}$,
$P_w := -T(A_a) - T(A_b) + T(A_c)$	if $R =$ cut with $N^- = \{a, b\}$ and $N^+ = \{c\}$,
$P_w := -T(A_a) + T(A_b) + T(A_c)$	if $R =$ split with $N^- = \{a\}$ and $N^+ = \{b, c\}$.

By double counting, the following polynomial identity holds:

$$\sum_{u \in J} B(u) P_u = \sum_{w \in I} F(w) P_w. \tag{6}$$

Let s be the sink of $G(\Pi, F)$ that is labelled by the derived clause A. Since $B(s) > 0$, equation (6) rewrites into $\sum_{w \in I} F(w)/B(s) P_w + \sum_{u \in J \setminus \{s\}} -B(u)/B(s) P_u = P_s$. We claim that this identity is a legitimate Sherali-Adams proof of $T(A) \geq 0$ from the inequalities $T(A_1) \geq 0, \ldots, T(A_m) \geq 0$. First, $P_s = T(A_s) = T(A)$, i.e. the right-hand side is correct. Second, each term $(F(w)/B(s))P_w$ for $w \in I$ is a sum of legitimate terms of a Sherali-Adams proof by the definition of P_w and Parts 1, 2 and 3 of Lemma 2. Third, since each source $u \in I$ of $G(\Pi, F)$ has $B(u) < 0$ and is labelled by a formula in $A_1, \ldots, A_m$, the term $(-B(u)/B(s))P_u$ of a source $u \in I$ is a positive multiple of $T(A_u)$ and hence also a legitimate term of a Sherali-Adams proof from $T(A_1) \geq 0, \ldots, T(A_m) \geq 0$. And forth, since each non-source $u \in I$ of $G(\Pi, F)$ has $B(u) \geq 0$, each term $(-B(u)/B(s))P_u$ of a non-source $u \in I$ is a sum of legitimate terms of a Sherali-Adams proof by the definition of P_u and Part 4 of Lemma 2. The monomial size and degree of this Sherali-Adams proof are as claimed. □

Lemma 4. *Let $A_1, \ldots, A_m$ and A be non-tautological clauses. If there is a Sherali-Adams proof of $T(A) \geq 0$ from $T(A_1) \geq 0, \ldots, T(A_m) \geq 0$ of monomial size s and degree d, then there is a Circular Resolution proof of A from $A_1, \ldots, A_m$ of length $O(s)$ and width d.*

Proof. Fix a Sherali-Adams proof of $T(A) \geq 0$ from $T(A_1) \geq 0, \ldots, T(A_m) \geq 0$, say $\sum_{j=1}^{t} Q_j P_j = T(A)$, where each Q_j is a non-negative linear combination of monomials on the variables $X_1, \ldots, X_n$ and $\bar{X}_1, \ldots, \bar{X}_n$, and each P_j is a polynomial from among $T(A_1), \ldots, T(A_m)$ or from among the list of basic polynomials from the definition of Sherali-Adams in Section 2.

Our goal is to massage the proof until it becomes a Circular Resolution proof in disguise. Towards this, as a first step, we claim that the proof can be transformed into a *normalized proof* of the form $\sum_{j=1}^{t'} Q'_j P'_j = T(A)$ that has the following properties: 1) each Q'_j is a positive multiple of a multilinear monomial, and $Q'_j P'_j$ is multilinear, and 2) each P'_j is a polynomial among $T(A_1), \ldots, T(A_m)$, or among the polynomials in the set $\{-X_i\bar{X}_i, 1 - X_i - \bar{X}_i, X_i + \bar{X}_i - 1 : i \in [n]\} \cup \{1\}$. Comparing the list of Boolean axioms in 2) with the original list of basic polynomials in the definition of Sherali-Adams, note that we have replaced the polynomials $X_i - X_i^2$ and $X_i^2 - X_i$ by $-X_i\bar{X}_i$. Note also that, by splitting the Q_j's into their terms, we may assume without loss of generality that each Q_j is a positive multiple of a monomial on the variables $X_1, \ldots, X_n$ and $\bar{X}_1, \ldots, \bar{X}_n$.

In order to prove the claim we rely on the well-known fact that each real-valued function over Boolean domain has a unique representation as a multilinear polynomial. With this fact in hand, it suffices to convert each $Q_j P_j$ in the left-hand side of the proof into a $Q'_j P'_j$ of the required form (or 0), and check that $Q_j P_j$ and $Q'_j P'_j$ are equivalent over the 0-1 assignments to its variables (without relying on the constraint that $\bar{X}_i = 1 - X_i$). The claim will follow from the fact that $T(A)$ is multilinear since, by assumption, A is non-tautological.

We proceed to the conversion of each $Q_j P_j$ into a $Q'_j P'_j$ of the required form. Recall that we assumed already, without loss of generality, that each Q_j is a positive multiple of a monomial. The multilinearization of a monomial Q_j is the monomial $M(Q_j)$ that results from replacing every factor Y^k with $k \geq 2$ in Q_j by Y. Obviously Q_j and $M(Q_j)$ agree on 0-1 assignments, but replacing each Q_j by $M(Q_j)$ is not enough to guarantee the normal form that we are after. We need to proceed by cases on P_j.

If P_j is one of the polynomials among $T(A_1), \ldots, T(A_m)$, say $T(A_i)$, then we let Q'_j be $M(Q_j)$ with every variable that appears in A_i deleted, and let P'_j be $T(A_i)$ itself. It is obvious that this works. If P_j is $1 - X_i - \bar{X}_i$, then we proceed by cases on whether Q_j contains X_i or $\bar{X}_i$ or both. If Q_j contains neither X_i nor $\bar{X}_i$, then the choice $Q'_j = M(Q_j)$ and $P'_j = P_j$ works. If Q_j contains X_i or $\bar{X}_i$, call it Y, but not both, then the choice $Q'_j = M(Q_j)/Y$ and $P'_j = -X_i\bar{X}_i$ works. If Q_j contains both X_i and $\bar{X}_i$, then the choice $Q'_j = M(Q_j)/(X_i\bar{X}_i)$ and $P'_j = -X_i\bar{X}_i$ works. If P_j is $X_i + \bar{X}_i - 1$, then again we proceed by cases on whether Q_j contains X_i or $\bar{X}_i$ or both. If Q_j contains neither X_i nor $\bar{X}_i$, then the choice $Q'_j = M(Q_j)$ and $P'_j = P_j$ works. If Q_j contains X_i or $\bar{X}_i$, call it Y, but not both, then the choice $Q'_j = M(Q_j)\bar{Y}$ and $P'_j = 1$ works. If Q_j contains both X_i and $\bar{X}_i$, then the choice $Q'_j = M(Q_j)$ and $P'_j = 1$ works. If P_j is the polynomial 1, then the choice $Q'_j = M(Q_j)$ and $P'_j = 1$ works. Finally, if P_j is of the form $X_i - X_i^2$ or $X_i^2 - X_i$, then we replace $Q_j P_j$ by 0. Observe that in this case $Q_j P_j$ is always 0 over 0-1 assignments, and the conversion is correct. This completes the proof that the normalized proof exists.

It remains to be seen that the normalized proof is a Circular Resolution proof in disguise. For each $j \in [m]$, let a_j and M_j be the positive real and the multilinear monomial, respectively, such that $Q'_j = a_j \cdot M_j$. Let also C_j be the unique clause on the variables $X_1, \ldots, X_n$ such that $T(C_j) = -M_j$. Let $[t']$ be partitioned into five sets $I_0 \cup I_1 \cup I_2 \cup I_3 \cup I_4$ where

1. I_0 is the set of $j \in [t']$ such that $P'_j = T(A_{i_j})$ for some $i_j \in [m]$,
2. I_1 is the set of $j \in [t']$ such that $P'_j = -X_{i_j}\bar{X}_{i_j}$ for some $i_j \in [n]$,
3. I_2 is the set of $j \in [t']$ such that $P'_j = 1 - X_{i_j} - \bar{X}_{i_j}$ for some $i_j \in [n]$,
4. I_3 is the set of $j \in [t']$ such that $P'_j = X_{i_j} + \bar{X}_{i_j} - 1$ for some $i_j \in [n]$,
5. I_4 is the set of $j \in [t']$ such that $P'_j = 1$.

Define new polynomials P''_j as follows:

$P''_j := T(C_j \vee A_{i_j})$ for $j \in I_0$,
$P''_j := T(C_j \vee \overline{X_{i_j}} \vee X_{i_j})$ for $j \in I_1$,
$P''_j := -T(C_j) + T(C_j \vee \overline{X_{i_j}}) + T(C_j \vee X_{i_j})$ for $j \in I_2$,
$P''_j := -T(C_j \vee \overline{X_{i_j}}) - T(C_j \vee X_{i_j}) + T(C_j)$ for $j \in I_3$,
$P''_j := T(C_j)$ for $j \in I_4$.

With this notation, the normalized proof rewrites into

$$\sum_{j \in I_0} a_j P''_j + \sum_{j \in I_1} a_j P''_j + \sum_{j \in I_2} a_j P''_j + \sum_{j \in I_3} a_j P''_j = T(A) + \sum_{j \in I_4} a_j P''_j. \qquad (7)$$

Finally we are ready to construct the circular proof. We build it by listing the inference-vertices with their associated flows, and then we identify together all the formula-vertices that are labelled by the same clause.

Intuitively, I_0's are weakenings of hypothesis clauses, I_1's are weakenings of axioms, I_2's are cuts, and I_3's are splits. Formally, each $j \in I_0$ becomes a chain of $|C_j|$ many split vertices that starts at the hypothesis clause A_{i_j} and produces its weakening $C_j \vee A_{i_j}$; all split vertices in this chain have flow a_j. Each $j \in I_1$ becomes a sequence that starts at one axiom vertex that produces $X_{i_j} \vee \overline{X_{i_j}}$ with flow a_j, followed by a chain of $|C_j|$ many split vertices that produces its weakening $C_j \vee X_{i_j} \vee \overline{X_{i_j}}$; all split vertices in this chain also have flow a_j. Each $j \in I_2$ becomes one cut vertex that produces C_j from $C_j \vee X_{i_j}$ and $C_j \vee \overline{X_{i_j}}$ with flow a_j. And each $j \in I_3$ becomes one split vertex that produces $C_j \vee X_{i_j}$ and $C_j \vee \overline{X_{i_j}}$ from C_j with flow a_j.

This defines the inference-vertices of the proof graph. The construction is completed by introducing one formula-vertex for each different clause that is an antecedent or a consequent of these inference-vertices. The construction was designed in such a way that equation (7) is the proof that, in this proof graph and its associated flow assignment, the following hold (see the full version for details): 1) there is a sink with balance 1 and that is labelled by A, 2) all sources are labelled by formulas among $A_1, \ldots, A_m$, and 3) all other formula-vertices have non-negative balance. This proves that the construction is a correct Circular Resolution proof. The claim that the length of this proof is $O(s)$ and its width is d follows by inspection. $\square$

5 Further Remarks

One immediate consequence of Theorem 2 is that there is a polynomial-time algorithm that automates the search for Circular Resolution proofs of bounded width:

Corollary 1. *There is an algorithm that, given an integer parameter w and a set of clauses $A_1, \ldots, A_m$ and A with n variables, returns a width-w Circular Resolution proof of A from $A_1, \ldots, A_m$, if there is one, and the algorithm runs in time polynomial in m and n^w.*

The proof-search algorithm of Corollary 1 relies on linear programming because it relies on our translations to and from Sherali-Adams, whose automating algorithm does rely on linear programming [22]. Based on the fact that the number of clauses of width w is about n^w, a direct proof of Corollary 1 is also possible, but as far as we see it still relies on linear programming for finding the flow assignment. It remains as an open problem whether a more combinatorial algorithm exists for the same task.

Another consequence of the equivalence with Sherali-Adams is that Circular Resolution has a length-width relationship in the style of the one for Dag-like Resolution [5]. This follows from Theorem 2 in combination with the size-degree relationship that is known to hold for Sherali-Adams (see [20,1]). Combining this with the known lower bounds for Sherali-Adams (see [15,20]), we get the following:

Corollary 2. *There are families of 3-CNF formulas $(F_n)_{n\geq 1}$, where F_n has $O(n)$ variables and $O(n)$ clauses, such that every Circular Resolution refutation of F_n has width $\Omega(n)$ and size $2^{\Omega(n)}$.*

It should be noticed that, unlike the well-known observation that tree-like and dag-like width are equivalent measures for Resolution, dag-like and circular width are *not* equivalent for Resolution. The sparse graph pigeonhole principle from Section 4 illustrates the point. This shows that bounded-width circular Resolution proofs cannot be *unfolded* into bounded-width tree-like Resolution proofs (except by going infinitary?).

For Resolution it makes a big difference whether the proof-graph has tree-like structure or not [7]. For Frege proof systems this is not the case, since Tree-like Frege polynomially simulates Dag-like Frege, and this holds true of any inference-based proof system with the set of all formulas as its set of allowed formulas, and a finite set of inference rules that is implicationally complete [18]. In the full version of the paper we show that, contrary to resolution, circular Frege proofs are no more powerful than standard ones. The main idea is to simulate in Dag-like Frege an alternative proof of the soundness of circular Frege that is based on linear programming. To do that we use a formalization of linear arithmetic, due to Buss[10] and Goerdt[14], which was originally designed to simulate counting arguments and Cutting Planes proofs in Dag-like Frege. Since Cutting Planes subsumes linear programming, the proof of soundness of circular Frege based on linear programming can be formalized in Dag-ike Frege.

Theorem 5 **([2]).** *Tree-like Frege and Circular Frege polynomially simulate each other.*

It is known that Tree-like Bounded-Depth Frege simulates Dag-like Bounded-Depth Frege, at the cost of increasing the depth by one. Could the simulation of Circular Frege by Tree-like Frege be made to preserve bounded depth? The (negative) answer is also provided by the pigeonhole principle which is known to be hard for Bounded-Depth Frege but is easy for Circular Resolution, and hence for Circular Depth-1 Frege.

One last aspect of the equivalence between Circular Resolution and Sherali-Adams concerns the theory of SAT-solving. As is well-known, state-of-the-art SAT-solvers produce Resolution proofs as certificates of unsatisfiability and, as a result, will not be able to handle counting arguments of pigeonhole type. This has motivated the study of so-called *pseudo-Boolean solvers* that handle counting constraints and reasoning through specialized syntax and inference rules. The equivalence between Circular Resolution and Sherali-Adams suggests a completely different approach to incorporate counting capabilities: instead of enhancing the syntax, keep it to clauses but enhance the *proof-shapes*. Whether circular proof-shapes can be handled in a sufficiently effective and efficient way is of course in doubt, but certainly a question worth studying.

It turns out that Circular Resolution has unexpected connections with Dual Rail MaxSAT Resolution [17]. MaxSAT Resolution is a variant of resolution where proofs give upper bounds on the number of clauses of the CNF that can be satisfied simultaneously. At the very least, when the upper bound is less than the number of clauses, MaxSAT resolution provides a refutation of the formula. The Dual Rail encoding is a special encoding of CNF formulas, and Dual Rail MaxSAT Resolution is defined to be MaxSAT resolution applied to the Dual Rail encoding of the input formula. In [6], the authors show that Dual Rail encoding gives strength to the proof system, providing a polynomial refutation of the pigeonhole principle formula. Recently [25] has argued that Circular Resolution polynomially simulates Dual Rail MaxSAT resolution, in the sense that when the Dual Rail encoding of a CNF formula F has a MaxSAT Resolution refutation of length ℓ and width w, then F has a Circular Resolution refutation of length $O(\ell w)$. This is interesting per se, and also provides yet another proof of Theorem 4.

Acknowledgments Both authors were partially funded by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement ERC-2014-CoG 648276 (AUTAR). First author partially funded by MICINN through TIN2016-76573-C2-1P (TASSAT3). We acknowledge the work of Jordi Coll who conducted experimental results for finding and visualizing actual circular resolution proofs of small instances of the sparse pigeonhole principle.

References

1. A. Atserias and T. Hakoniemi. Size-Degree Trade-offs for Sums-of-Squares and Positivstellensatz Proofs. To appear in Proceedings of 34th Annual Conference on Computational Complexity. Long version in arXiv:1811.01351 [cs.CC], 2018.
2. A. Atserias and M. Lauria. Circular (yet sound) proofs. *CoRR*, abs/1802.05266, 2018.
3. A. Atserias, M. Lauria, and J. Nordström. Narrow proofs may be maximally long. *ACM Trans. Comput. Log.*, 17(3):19:1–19:30, 2016.
4. Yu Hin Au and L. Tunçel. A comprehensive analysis of polyhedral lift-and-project methods. *SIAM J. Discrete Math.*, 30(1):411—451, 2016.

5. E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001.
6. M. L. Bonet, S. Buss, A. Ignatiev, J. Marques-Silva, and A. Morgado. MaxSAT Resolution With the Dual Rail Encoding. In *Proc. 32nd AAAI Conference on Artificial Intelligence*, 2018.
7. M. L. Bonet, J. L. Esteban, N. Galesi, and J. Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM J. Comp.*, 30(5), 2000.
8. J. Brotherston. *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, University of Edinburgh, November 2006.
9. J. Brotherston and A. Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2011.
10. Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52(4):916–927, 1987.
11. S. S. Dantchev. Rank complexity gap for Lovász-Schrijver and Sherali-Adams proof systems. In *Proc. 39th Annual ACM Symposium on Theory of Computing*, pages 311–317, 2007.
12. S. S. Dantchev, B. Martin, and M. N. C. Rhodes. Tight rank lower bounds for the Sherali-Adams proof system. *Theor. Comput. Sci.*, 410(21-23):2054–2063, 2009.
13. C. Dax, M. Hoffman, and M. Lange. *A Proof System for the Linear Time μ-Calculus*, pages 273–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
14. Andreas Goerdt. Cutting plane versus frege proof systems. In Egon Börger, Hans Kleine Büning, Michael M. Richter, and Wolfgang Schönfeld, editors, *Computer Science Logic: 4th Workshop, CSL '90 Heidelberg, Germany, October 1–5, 1990 Proceedings*, pages 174–194, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.
15. D. Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613 – 622, 2001.
16. A. Haken. The intractability of resolution. *Theor. Comp. Sci.*, 39:297 – 308, 1985.
17. Alexey Ignatiev, António Morgado, and João Marques-Silva. On tackling the limits of resolution in SAT solving. In *Proc. 20th International Conference on Theory and Applications of Satisfiability Testing - SAT 2017*, pages 164–183, 2017.
18. J. Krajíček. *Bounded Arithmetic, Propositional Logic, Complexity Theory*. Cambridge, 1994.
19. D. Niwiński and I. Walukiewicz. Games for the μ-calculus. *Theor. Comp. Sci.*, 163(1), 1996.
20. T. Pitassi and N. Segerlind. Exponential Lower Bounds and Integrality Gaps for Tree-Like Lovász-Schrijver Procedures. *SIAM J. Comput.*, 41(1):128–159, 2012.
21. H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Disc. Math.*, 3(3), 1990.
22. H.D. Sherali and W.P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411–430, 1990.
23. J. Shoesmith and T. J. Smiley. *Multiple-Conclusion Logic*. Cambridge, 1978.
24. T. Studer. On the proof theory of the modal mu-calculus. *Studia Logica*, 89(3), 2008.
25. M. Vinyals. Personal communication, 2018.