Conditional Autarkies: Hard Formulas Made Easy

Ilario Bonacina
Universitat Politècnica de Catalunya, Barcelona, Spain

Maria Luisa Bonet
Universitat Politècnica de Catalunya, Barcelona, Spain

Antonina Kolokolova
Memorial University of Newfoundland, St. Johns', Canada

Massimo Lauria
Università di Roma La Sapienza, Italy

Abstract

State-of-the-art SAT solvers increasingly use techniques beyond resolution. For instance, adding redundant clauses allows the solver to reduce the solution space, *e.g.*, to break symmetries. We investigate the strength of relatively weak redundancy reasoning: conditional autarkies and set-blocked clauses, with no new variables and no deletions.

We show that adding conditional autarkies (as set-blocked clauses) on top of resolution allows efficient refutations of a number of natural combinatorial principles that may occur in SAT benchmarks. In particular, we give efficient proofs of the perfect matching on a grid, the mutilated chessboard, the counting principle modulo 3, and the relativized pigeonhole principle.

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Automated reasoning

Keywords and phrases conditional autarkies, perfect matching principle, pigeonhole principle, redundancy rules, proof complexity

Digital Object Identifier 10.4230/LIPIcs.SAT.2026.8

Funding *Ilario Bonacina*: Funded by the AEI with the grant number PID2022-138506NB-C22.
Maria Luisa Bonet: Funded by the AEI with the grant number PID2022-138506NB-C21.
Antonina Kolokolova: Supported by an NSERC Discovery grant.
Massimo Lauria: Supported by project "Logical Methods in Combinatorics" N. 2022BXH4R5 of the Italian Ministry of University and Research (MIUR).

1 Introduction

Satisfiability testing is at the core of automated reasoning. The simple language of Boolean logic makes encoding problems into formula satisfiability a viable strategy to solve them. This motivates a fast development of algorithms for satisfiability. As a result, nowadays, problems in formal verification, security, planning and classical mathematics, are regularly solved in this fashion.

Conflict-driven clause learning (CDCL) [21] is the dominant approach to deciding propositional satisfiability (SAT). Starting from a partial assignment, a CDCL solver repeatedly applies unit propagation. If propagation yields a conflict (*i.e.*, a clause is falsified), it analyzes the conflict and learns a clause that blocks the same combination of decisions, then backjumps to an earlier decision level and continues. If the learned clause is empty, the formula is unsatisfiable. If propagation ends without conflict and all variables are assigned, the formula is satisfiable; otherwise the solver assigns another variable according to some decision heuristic, and repeats the process.

It is well-known that CDCL solvers have the same power as resolution proofs [22, 2]. This means that CDCL solvers suffer from the same limitations as the resolution proof system. For instance, there are seemingly simple formula families that require exponential size resolution proofs, implying that solving these formulas with CDCL takes exponential time. A large number of of principles are hard for resolution, and therefore also for CDCL: the pigeonhole principle, the mutilated chessboard principle, the clique-coloring principle, perfect matching, Tseitin, and many others [5].

Given the limitations of Resolution, people have been trying to improve/extend SAT solvers to harness the power of stronger proof systems. One way is to allow the solver to add *redundant clauses*, namely clauses that are not necessarily logical consequences of the formula, yet they reduce the search space of satisfying assignments. Concretely, a satisfiable formula remains satisfiable after the addition of a redundant clause, but possibly by fewer assignments. A new generation of SAT solvers [11] allows the addition of *propagation redundant* clauses, a type of clauses for which the redundancy is easy to certify [11]. The corresponding proof system called *propagation redundancy* (PR) models these solvers.

A common way of adding redundancy is via partial assignments that satisfy all clauses they affect. Given a propositional formula in conjunctive normal form, an *autarky* is a truth assignment that satisfies every clause it touches. A *conditional autarky* [26, 18] is a partial assignment divided into two parts, the *conditional part* and the *autarky part*, where the autarky part must be an autarky for the formula restricted by the conditional part. From conditional autarkies we obtain redundant clauses called *set-blocked clauses*, which are a special case of propagation redundant clauses, hence they are allowed in PR proofs.

To find redundant clauses, Heule et al. [12] introduced Satisfaction-Driven Clause Learning (SDCL), a SAT solving technique that extends CDCL. If the current assignment plus unit propagation does not lead to a conflict, we do not obtain a new variable assignment as CDCL would do. Instead, we first try to prune the search space by learning a propagation redundant clause. This is done by another SAT solver call that tries to produce a conditional autarky assignment, from which to obtain propagation redundant clauses. This SDCL system allows short proofs of the pigeonhole formulas [11], circumventing the limitations of pure CDCL-based solvers.

Redundancy learning is often associated with "*without loss of generality*" reasoning [23], most notably symmetry breaking: adding redundant constraints can restrict attention to canonical representatives of symmetric solutions without changing satisfiability. While this is an important use case of redundancy, our upper bounds highlight a different phenomenon: even in the absence of large symmetry groups, local redundancy rules based on conditional autarkies can have a dramatic effect. Concretely, the proofs we give show that proof system SBC^- (*i.e.*, resolution plus set-blocked clauses) can efficiently refute natural combinatorial principles that are hard for resolution, and have very few symmetries.[1] This suggests that SAT solvers with SDCL-like redundancy learning may succeed on such structured instances as well.

Finally, our work is in the line of recent proof-complexity studies of redundancy systems. Buss and Thapen [6] studied redundancy proof systems without new variables, in particular showing that *Set Propagation Redundancy* (SPR^-), a proof system simulating SBC^- (and strictly stronger), can prove, polynomially, many principles hard for resolution. Yolcu and Heule studied the strength of SBC^- [28], for instance giving polynomial upper bounds on the Pigeonhole Principle.

[1] Following [6], we use the superscript $^-$, as in SBC^- or PR^-, when the system is not allowed to introduce new variables. With new variables both SBC and PR become equivalent to Extended Resolution.

The present paper pushes this line further and shows that the conditional autarky/set-blocked clause viewpoint yields proofs for principles that are, a priori, not obviously susceptible to weak redundancy reasoning. For instance, we show polynomial SBC$^-$ refutations for principles with very few symmetries.

Recall that SBC$^-$ is the proof system allowing the introduction of set-blocked clauses on top of the resolution rules, but without deletions and without new variables.

Summary of results. In this work we show that several variants of the pigeonhole principle (with the usual propositional encoding) and perfect matching have polynomial size SBC$^-$ refutations. In particular, we show that the perfect matching principle on an $n \times n$ grid graph has $\mathcal{O}(n^3)$ refutations in the proof system SBC$^-$ (Theorem 3.3). If we consider Frege refutations where the formulas in the proof are a depth d in the basis given by $\wedge$, $\vee$, and $\neg$, then the principle above has polynomial size refutations when $d = \Theta(\log n)$ [7], while it does not admit polynomial refutations when $d = \mathcal{O}(\log n / \log\log n)$ [13]. This, together with Theorem 3.3 shows that depth-$\mathcal{O}(\log n/\log\log n)$ Frege cannot p-simulate SBC$^-$.

A simple modification of the argument in Theorem 3.3 shows that the mutilated chessboard principle on an $n \times n$ chessboard also has $\mathcal{O}(n^3)$ refutations in SBC$^-$ (Theorem 3.5). This answers an open question from [25], and also improves on the empirical PR$^-$ upper bound of $\mathcal{O}(n^3)$ for the mutilated chessboard principle in [10], both by giving an explicit refutation and doing it in a weaker redundancy system.

The counting principle, a generalization of the matching principle, falsely claims that set $\{1, \ldots, kn+1\}$ can be partitioned into sets of size k. This principle is hard for bounded depth Frege refutations [1, 4]. In Theorem 3.2 we show an SBC$^-$ refutation for $k = 3$ of length $\mathcal{O}(n^7)$.

The last formula for which we show a small refutation is the relativized pigeonhole principle. The concept of relativization was introduced by Krajíček [19] as a systematic way to increase the hardness of combinatorial principles. For instance, some semi-algebraic proofs systems with polynomial-size refutations of the pigeonhole principle do not have polynomial size refutations of the (harder) relativized pigeonhole principle [3]. In Theorem 4.1 we show that even the relativized version of the pigeonhole principle formula has polynomial size refutations in SBC$^-$. This is interesting in particular because it seems difficult to obtain short PR$^-$ proofs of it in practice [24, 25].

Structure of the paper. The rest of this work is structured as follows. Section 2 contains all the preliminaries needed, in particular it describes the equivalence between conditional autarkies and set-blocked clauses. Section 3 contains SBC$^-$ upper bounds for perfect matching principle on some graphs (the complete bipartite, the square grid and the mutilated chessboard graph), and for the counting principle modulo 3. In Section 4 we switch the focus to the relativized pigeonhole principle and show it has polynomial size SBC$^-$ refutations. Finally, Section 5 concludes the paper with some final remarks and open problems.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n]$ be the set $\{1, \ldots, n\}$. A *Boolean variable* x takes values in $\{0, 1\}$ and a *literal* is either a variable x or its negation $\overline{x}$. A *clause* is a disjunction of literals, and the empty (false) clause is $\bot$. A formula φ in *Conjunctive Normal Form* (CNF) is a conjunction of clauses. We identify a CNF as the set of its clauses, and a clause as the set of its literals. A clause is a *tautology* if it contains both literals x and $\overline{x}$ for some variable x.

Semantics. A *partial assignment* on a set of variables X is a partial function $\sigma\colon X \to \{0,1\}$. If σ is total, we say simply that σ is an *assignment*. Partial assignments map CNF formulas φ into CNF formulas $\varphi\restriction_\sigma$ in the natural way: $x\restriction_\sigma = x$ and $\overline{x}\restriction_\sigma = \overline{x}$ when $x \notin \mathrm{dom}(\sigma)$, $(\psi \wedge C)\restriction_\sigma = \psi\restriction_\sigma \wedge C\restriction_\sigma$, $(C \vee \ell)\restriction_\sigma = C\restriction_\sigma \vee \ell\restriction_\sigma$, and disjunctions/conjunctions are simplified using the usual logic rules, *i.e.*, $D \vee 0 = D$, $D \vee 1 = 1$, $F \wedge 1 = F$, $F \wedge 0 = 0$. We say that σ *satisfies* a clause C ($\sigma \vDash C$) if $C\restriction_\sigma$ is either 1 or tautological. We say that σ *satisfies* a CNF formula φ ($\sigma \vDash \varphi$), when $\varphi\restriction_\sigma = 1$. We say that $\varphi \vDash C$ if for every assignment σ, if $\sigma \vDash \varphi$ then $\sigma \vDash C$. We identify a literal ℓ with the partial assignment that assigns ℓ to 1 and leaves all other variables unassigned, hence we use notation $\varphi\restriction_\ell$. Likewise, given a clauses C we denote as $\overline{C}$ the unique minimal partial assignment that maps all literals in C to false, and use the notation $\varphi\restriction_{\overline{C}}$.

Unit propagation. The unit propagation for a CNF φ is an iterative process. We start with an empty partial assignment α. At each iteration we look at $\varphi\restriction_\alpha$ and we have three possible outcomes. First, if $\varphi\restriction_\alpha$ contains $\bot$, then the process stops and we say that φ *implies the contradiction by unit propagation*, and we denote this as $\varphi \vdash_1 \bot$. Otherwise, if $\varphi\restriction_\alpha$ contains a unit clauses ℓ we set $\alpha := \alpha \cup \{\ell\}$ and we do another iteration. In the remaining case, when $\varphi\restriction_\alpha$ does not contain neither unit nor empty clauses, the process stops. We use the notation $\varphi \vdash_1 C$ to indicate that $\varphi \wedge \overline{C} \vdash_1 \bot$, and we say that φ implies C by *reverse unit propagation (RUP)*. We further generalize this notation indicating with $\varphi \vdash_1 \Delta$ the fact that $\varphi \vdash_1 C$ for every $C \in \Delta$.

Redundancy and equisatisfiability. If a clause C is a logical consequence of φ, then φ and $\varphi \cup \{C\}$ are *equisatisfiable*, *i.e.*, if φ is satisfiable then $\varphi \cup \{C\}$ is also satisfiable, and vice versa. SAT solvers leverage this observation, and add new clauses to a CNF formula while trying to find some satisfying assignment for it. The new clauses rein the search process, and being logical consequences, do not exclude any potential solution. To further improve the search, solvers identify clauses that, despite not being logical consequences, do not change the satisfiability of the formula if added to (or removed from) it. In particular this can be done either before (see for example [8]) or during the solving process [14].

▶ **Definition 2.1.** *A clause C is* redundant *for φ when φ and $\varphi \cup \{C\}$ are equisatisfiable.*

A solver could add redundant clauses, to further guide the search, but could also remove redundant clauses to make the formula smaller, in order to speed up unit propagation or to delete variables. This is often what preprocessing and inprocessing phases in SAT solvers do. Notice that the property of redundancy is not monotone: for instance, both x_1 and x_2 are redundant for $\{x_1 \vee x_2, \overline{x}_1 \vee \overline{x}_2\}$, but if either is added to the formula, the other is not redundant anymore.

2.1 Autarkies and conditional autarkies

Finding redundant clauses to add is a major challenge. Even testing that a particular clause is redundant requires a SAT solver to call itself (this is indeed the approach of SDCL-based solvers [12]). Nevertheless, there are examples of redundant clauses that are easy to recognize. For example, when a literal ℓ occurs in φ and its opposite $\overline{\ell}$ does not, we say ℓ is a *pure literal*, and ℓ is redundant for φ. This idea leads to the concept of *autarky*.

▶ **Definition 2.2** (autarky)**.** *An* autarky *for a CNF formula φ is a partial assignment σ so that for each clause $C \in \varphi$, either $C\restriction_\sigma = 1$ or $C\restriction_\sigma = C$.*

From the definition it is immediate that the unit clauses inducing assignment σ are all redundant, when added in any order. Pure literals are special cases of autarkies, as it is any satisfying assignment. We should not expect formulas to have many autarkies, but it is possible to study *conditional autarkies*, meaning autarkies that only apply after a restriction.

▶ **Definition 2.3** (conditional autarky [26, 18]). *A* conditional autarky *for a CNF formula φ is an implication $\alpha \to \beta$ where α and β are two partial assignments on the variables of φ with $\mathrm{dom}(\alpha) \cap \mathrm{dom}(\beta) = \emptyset$, so that β is an autarky for $\varphi\restriction_\alpha$.*

Both autarkies and conditional autarkies are easy to recognize, and induce redundant clauses. The following is a characterization of conditional autarkies we will use.

▶ **Observation 2.4.** *An implication $\alpha \to \beta$ with $\mathrm{dom}(\alpha) \cap \mathrm{dom}(\beta) = \emptyset$ is a conditional autarky for a CNF formula φ if and only if for every clause $C \in \varphi$, $\alpha(C) = 1$ or $\beta(C) = 1$ or $\beta(C) = C$.*

▶ **Lemma 2.5** ([15]). *If $\bigwedge_{i\in I} a_i \to \bigwedge_{j\in J} b_j$ is a conditional autarky for φ, then φ and $\varphi \cup \{\bigvee_{i\in I} \overline{a}_i \vee b_j \mid j \in J\}$ are equisatisfiable. In particular, φ and $\varphi \cup \{\bigvee_{i\in I} \overline{a}_i \vee \bigvee_{j\in J} b_j\}$ are also equisatisfiable.*

In the next section, we recall proof systems (SBC$^-$ and PR$^-$) that allow the introduction of equisatisfiable clauses in addition to resolution inferences. The introduction of clauses coming from conditional autarkies (as in Lemma 2.5) is formalisable in the systems PR$^-$ and SBC$^-$. The addition of clauses $\{\bigvee_{i\in I} \overline{a}_i \vee \bigvee_{j\in J} b_j\}$ is done in the system SBC$^-$, and the addition of $\{\bigvee_{i\in I} \overline{a}_i \vee b_j \mid j \in J\}$ is done in PR$^-$. Notice that the set of clauses added in PR$^-$ is stronger than the single clause added in SBC$^-$. When used in practice, conditional autarkies are introduced as PR$^-$ clauses, since they are smaller and can be further minimized [12].

In this paper we focus on the strength of the more limited system SBC$^-$. This system and PR$^-$ are defined formally in the next section, but indeed all the redundant clauses that can be added in SBC$^-$ come from conditional autarkies (Proposition 2.7), therefore the next subsection is not strictly needed to understand sections 3 to 5.

2.2 From conditional autarkies to proof systems with redundancy

When modern SAT solvers claim that a formula is UNSAT, they produce (more or less explicitly) a refutation in some proof system of length proportional to the running time of the solver. The proof systems we consider in this paper match the following general scheme. Given a CNF formula φ with m clauses, a refutation is a sequence $D_1, D_2, \ldots, D_t$ of clauses on the variables of φ, so that the first m clauses are the clauses of φ itself, $D_t = \bot$, and for $i > m$ each D_i is added to the sequence according to some rule so that D_i is redundant for $\{D_1, \ldots D_{i-1}\}$. The proof must be verifiable efficiently, hence there must be a way to witness efficiently the redundancy of each addition. For our discussion the order in which we enumerate the clauses of φ at the beginning of the refutation is irrelevant.

Resolution. Resolution is the most important proof system in literature and it is at the core of all CDCL solvers [21]. It is based on the following simple observation: two clauses $A \vee x$ and $B \vee \overline{x}$ logically imply the *resolvent* $A \vee B$. Since logically implied clauses are redundant, resolution refutations are an instantiation of the general scheme above, where each D_i is the resolvent of two clauses among $\{D_1, \ldots D_{i-1}\}$.

SBC refutations. An *SBC refutation* is an instantiation of the general scheme above, where each clause D_i is either the resolvent of two preceding clauses, or is a set-blocked clause w.r.t. all the preceding clauses. A clause C is a *set-blocked clause* w.r.t. a set of clauses Γ when C can be partitioned as $A \vee B$ with non empty $B = \{\ell_1, \ldots, \ell_t\}$, so that for every $D \in \Gamma$ with $D \cap B = \emptyset$ and some $\overline{\ell}_i \in D$, it must hold that $A \cup (D \setminus \{\overline{\ell}_1, \ldots, \overline{\ell}_t\})$ is a tautology [17]. It is intended that each set-blocked clause is annotated with its partition, in order to make the SBC refutation efficiently checkable.

When SBC refutations are allowed to introduce new variables, the system becomes equivalent to Extended Resolution (aka Extended Frege). Following [6], SBC$^-$ is the fragment of SBC with no addition of new variables, and this is the proof system our paper focuses on.

▶ Remark 2.6. Regarding the terminology, the paper [17] defines *set-blocked*, *semantically blocked* and *super-blocked* clauses. Semantically-blocked and super-blocked are the same concept and are a stronger notion which is not even certifiable with a witness. They are still "local" in some precise sense. The *globally blocked* clauses, defined later in [15, Definition 14], are the ones obtained from a conditional autarky $\alpha \to b_1 b_2 \cdots b_t$ by adding the smaller clauses $\overline{\alpha} \vee b_i$. They are not set-blocked, but are propagation redundant (see the definition later).

The notion of set-blocked clause looks quite artificial, but its redundancy becomes clear once we observe that it is closely related to the one of conditional autarky. Due to this connection, in the rest of the paper, we will always identify set-blocked clauses and conditional autarkies.

▶ **Proposition 2.7** ([15]). *If the implication $\bigwedge_i a_i \to \bigwedge_j b_j$ is a conditional autarky for φ, then the clause $\bigvee_i \overline{a}_i \vee \bigvee_j b_j$ is a set-blocked clause for φ. If C, partitioned as $\bigvee_i \overline{a}_i \vee \bigvee_j b_j$, is a set-blocked clause for φ then $\bigwedge_i a_i \to \bigwedge_j b_j$ is a conditional autarky for φ.*

▶ **Observation 2.8.** *Without loss of generality we can assume that in any SBC refutation, all set-blocked clauses are added before all resolvents.*

PR refutations. A PR *refutation* [11] is an instantiation of our general scheme, where each clause D_i is either the resolvent of two preceding clauses, or is a propagation redundant clause w.r.t. all the preceding clauses. A clause C is *propagation redundant* w.r.t. a set of clauses Γ if there is an assignment σ such that $\Gamma|_{\overline{C}} \vdash_1 \Gamma|_\sigma$ and $C|_\sigma = 1$. It is intended that each such clause is annotated with its witnessing assignment, in order to make the PR refutation efficiently checkable. Following [6], PR$^-$ is the fragment of PR refutations that does not allow new variables. Similarly as for SBC, when PR refutations are allowed to introduce new variables the system becomes equivalent to Extended Resolution.

If $\bigwedge_{i \in I} a_i \to \bigwedge_{j \in J} b_j$ is a conditional autarky for φ the clauses $\{\bigvee_{i \in I} \overline{a}_i \vee b_j \mid j \in J\}$ are not set-blocked clauses, but they can be added one by one as propagation redundant clauses.

▶ **Lemma 2.9** ([15]). *If $\bigwedge_{i \in I} a_i \to \bigwedge_{j \in J} b_j$ is a conditional autarky for φ, then the set of clauses $\varphi \cup \{\bigvee_{i \in I} \overline{a}_i \vee b_j \mid j \in J\}$ can be obtained from φ adding one by one the clauses in $\{\bigvee_{i \in I} \overline{a}_i \vee b_j \mid j \in J\}$ as propagation redundant clauses w.r.t. the clauses obtained before.*

How the systems compare to each other. Any resolution refutation is also a SBC$^-$ refutation (by definition), likewise SBC$^-$ refutations are also PR$^-$ refutations. To see this compare Proposition 2.7 and Lemma 2.9: the witness for the clauses in the latter lemma is the same as the one for the corresponding set-blocked clause in the former. Moreover, SBC$^-$ sometimes allows exponentially shorter refutations than resolution (see for example [28]),

and PR$^-$ sometimes allows exponentially shorter refutations than SBC$^-$ [27, 6]. Since SBC$^-$ is weaker than PR$^-$, clause constructions such as the ones in section 3 are more involved. Notice that all systems using redundancy become more powerful if they are allowed to delete clauses [16, 6], for this reason we focus on upper bounds for a weaker system (SBC$^-$) without deletions.

3 Matching and counting principles

Given a graph $\mathcal{G} = (V, E)$, consider Boolean variables p_e for each edge in $e \in E$. The *perfect matching principle* PM($\mathcal{G}$) states that $\mathcal{G}$ has a perfect matching covering all vertices, and hence is unsatisfiable whenever $\mathcal{G}$ does not admit a perfect matching. It is the conjunction of the following clauses

$$\text{(Totality axioms)} \qquad \bigvee_{e \ni v} p_e \qquad \text{for every } v \in V, \tag{1}$$

$$\text{(1-to-1 axioms)} \qquad \overline{p}_e \vee \overline{p}_{e'} \qquad \text{for every } e, e' \in E \text{ s.t. } e \cap e' \neq \emptyset. \tag{2}$$

Natural generalizations of the perfect matching on a complete graph with an odd number of vertices are *Counting Principles* (Count$_n^k$) for $k \in \mathbb{N}$. Formula Count$_n^k$ states that the set $[n]$ can be partitioned into sets of size k, hence if k does not divide n, it is unsatisfiable. The formula uses a Boolean variable x_S for each subset $S \subseteq [n]$ of size k, and is the conjunction of the following clauses

$$\text{(Totality axioms)} \qquad \bigvee_{S \ni v} x_S \qquad \text{for every } v \in [n], \tag{3}$$

$$\text{(1-to-1 axioms)} \qquad \overline{x}_S \vee \overline{x}_{S'} \qquad \text{for every } S, S' \text{ s.t. } S \cap S' \neq \emptyset. \tag{4}$$

In this section, we prove that SBC$^-$ allows polynomial size refutations of the following principles:

- PM($\mathcal{K}_{n+1,n}$), where $\mathcal{K}_{n+1,n}$ is the complete bipartite graph between $[n+1]$ and $[n]$ (Theorem 3.1);
- Count$_{3n+1}^3$ (Theorem 3.2);
- PM($\mathcal{G}_{2n+1,2n+1}$), the grid graph with $(2n+1) \times (2n+1)$ vertices (Theorem 3.3);
- PM($\mathcal{M}_{2n,2n}$), the graph associated with the mutilated chessboard, i.e. a grid graph $2n \times 2n$ with the top-left and the bottom-right vertices removed (Theorem 3.5).

The intuition for the SBC$^-$ refutation of all these formulas is the same, and for the sake of clarity we formulate it in the language of matching. There are many possible partial matching in a graph, but the introduction of certain set-blocked clauses can narrow down the focus of the proofs to a specific partial matching $e_1, e_2, e_3, \ldots$, "without loss of generality". When the proof eventually deduces the corresponding unit literals $p_{e_1}, p_{e_2}, p_{e_3}, \ldots$, the formula becomes trivial and immediate to refute. The proof introduces such set-blocked clauses in phases: at the ith phase we introduce various clauses

$$\bigwedge_{j<i} p_{e_j} \to p_{e_i} \vee D_1 \qquad \bigwedge_{j<i} p_{e_j} \to p_{e_i} \vee D_2 \qquad \ldots,$$

that later are used to deduce $\bigwedge_{j<i} p_{e_j} \to p_{e_i}$. The prefix $\bigwedge_{j<i} p_{e_j}$ is a subset of the conditional part, and this guarantees that the clauses of the ith phase are set-blocked clauses with respect to the clauses from previous phases. Once all clauses $\bigwedge_{j<i} p_{e_j} \to p_{e_i}$ are available, the proof easily obtains units $p_{e_1}, p_{e_2}, \ldots p_{e_i}, \ldots$ and then the empty clause.

Regarding the proof of Theorem 3.1, we note that the argument in [6, Theorem 4.3] shows a size upper bound for the Pigeonhole Principle PHP_n^{n+1} in SPR$^-$. That argument directly adapts to an SBC$^-$ upper bound for PM($\mathcal{K}_{n+1,n}$). Nevertheless, for this latter principle we give a simpler proof that also prepares for the other results of this section.

▶ **Theorem 3.1.** PM($\mathcal{K}_{n+1,n}$) *has* SBC$^-$ *refutations with* $\mathcal{O}(n^3)$ *clauses.*

Proof. The idea of the proof is to enforce the partial matching $(1,1),(2,2),\dots,(n-1,n-1)$.

For $\{i,j\} \in E(\mathcal{K}_{n+1,n})$, we denote the variable $p_{\{i,j\}}$ simply as $p_{i,j}$. Following Observation 2.8 we introduce all set-blocked clauses as conditional autarkies at the beginning and then apply resolution rules. We add the clauses

$$C_{i,j,k} = \overbrace{\bigvee_{\ell=1}^{i-1} \overline{p}_{\ell,\ell} \vee \bigvee_{\ell \neq i,j} (p_{\ell,k} \vee p_{j,\ell} \vee p_{\ell,i} \vee p_{i,\ell})}^{\text{conditional part}} \vee \overbrace{p_{i,i} \vee \overline{p}_{j,i} \vee \overline{p}_{i,k} \vee p_{j,k}}^{\text{autarky part}} \tag{5}$$

for each $i \in [n-1]$ and each $j \in [n+1], k \in [n]$ and $j,k > i$. See Figure 1 for a visual representation of one such clause. Intuitively, the conditional part of $C_{i,j,k}$ assumes that matching $(1,1),(2,2),\dots,(i-1,i-1)$ is already in place, and that vertices $\{i,j\}$ on one side can only be matched with $\{i,k\}$ on the other side. Under these assumptions, the autarky part enforces the choice of edges (i,i) and (j,k).

We add the conditional autarkies $C_{i,j,k}$ in a specific order: first all clauses of the form $C_{1,j,k}$, then all clauses of the form $C_{2,j,k}$, etc. Given a particular i the order of the $C_{i,j,k}$ clauses does not matter.

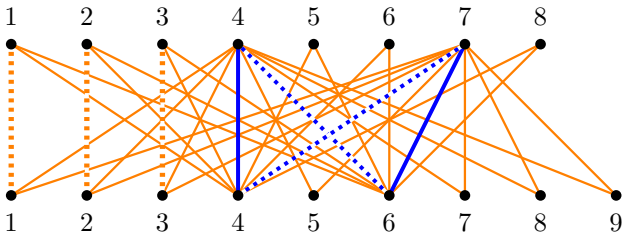


Figure 1 A visual representation of the clause $C_{4,6,7}$ in $\mathcal{K}_{9,8}$. An edge e is dotted if the clause contains p_e negatively, it is solid if it contains p_e positively. The edge is **orange** if the corresponding literal is in the conditional part of the clause, while it is **blue** if it is in the autarky part of the clause.

The assignment $\alpha_{i,j,k}$ corresponding to the conditional part of $C_{i,j,k}$ is $p_{i',i'} = 1$ for all $i' < i$ and $p_{\ell,k} = p_{k,\ell} = p_{\ell,i} = p_{i,\ell} = 0$ for all $\ell \neq i,j$. The assignment $\beta_{i,j,k}$ corresponding to the autarky part of $C_{i,j,k}$ is $p_{i,i} = 1, p_{j,i} = 0, p_{i,k} = 0, p_{j,k} = 1$.

Now consider all clauses in PM($\mathcal{K}_{n+1,n}$) $\cup \{C_{i',j',k'} \mid i' \leq i \wedge j', k' > i'\}$. The assignment $\beta_{i,j,k}$ only touches clauses containing the variables $p_{i,i}, p_{j,i}, p_{i,k}, p_{j,k}$. Most of these clauses, including the $C_{i',j,k}$, are satisfied by $\alpha_{i,j,k}$. The only clauses that survive $\alpha_{i,j,k}$ that are touched by $\beta_{i,j,k}$ are the totality on $i \in [n+1]$, $i \in [n]$, $j \in [n+1]$, $k \in [n]$, and the 1-to-1 axioms $\overline{p}_{i,i} \vee \overline{p}_{i,k}$, $\overline{p}_{i,i} \vee \overline{p}_{j,i}$, $\overline{p}_{j,k} \vee \overline{p}_{j,i}$, $\overline{p}_{j,k} \vee \overline{p}_{i,k}$. All such clauses are satisfied by $\beta_{i,j,k}$, hence $\alpha_{i,j,k} \to \beta_{i,j,k}$ is a conditional autarky for PM($\mathcal{K}_{n+1,n}$) $\cup \{C_{i',j',k'} \mid i' \leq i \wedge j', k' > i'\}$, and we can add all clauses $C_{i,j,k}$ as set-blocked clauses.

Now we proceed with a resolution proof. We show, by induction on $i \in [n-1]$, that we can derive all clauses $p_{i,i}$. Suppose we derived all clauses $p_{1,1}, \ldots, p_{i-1,i-1}$. Using the 1-to-1 axioms we derive all clauses $\overline{p}_{i,\ell}$ and $\overline{p}_{\ell,i}$ for $\ell < i$. First we resolve away all positive literals from $C_{i,j,k}$ to get the clause

$$C'_{i,j,k} = \overline{p}_{1,1} \vee \overline{p}_{2,2} \vee \cdots \vee \overline{p}_{i-1,i-1} \vee \overline{p}_{i,k} \vee \overline{p}_{j,i} \ .$$

We do this applying resolution between $C_{i,j,k}$ and all the 1-to-1 axioms $\overline{p}_{\ell,\ell} \vee \overline{p}_{k,\ell}$, $\overline{p}_{\ell,\ell} \vee \overline{p}_{\ell,k}$, $\overline{p}_{\ell,\ell} \vee \overline{p}_{i,\ell}$, $\overline{p}_{\ell,\ell} \vee \overline{p}_{\ell,i}$ for $\ell < i$. Now cut $C'_{i,j,k}$ with $p_{1,1}, \ldots, p_{i-1,i-1}$ to get

$$C''_{i,j,k} = \overline{p}_{i,k} \vee \overline{p}_{j,i} \ .$$

Now cut $C''_{i,j,k}$ for all ks and js with the totality on $i \in [n+1]$ and the totality on $i \in [n]$, that is $\bigvee_{j\in[n]} p_{i,j}$ and $\bigvee_{j\in[n+1]} p_{j,i}$. We get

$$D_i = p_{i,i} \vee \bigvee_{\ell<i} (p_{i,\ell} \vee p_{\ell,i}) \ .$$

Finally cutting D_i with all $\overline{p}_{\ell,i}, \overline{p}_{i,\ell}$ for $\ell < i$ we get $p_{i,i}$. Once we have derived all $p_{1,1}, \ldots, p_{n-1,n-1}$, by unit propagation on the formula $\mathrm{PM}(\mathcal{K}_{n+1,n})$ we get a formula equivalent to $\mathrm{PM}(\mathcal{K}_{2,1})$ which is clearly unsatisfiable (with a constant size resolution refutation). ◀

Notice that the argument above would work almost without changes to prove an $\mathcal{O}(n^3)$ upper bound in SBC^- for $\mathrm{PM}(\mathcal{K}_{2n+1})$. Instead of doing this we show how to modify the argument for the slightly more complicated upper bound for Count^3_{3n+1}.

▶ **Theorem 3.2** (Counting modulo 3). *Count^3_{3n+1} has refutations in SBC^- of size $\mathcal{O}(n^7)$.*

Proof. We denote the variable $x_{\{i,j,k\}}$ simply as $x_{i,j,k}$. Following Observation 2.8, we introduce all set-blocked clauses as conditional autarkies at the beginning, and then apply resolution rules. The idea of the proof is to enforce the partition $\{1,2,3\}, \{4,5,6\}, \ldots, \{3n-5, 3n-4, 3n-3\}$ so that the formula becomes trivial to refute. For each $0 \le i < n$, we denote $\{3i+1, 3i+2, 3i+3\}$ as S_i.

For each $i \le n-2$, and any two $J = \{j_1, j_2, j_3\}$ and $K = \{k_1, k_2, k_3\}$ that are disjoint subsets of $\{3i+4, \ldots, 3n+1\}$, we add set-blocked clauses

$$C_{i,J,K} = \overbrace{\bigvee_{\ell=0}^{i-1} \overline{x}_{S_\ell} \vee \bigvee_{\substack{T\cap(S_i\cup J\cup K)\neq\emptyset \\ T\neq S_i, T\neq J, T\neq K \\ T\neq\{3i+1,j_1,k_1\} \\ T\neq\{3i+2,j_2,k_2\} \\ T\neq\{3i+3,j_3,k_3\}} x_T}^{\text{conditional part}} \vee \overbrace{x_{S_i} \vee \overline{x}_{3i+1,j_1,k_1} \vee \overline{x}_{3i+2,j_2,k_2} \vee \overline{x}_{3i+3,j_3,k_3} \vee x_J \vee x_K}^{\text{autarky part}}$$

$$C'_{i,J} = \overbrace{\bigvee_{\ell=0}^{i-1} \overline{x}_{S_\ell} \vee \bigvee_{\substack{T\cap(S_i\cup J)\neq\emptyset \\ T\neq S_i, T\neq J \\ T\neq\{3i+1,3i+2,j_1\} \\ T\neq\{3i+3,j_2,j_3\}} x_T}^{\text{conditional part}} \vee \overbrace{x_{S_i} \vee \overline{x}_{3i+1,3i+2,j_1} \vee \overline{x}_{3i+3,j_2,j_3} \vee x_J}^{\text{autarky part}}$$

$$C''_{i,J} = \overbrace{\bigvee_{\ell=0}^{i-1} \overline{x}_{S_\ell} \vee \bigvee_{\substack{T\cap(S_i\cup J)\neq\emptyset \\ T\neq S_i, T\neq J \\ T\neq\{3i+1,3i+3,j_1\} \\ T\neq\{3i+2,j_2,j_3\}} x_T}^{\text{conditional part}} \vee \overbrace{x_{S_i} \vee \overline{x}_{3i+1,3i+3,j_1} \vee \overline{x}_{3i+2,j_2,j_3} \vee x_J}^{\text{autarky part}}$$

$$C'''_{i,J} = \overbrace{\bigvee_{\ell=0}^{i-1} \overline{x}_{S_\ell} \vee \bigvee_{\substack{T\cap(S_i\cup J)\neq\emptyset \\ T\neq S_i, T\neq J \\ T\neq\{3i+2,3i+3,j_1\} \\ T\neq\{3i+1,j_2,j_3\}} x_T}^{\text{conditional part}} \vee \overbrace{x_{S_i} \vee \overline{x}_{3i+2,3i+3,j_1} \vee \overline{x}_{3i+1,j_2,j_3} \vee x_J}^{\text{autarky part}} .$$

See Figure 2 for a visual representation of the autarky parts of the clauses above. The conditional part assumes that sets $\{1,2,3\},\ldots,\{3i-2,3i-1,3i\}$ have been chosen, and furthermore that all but two specific ways to cover $\{3i+1,3i+2,3i+3\}$ have been excluded. For example, the conditional part of $C_{i,J,K}$ only allows the solid and the dotted patterns from Figure 2(a) and the autarky part enforces the choice of the former.

With an argument analogous to Theorem 3.1, it is easy to see that the clauses $C_{i,J,K}$, $C'_{i,J}$, $C''_{i,J}$, $C'''_{i,J}$ can be added in any order that is non-decreasing with respect to i.

Again, as in the proof of Theorem 3.1, for the resolution part we assume by induction that we have deduced $x_{S_0},\ldots,x_{S_{i-1}}$, and now our goal is to deduce x_{S_i}. Using these assumptions, we resolve away the disjunction $\bigvee_{\ell=0}^{i-1}\overline{x}_{S_\ell}$ from the conditional autarkies, and then we remove all their positive literals using the 1-to-1 axioms. This is possible since by construction the set corresponding to each such positive literal intersects the set corresponding to one of the negative literals. In the end we get the following clauses.

$$D_{i,J,K} = \overline{x}_{3i+1,j_1,k_1} \vee \overline{x}_{3i+2,j_2,k_2} \vee \overline{x}_{3i+3,j_3,k_3} \tag{6}$$

$$D'_{i,J} = \overline{x}_{3i+1,3i+2,j_1} \vee \overline{x}_{3i+3,j_2,j_3} \tag{7}$$

$$D''_{i,J} = \overline{x}_{3i+1,3i+3,j_1} \vee \overline{x}_{3i+2,j_2,j_3} \tag{8}$$

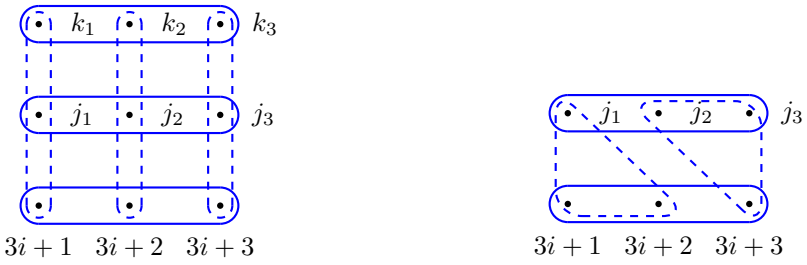
$$D'''_{i,J} = \overline{x}_{3i+2,3i+3,j_1} \vee \overline{x}_{3i+1,j_2,j_3} \tag{9}$$

Let Γ be the set of clauses of Count^3_{3n+1} plus the assumptions $x_{S_0},\ldots,x_{S_{i-1}}$, plus clauses (6)–(9). To show how to deduce x_{S_i} from Γ, we actually show how to refute $\Gamma\restriction_{\overline{x}_{S_i}}$ via a decision tree of size $\mathcal{O}(n^6)$. It is a well known fact that this translates into the deduction we wanted, preserving proof size.

The tree queries all variables x_A with $(3i+1)\in A$, except for x_{S_i}, until one of them is assigned 1. If they are all assigned 0 the totality axiom for $3i+1$ is violated. In the surviving branches the tree queries all x_B with $(3i+2)\in B$ until one of them is 1. Notice that some of these variables are skipped if they have been queried in the previous phase, and moreover this whole phase is skipped if some x_B was already set to 1 in the previous phase (this is possible when $3i+1\in B$). Again if all the answers are negative, the totality axiom for $3i+2$ is violated. Now we do the same for all x_C with $(3i+3)\in C$. Again, some are skipped when their values are known to be 0, and the whole phase could be skipped if $(3i+3)$ was already matched in a previous phase. For each surviving branch we can uniquely identify three sets A, B, C so that $x_A=1$, $x_B=1$, $x_C=1$, and so that $3i+1\in A$, $3i+2\in B$, $3i+3\in C$. There are at most n^6 of those triples.

We now analyze the leaves of the decision tree. If any of the three sets contains an element in $\{1,\ldots,3i\}$ then, because of the assumptions, a 1-to-1 axiom is violated. If any two of the three sets have non-empty intersection, yet they are distinct, they will violate some 1-to-1 axiom. It is impossible that $A=B=C$ because that would mean $A=S_i$ but x_{S_i} was set to 0. The remaining cases are when A,B,C are disjoint, and that falsifies (6), when $A=B$ disjoint from C, and that falsifies (7), when $A=C$ disjoint from B, and that falsifies (8), when $B=C$ disjoint from A, and that falsifies (9). This concludes the refutation of $\Gamma\restriction_{\overline{x}_{S_i}}$, hence of the induction step of proving x_{S_i}.

Once all $x_{S_0}, \dots x_{S_{n-2}}$ are proved, by unit propagation the formula reduces to Count_4^3 (up to renaming of variables) which is unsatisfiable and of constant size. ◀



(a) Autarky part of the clause $C_{i,J,K}$. **(b)** Autarky part of the clause $C'_{i,J}$.

Figure 2 Visual representations of the autarkies in the clauses $C_{i,J,K}$ and $C'_{i,J}$. A solid circle around points $\{i, j, k\}$ means the variable $x_{i,j,k}$ is positive, while a dashed circle means the variable is negative.

We now prove that the perfect matching on grids has polynomial size SBC$^-$ refutations. Unlike the previous upper bounds, the inductive argument is slightly more delicate.

▶ **Theorem 3.3** (Perfect matching on odd grids). $\text{PM}(\mathcal{G}_{2n+1,2n+1})$ *has refutations in* SBC$^-$ *of size* $\mathcal{O}(n^3)$*, where* $\mathcal{G}_{m,m}$ *denotes the* $m \times m$ *grid graph.*

Proof. Let the vertices of $\mathcal{G}_{2n+1,2n+1}$ be all the integer valued (x, y)s with $0 \leq x \leq 2n$ and $0 \leq y \leq 2n$. The bottom-left corner of the grid is the vertex $(0, 0)$ and the top-right corner the vertex $(2n, 2n)$. We consider the partial matching consisting of the $2n^2$ horizontal edges of the form $\{(2x, y), (2x + 1, y)\}$ with $x \in \{0, \dots, n - 1\}$ and $y \in \{0, \dots, 2n - 1\}$. See Figure 3c for an example of such matching in $\mathcal{G}_{7,7}$.

We fix the total ordering $e_1, \dots, e_{2n^2}$ on the edges in the matching from the bottom-left up filling the columns from left to right (see Figure 3). More precisely $e_i = \{(2x, y), (2x + 1, y)\}$ comes before $e_j = \{(2x', y'), (2x' + 1, y')\}$ either when $x < x'$ or when $x = x'$ and $y < y'$.

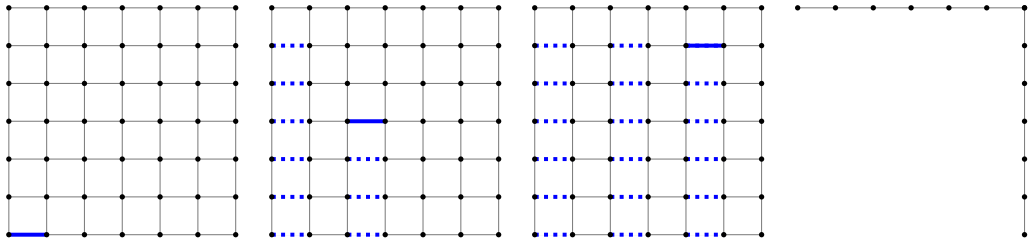
To ease notation, we also denote by e the Boolean variable corresponding to the edge e indicating whether that edge is in the matching or not. The main goal of the refutation is to obtain the clauses

$$e_1, \quad e_1 \to e_2, \quad \dots \quad \left(\bigwedge_{j<i} e_j\right) \to e_i, \quad \dots \quad \left(\bigwedge_{j<2n^2} e_j\right) \to e_{2n^2} \,. \tag{10}$$

For clarity, Figure 3a, 3b, and 3c show a pictorial representation of some of the clauses in (10) for the graph $\mathcal{G}_{7,7}$. Once we derived the clauses in (10), by unit propagation we obtain the clauses $e_1, e_2, \dots, e_{2n^2}$, and hence all the edges intersecting them are set to false. After these variables are set, the resulting formula reduces to the claim that there is a perfect matching in the path of odd length containing the vertices in the rightmost column and topmost row (see Figure 3d). This leads to an immediate contradiction, again just by unit propagation.

The general scheme of the proof is to first introduce some set-blocked clauses as conditional autarkies, then to use these clauses to infer via resolution the clauses in (10), and finally to conclude the proof via unit propagation as we discussed above.

Fix i, we now explain the conditional autarkies we use to derive $\bigwedge_{j<i} e_j \to e_i$. These clauses correspond to even cycles containing edge e_i. For each such cycle there are two possible matching, and the idea is to enforce the one containing e_i. To understand these clauses we refer to the visual representation in Figure 4.



(a) First clause. **(b)** Intermediate clause. **(c)** Last clause. **(d)** Residual graph.

Figure 3 The first three figures give a visual representation of the clauses e_1, $\bigwedge_{i=1}^{9} e_i \to e_{10}$, and $\bigwedge_{i=1}^{17} e_i \to e_{18}$ for the 7×7 grid. An edge is dotted if the clause contains the corresponding variable negatively, and it is solid if it contains it positively. The fourth picture show the resulting matching problem after the edges $e_1, \ldots, e_{18}$ edges are set to true, and the edges intersecting them are set to false.

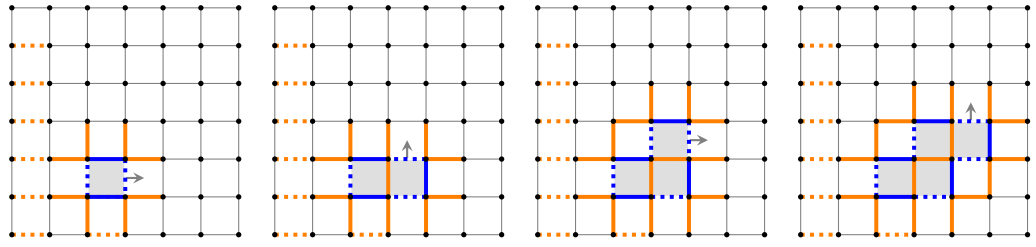
For any number $m \geq 1$, we describe an alternating cycle $\mathcal{C}_{i,m}$ with bottom-left edge e_i and that surrounds m squares. The cycle $\mathcal{C}_{i,1}$ has length 4, surrounding the square with bottom edge e_i. In general, if m is odd, $\mathcal{C}_{i,m+1}$ is the cycle surrounding the same squares of $\mathcal{C}_{i,m}$ plus the square on the right of the top-most square of $\mathcal{C}_{i,m}$. If m is even, $\mathcal{C}_{i,m+1}$ is the cycle surrounding the same squares of $\mathcal{C}_{i,m}$ plus the square on the top of the right-most square of $\mathcal{C}_{i,m}$ (the blue edges in Figure 4 are examples of theses cycles).

To each cycle $\mathcal{C}_{i,m}$ we associate a clause $D_{i,m}$ that we will prove is a conditional autarky w.r.t the initial axioms and the previously introduced $D_{i',m'}$. The conditional part of $D_{i,m}$ contains the negative literals $\neg e_1, \ldots, \neg e_{i-1}$ (the yellow dotted edges in Figure 4). Hence the condition assumes that these edges are chosen. The conditional part of $D_{i,m}$ also contains as positive literals all the edges incident to cycle $\mathcal{C}_{i,m}$ but not part it (the yellow solid edges in Figure 4). This part of the condition assumes that no other edge in the matching touches the vertices of the cycle. The autarky part of the clause $D_{i,m}$ contains the literals corresponding to the edges in the alternating cycle $\mathcal{C}_{i,m}$ with alternating polarity, starting with the edge e_i as a positive literal (the blue edges in Figure 4). The intended meaning is that under the condition the vertices in the cycles are matched using the solid blue edges.

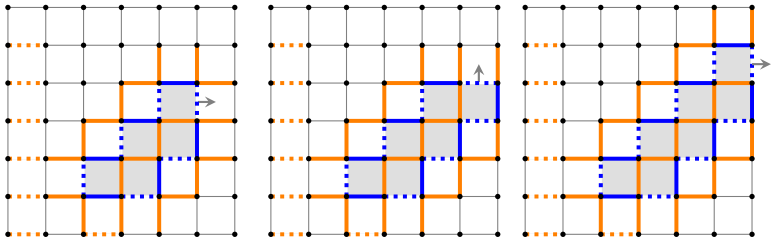
The clauses $D_{i,m}$ are set-blocked with respect to $\mathrm{PM}(\mathcal{G}_{2n+1,2n+1}) \cup \{D_{j,\ell} \mid j \leq i\}$. Indeed, the negation of the conditional part of the clause $D_{i,m}$ satisfies all the clauses in $\{D_{j,\ell} \mid j < i\}$ by setting e_js to 1 for $j < i$. The autarky part furthermore satisfies all $D_{i,\ell}$ by setting e_i to 1. To verify that $D_{i,m}$ is a conditional autarky for $\mathrm{PM}(\mathcal{G}_{2n+1,2n+1})$ observe that the conditional part disconnects cycle $\mathcal{C}_{i,m}$ and that the autarky part is a matching for it, leaving alone the rest of the graph. Therefore, we can add one by one all clauses $D_{i,m}$ as set-blocked clauses.

To conclude the argument, we show how to derive the clause $\left(\bigwedge_{j<i} e_j\right) \to e_i$ using all the clauses $D_{i,m}$. First, remove from $D_{i,m}$ all positive edges in the conditional autarky outside the cycle except for the ones adjacent to one of the e_js for $j < i$. We will remove these at most 4 literals at the end. The others can be removed resolving $D_{i,m}$ with the 1-to-1 axioms, since they are always adjacent to a negative edge in the cycle. Let $\widetilde{D}_{i,m}$ be the obtained clauses (see Figure 5a and Figure 5c). Let $e_{i,m}$ be the edge where we attach the new square to get $\mathcal{C}_{i,m+1}$ from $\mathcal{C}_{i,m}$ ($e_{i,m}$ is marked with a gray arrow in Figure 4 and Figure 5) and let $E_{i,m}$ be the clause which is the same as $\widetilde{D}_{i,m}$ but the literal $\overline{e}_{i,m}$ removed (see Figure 5b and Figure 5d).

First, notice we can obtain $E_{i,\ell}$ when ℓ corresponds to the longest possible cycle among all the $\mathcal{C}_{i,m}$: take the clause $\widetilde{D}_{i,\ell}$ and cut it with the totality axiom on the vertex in $e_{i,\ell} \cap e_{i,\ell-1}$. That is we cut with the totality axioms on one of the nodes of $e_{i,\ell}$, the one which does not introduce a new literal.

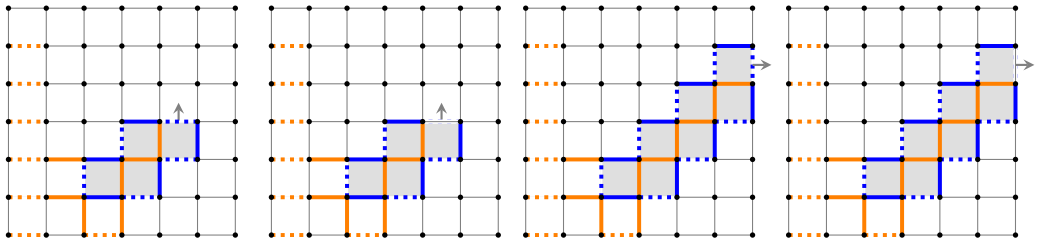


(a) The clause $D_{8,1}$. **(b)** The clause $D_{8,2}$. **(c)** The clause $D_{8,3}$. **(d)** The clause $D_{8,4}$.



(e) The clause $D_{8,5}$. **(f)** The clause $D_{8,6}$. **(g)** The clause $D_{8,7}$.

Figure 4 A visual representation of the clauses $D_{i,m}$. The pictures show $D_{8,m}$ for $m = 1, \ldots, 7$ in the 7×7 grid. The color of the edges follows the conventions of Figure 1. In gray we highlight the region of the grid surrounded by the blue alternating cycle $\mathcal{C}_{i,m}$. The gray arrow shows where the square of the cycle one square larger would be attached.



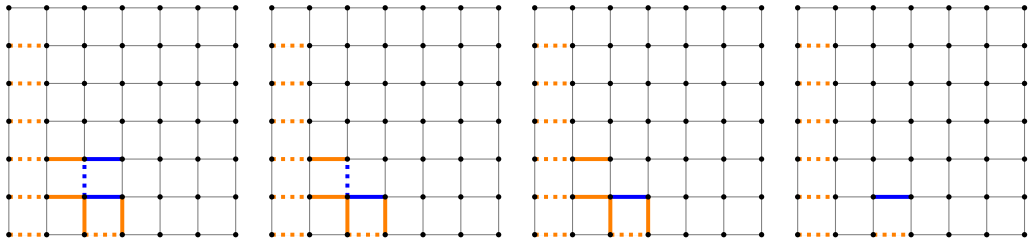
(a) The clause $\widetilde{D}_{8,4}$. **(b)** The clause $E_{8,4}$. **(c)** The clause $\widetilde{D}_{8,7}$. **(d)** The clause $E_{8,7}$.

Figure 5 A visual representation of the clauses $\widetilde{D}_{i,m}$ and $E_{i,m}$.

Now we show how to obtain $E_{i,m-1}$ from $E_{i,m}$ for each m between ℓ and 2: we just need to remove the rest of the edges of the last square of $E_{i,m}$. If m is odd, first resolve $E_{i,m}$ with the 1-to-1 axiom between the top and left edges to remove the top edge. Then, resolve with the totality axiom on the bottom-left corner (of the last square) to remove the vertical negative edge. Finally, resolve with $\widetilde{D}_{i,m-1}$ to remove $\overline{e}_{i,m-1}$. The remaining clause is $E_{i,m-1}$. If m is even, first resolve $E_{i,m}$ with the 1-to-1 axiom between the rightmost and bottom edges to remove the right-most edge. Then, resolve with the totality on the bottom-left corner (of the last square) to remove the horizontal negative edge. Finally, resolve with $\widetilde{D}_{i,m-1}$ to remove $\overline{e}_{i,m-1}$. The remaining clause is $E_{i,m-1}$.

To conclude the argument, we need to show how to derive $\left(\bigwedge_{j<i} e_j\right) \to e_i$ from $E_{i,1}$ (see Figure 6). First, resolve $E_{i,1}$ with the 1-to-1 axiom between the top and left edges to remove the top edge. Then, resolve with the totality axiom on the bottom-left corner to remove the vertical negative edge. Then, resolve with the 1-to-1 axioms using the previous negative e_js.

Regarding the overall size of the refutation, for each of the $2n^2$ edges e_i in the matching M we introduced $\mathcal{O}(n)$ clauses $D_{i,m}$, with an additional linear cost we derived $\left(\bigwedge_{j<i} e_j\right) \to e_i$, and the rest of the argument is unit propagation. The total size of the refutation is $\mathcal{O}(n^3)$. ◀



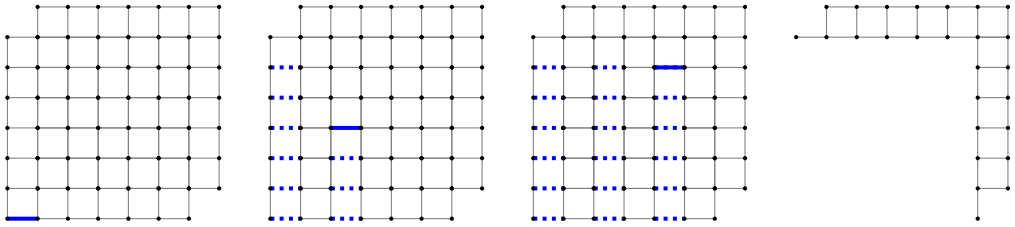
(a) $E_{8,1}$.

(b) Cut $E_{9,1}$ with 1-to-1 axiom.

(c) Cut the previous clause with totality axiom.

(d) Cut the previous clause with 1-to-1 axioms.

Figure 6 A visual representation of how to derive $\bigwedge_{i=1}^{7} e_i \to e_8$ from $E_{8,1}$.



(a) First clause.

(b) Intermediate clause.

(c) Last clause.

(d) Residual graph.

Figure 7 The sequence of clauses $\left(\bigwedge_{j<i} e_j\right) \to e_i$, similar to Figure 3.

Notice that SBC$^-$ cannot p-simulate depth-$\mathcal{O}(\log n)$ Frege, since SBC$^-$ does not have polynomial size refutations of the bit Pigeonhole principle [27], unlike depth-$\mathcal{O}(\log n)$ Frege. Interestingly, from Theorem 3.3 and the lower bound in [13] we get the following corollary.

▶ **Corollary 3.4.** *Depth-$\mathcal{O}(\log n/\log\log n)$ Frege does not p-simulate* SBC$^-$.

▶ **Theorem 3.5** (Mutilated chessboard)**.** *Let $\mathcal{M}_{m,m}$ be the $m \times m$ grid graph to which vertices $(0, m-1)$ and $(m-1, 0)$ have been removed.* PM($\mathcal{M}_{2n,2n}$) *has refutations in* SBC$^-$ *of size $\mathcal{O}(n^3)$.*

Proof sketch. The proof is an adaptation of the one for Theorem 3.3. Again, we prove a chain of clauses of the form $\left(\bigwedge_{j<i} e_j\right) \to e_i$ where $\{e_i\}_i$ is the sequence of horizontal edges $\{(2x, y),(2x+1, y)\}$ for $x \in \{0, \ldots, n-2\}$ and $y \in \{0, \ldots, 2n-3\}$ (see Figure 7). That is, now the matching we use stays two squares away from the top and right borders of the grid.

The set-blocked clauses introduced, and the way to obtain the clauses $\left(\bigwedge_{j<i} e_j\right) \to e_i$ is almost the same as in Theorem 3.3. Let e_a and e_b the topmost left edge and the bottommost right edge in the sequence, respectively. The first small difference with Theorem 3.3 is that clauses $D_{a,1}$, $D_{a,2}$, $D_{a,3}$ and $D_{b,2}$, $D_{b,3}$, $D_{b,4}$, $D_{b,5}$ miss the positive literals that correspond to the solid orange edges present in the original grid graph, but are missing in the mutilated corners. In the proof from Theorem 3.3 these literals were just resolved away using the 1-to-1 axioms. Here we just skip those steps. Another difference with Theorem 3.3 is the residual graph (see Figure 7d), but also in this case unit propagation ends the refutation. ◀

Notice that the upper bound in Theorem 3.3 is also an upper bound on the *onto-functional* version of the Pigeonhole principle on the grid, a sparse graph. In the next section we prove that even a harder version of PHP has short proofs in SBC$^-$.

4 The relativized pigeonhole principle

In its unsatisfiable CNF form, the pigeonhole principle claims that there is a way to make $n+1$ pigeons fly into n holes while avoiding any two pigeons to end up in the same hole. This formula is famously hard to refute for Resolution [9] and for BC$^-$ [20]. A short refutation can instead be obtained in PR, in SPR or even in SBC$^-$ [11, 6, 27]. Here we show an SBC$^-$ refutation for a harder variant.

The *relativized pigeonhole principle* is a variant of the pigeonhole principle where $n+1$ pigeons fly to $m > n$ intermediate resting places and then fly into n holes. The CNF encoding claims that there are no collisions, neither at the resting places nor at the holes. The formula itself is denoted as $\mathrm{RPHP}^{n+1}_{m,n}$, and its clauses are

$$\text{(Totality axioms 1)} \quad \bigvee_{a=1}^{m} p_{i,a} \qquad \text{for } i \in [n+1] \qquad (11)$$

$$\text{(Hole axioms 1)} \quad \overline{p}_{i,a} \vee \overline{p}_{j,a} \qquad \text{for } i,j \in [n+1], a \in [m] \text{ with } i \neq j \qquad (12)$$

$$\text{(Activation axioms)} \quad \overline{p}_{i,a} \vee r_a \qquad \text{for } i \in [n+1], a \in [m] \qquad (13)$$

$$\text{(Totality axioms 2)} \quad \overline{r}_a \vee \bigvee_{k=1}^{n} q_{a,k} \qquad \text{for } a \in [m] \qquad (14)$$

$$\text{(Hole axioms 2)} \quad \overline{r}_a \vee \overline{r}_b \vee \overline{q}_{a,k} \vee \overline{q}_{b,k} \qquad \text{for } a,b \in [m], k \in [n] \text{ with } a \neq b \qquad (15)$$

▶ **Theorem 4.1.** $\mathrm{RPHP}^{n+1}_{m,n}$ *has a refutation in* SBC$^-$ *of size* $\mathcal{O}(n^3m^3)$.

Proof. The intuition behind the argument is to forbid pigeon i to fly into hole $k < i$, regardless of the intermediate resting place. As usual we add all the conditional autarkies first and then do the resolution steps.

Adding the conditional autarkies. For $j,k > i$, we consider clauses $C^{i\to a\to i}_{j\to b\to k}$ that claim that when pigeons i and j fly to k and i, respectively through distinct resting places a and b, without loss of satisfiability we can assume that pigeon i flies to i and that pigeon j flies to k. Formally, the clause $C^{i\to a\to i}_{j\to b\to k}$ is

$$\overbrace{\overline{p}_{i,a} \vee \bigvee_{i'\neq i} p_{i',a} \vee \overline{p}_{j,b} \vee \bigvee_{j'\neq j} p_{j',b} \vee \bigvee_{a'\notin\{a,b\}} q_{a',i} \vee \bigvee_{b'\notin\{a,b\}} q_{b',k} \vee \overline{r}_a \vee \overline{r}_b}^{\text{conditional part}} \vee \overbrace{q_{a,i} \vee \overline{q}_{b,i} \vee q_{b,k} \vee \overline{q}_{a,k}}^{\text{autarky part}} \qquad (16)$$

We add as conditional autarkies $C^{i\to a\to i}_{j\to b\to k}$ for all $i < n$, for all j where $i < j \leq n+1$, for all k where $i < k \leq n$, and for all distinct a,b in $[m]$. We add them in increasing order w.r.t. index i. Among clauses with the same index i, any ordering of them is fine.

Fix some $C^{i\to \hat{a}\to i}_{j\to \hat{b}\to \hat{k}}$ and let the partial assignment α be the (the negation of) its conditional part, and let $\beta = \{q_{\hat{a},i} = 1, q_{\hat{a},\hat{k}} = 0, q_{\hat{b},\hat{k}} = 1, q_{\hat{b},i} = 0\}$. We need to check that each clause in the proof before it is either satisfied by $\alpha \cup \beta$ or untouched by β.

We first check that $C^{i\to \hat{a}\to i}_{j\to \hat{b}\to \hat{k}}$ is conditional autarky w.r.t. to clauses (11)–(15). Assignment β only touches clauses mentioning one or both resting places $\hat{a}$ and $\hat{b}$: it satisfies their totality axioms (14), and satisfies the hole axioms (15) involving both. All other hole axioms touched by β are satisfied by α.

We now check that $C^{\hat{\imath}\to\hat{a}\to\hat{\imath}}_{\hat{\jmath}\to\hat{b}\to\hat{k}}$ is a conditional autarky against some previously added clause $C := C^{i\to a\to i}_{j\to b\to k}$, for $i \le \hat{\imath}$ (see (16)). When $\{i,k\}\cap\{\hat{\imath},\hat{k}\} = \emptyset$, the assignment β does not touch C at all. When the intersection is non-empty, we have six cases, and for all of them we show that $\alpha\cup\beta$ satisfies C. Notice that $\hat{k} \neq i$ because $\hat{k} > \hat{\imath} \ge i$.

- Case 1 ($\hat{\imath} = i, \hat{a} \neq b$): $\alpha\cup\beta$ sets $q_{\hat{a},\hat{\imath}}$, or equivalently variable $q_{\hat{a},i}$, to 1. Since $\hat{a} \neq a$, this variable occurs positively in C, in the disjunction $\bigvee_{a'\notin\{a,b\}} q_{a',i}$.
- Case 2 ($\hat{\imath} = i, \hat{a} = b$): α sets $p_{\hat{\imath},\hat{a}}$, or equivalently $p_{i,b}$, to 1. Since $j \neq i$ this variable occurs positively in C, in the disjunction $\bigvee_{j'\neq j} p_{j',b}$.
- Case 3 ($\hat{\imath} = k, \hat{a} \neq a$): $\alpha\cup\beta$ sets $q_{\hat{a},\hat{\imath}}$, or equivalently variable $q_{\hat{a},k}$, to 1. Since $\hat{a} \neq a$, this variable occurs positively in C, either in disjunction $\bigvee_{b'\notin\{a,b\}} q_{b',k}$ or in the autarky part.
- Case 4 ($\hat{\imath} = k, \hat{a} = a$): α sets $p_{\hat{\imath},\hat{a}}$, or equivalently $p_{k,a}$, to 1. Since $i < k$, this variable occurs positively in C in the disjunction $\bigvee_{i'\neq i} p_{i',a}$.
- Case 5 ($\hat{k} = k, \hat{b} \neq a$): $\alpha\cup\beta$ sets $q_{\hat{b},\hat{k}}$, or equivalently variable $q_{\hat{b},k}$, to 1. Since $\hat{b} \neq a$, this variable occurs positively in C, either in disjunction $\bigvee_{b'\notin\{a,b\}} q_{b',k}$ or in the autarky part.
- Case 6 ($\hat{k} = k, \hat{b} = a$): α sets $p_{\hat{\jmath},\hat{b}}$, or equivalently $p_{\hat{\jmath},a}$, to 1. Since $\hat{\jmath} > \hat{\imath} \ge i$, and in particular $\hat{\jmath} \neq i$, hence $p_{\hat{\jmath},a}$ occurs positively in C in the disjunction $\bigvee_{i'\neq i} p_{i',a}$.

We have just shown that $C^{\hat{\imath}\to\hat{a}\to\hat{\imath}}_{\hat{\jmath}\to\hat{b}\to\hat{k}}$ is a conditional autarky w.r.t. to all clauses of the relativized pigeonhole principle and all clauses $C^{i\to a\to i}_{j\to b\to k}$ added before. Therefore we can add them all in the SBC$^-$ proof system.

Resolution steps. First remove all positive literals from all the conditional autarkies, resolving the clauses $C^{i\to a\to i}_{j\to b\to k}$ with hole axioms involving $\overline{p}_{i,a}$, $\overline{p}_{j,b}$, $\overline{q}_{a,k}$ and $\overline{q}_{b,i}$, and then with activation axioms $\overline{p}_{i,a}\vee r_a$ and $\overline{p}_{j,b}\vee r_b$, to get clauses

$$\overline{p}_{i,a}\vee\overline{q}_{a,k}\vee\overline{p}_{j,b}\vee\overline{q}_{b,i} \tag{17}$$

for all $j,k > i$ and $a,b\in[m]$.

The rest of this refutation is the derivation, by induction on $i = 0,\dots,n$, that any pigeon of index above $j > i$ must be mapped to a hole $k > i$. Namely we want the clauses

$$\overline{p}_{j,b}\vee\bigvee_{k>i} q_{b,k} \qquad \text{for } j > i \text{ and } b\in[m]. \tag{18}$$

At step $i = 0$, clauses (18) are the result of resolving activation axiom $\overline{p}_{j,b}\vee r_b$ with the totality axiom (14) for resting place b. Now assume that we have (18) for step $i-1$. We have clause (17) for fixed a,b,j and varying $k > i$. We resolve all such clauses with $\overline{p}_{i,a}\vee\bigvee_{k>(i-1)} q_{a,k}$, that we have already by induction hyphothesis, and get clause

$$\overline{p}_{i,a}\vee\overline{p}_{j,b}\vee\overline{q}_{b,i}\vee q_{a,i}\ . \tag{19}$$

We should remark that last step is only valid for $i < n$. When $i = n$, clause $\overline{p}_{n,a}\vee q_{a,n}$ is available by induction hypothesis and therefore (19) can be obtained directly by weakening. Next goal is to deduce

$$\overline{p}_{i,a}\vee\overline{p}_{j,b}\vee\overline{q}_{b,i}\vee\overline{q}_{a,i} \tag{20}$$

by resolving two activation axioms with a hole axiom (15) for hole i. We resolve (19) and (20) and get $\overline{p}_{i,a}\vee\overline{p}_{j,b}\vee\overline{q}_{b,i}$. Resolve all such clauses, for varying a and fixed j,b,i, with the totality

axiom of pigeon i, to finally get $\overline{p}_{j,b} \vee \overline{q}_{b,i}$. By induction hypothesis we have $\overline{p}_{j,b} \vee \bigvee_{k>i-1} q_{b,k}$ which, resolved with clause $\overline{p}_{j,b} \vee \overline{q}_{b,i}$ that we just derived, gives our goal clause (18).

When we finally arrive to $i = n$, clause (18) is just $\overline{p}_{n+1,b}$ for all $b \in [m]$, which immediately implies contradiction with the totality axiom for pigeon $n + 1$.

The size of the initial formula is $\mathcal{O}(m^2 n)$, since $m \geq n$. There are $\mathcal{O}(n^3 m^2)$ clauses added as conditional autarkies. The proof is dominated by obtaining clauses (17), taking $\mathcal{O}(n + m)$ resolution steps for each conditional autarky added, so the proof size is $\mathcal{O}(n^3 m^3)$. ◀

5 Conclusions

We conclude this work with a couple of open questions. Are redundancy based systems (SBC$^-$/PR$^-$/...) able to give polynomial size refutations of the perfect matching principle on expander graphs, either on the random graph or even structured expanders such as Marguli-Gabber-Galil expanders? Theorem 3.3 can be seen also as a preliminary step in this direction. A similar question was already raised in [6].

We also showed that the relativized pigeonhole principle has polynomial-size SBC$^-$ refutations. This principle has some structural similarities with the clique-coloring principle. Does clique-coloring also have polynomial size SBC$^-$ refutations?

References

1 Miklos Ajtai. The independence of the modulo p counting principles. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 402–411. ACM, 1994.

2 Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *J. Artif. Intell. Res. (JAIR)*, 40:353–373, 2011. doi:10.1613/jair.3152.

3 Albert Atserias, Massimo Lauria, and Jakob Nordström. Narrow proofs may be maximally long. *ACM Trans. Comput. Logic*, 17(3), 2016. doi:10.1145/2898435.

4 Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 73:1–26, 1996.

5 Sam Buss and Jakob Nordström. Proof complexity and SAT solving. In Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability 2nd Edition*, pages 233–350. IOS Press, 2021. doi:10.3233/FAIA200990.

6 Sam Buss and Neil Thapen. DRAT and propagation redundancy proofs without new variables. *Log. Methods Comput. Sci.*, 17(2), 2021. URL: https://lmcs.episciences.org/7400.

7 Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52(4):916–927, December 1987. doi:10.2307/2273826.

8 Niklas Eén and Armin Biere. Effective preprocessing in SAT through variable and clause elimination. *Lecture Notes in Computer Science*, pages 61–75, 2005. doi:10.1007/11499107_5.

9 Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985. doi:10.1016/0304-3975(85)90144-6.

10 Marijn J. H. Heule, Benjamin Kiesl, and Armin Biere. Clausal proofs of mutilated chessboards. In *NASA Formal Methods - 11th International Symposium, NFM 2019, Houston, TX, USA, May 7-9, 2019, Proceedings*, pages 204–210, 2019. doi:10.1007/978-3-030-20652-9_13.

11 Marijn J. H. Heule, Benjamin Kiesl, and Armin Biere. Strong extension-free proof systems. *Journal of Automated Reasoning*, 64:533–554, 2020. Conference version at CADE '17. doi:10.1007/s10817-019-09516-0.

12 Marijn JH Heule, Benjamin Kiesl, Martina Seidl, and Armin Biere. Pruning through satisfaction. In *Haifa Verification Conference*, pages 179–194. Springer, 2017.

13 Johan Håstad. On small-depth Frege proofs for PHP. *TheoretiCS*, Phase 2, December 2025. doi:10.46298/theoretics.25.27.

14 Matti Järvisalo, Marijn JH Heule, and Armin Biere. Inprocessing rules. In *International Joint Conference on Automated Reasoning*, pages 355–370. Springer, 2012.

15 Benjamin Kiesl, Marijn J. H. Heule, and Armin Biere. Truth assignments as conditional autarkies. In *Automated Technology for Verification and Analysis - 17th International Symposium, ATVA 2019, Taipei, Taiwan, October 28-31, 2019, Proceedings*, pages 48–64, 2019. doi:10.1007/978-3-030-31784-3_3.

16 Benjamin Kiesl, Adrián Rebola-Pardo, and Marijn J. H. Heule. Extended resolution simulates DRAT. In Didier Galmiche, Stephan Schulz, and Roberto Sebastiani, editors, *IJCAR 2018*, pages 516–531, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-94205-6_34.

17 Benjamin Kiesl, Martina Seidl, Hans Tompits, and Armin Biere. Local redundancy in SAT: Generalizations of blocked clauses. *Logical Methods in Computer Science*, 14, 2018. doi:10.23638/LMCS-14(4:3)2018.

18 Hans Kleine Büning and Oliver Kullmann. Minimal unsatisfiability and autarkies. In *Handbook of Satisfiability*, pages 339–401. IOS Press, 2009. doi:10.3233/978-1-58603-929-5-339.

19 Jan Krajícek. Combinatorics of first order structures and propositional proof systems. *Arch. Math. Log.*, 43:427–441, May 2004. doi:10.1007/s00153-003-0186-y.

20 Oliver Kullmann. On a generalization of extended resolution. *Discrete Applied Mathematics*, 96-97:149–176, 1999. doi:10.1016/S0166-218X(99)00037-2.

21 João P Marques-Silva and Karem A Sakallah. Grasp: A search algorithm for propositional satisfiability. *Computers, IEEE Transactions on*, 48(5):506–521, 1999. doi:10.1109/12.769433.

22 Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artificial Intelligence*, 175(2):512–525, 2011. doi:10.1016/j.artint.2010.10.002.

23 Adrián Rebola-Pardo and Martin Suda. A theory of satisfiability-preserving proofs in SAT solving. In *LPAR*, pages 583–603, 2018. doi:10.29007/TC7Q.

24 Joseph E. Reeves, Marijn J. H. Heule, and Randal E. Bryant. Preprocessing of propagation redundant clauses. In *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, pages 106–124, 2022. doi:10.1007/978-3-031-10769-6_8.

25 Amar Shah, Twain Byrnes, Joseph E. Reeves, and Marijn J. H. Heule. Learning short clauses via conditional autarkies. In *Proceedings of the 25th Conference on Formal Methods in Computer-Aided Design, FMCAD 2025, Menlo Park, CA, USA, October 6-10, 2025*, 2025. doi:10.34727/2025/ISBN.978-3-85448-084-6_17.

26 Allen Van Gelder. Complexity analysis of propositional resolution with autarky pruning. *Discrete Appl. Math.*, 96–97(1):195–221, October 1999. doi:10.1016/S0166-218X(99)00040-2.

27 Emre Yolcu. Lower bounds for set-blocked clauses proofs. In *STACS 2024*, volume 289, pages 59:1–59:17, 2024. doi:10.4230/LIPIcs.STACS.2024.59.

28 Emre Yolcu and Marijn J. H. Heule. Exponential separations using guarded extension variables. In *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, pages 101:1–101:22, 2023. doi:10.4230/LIPIcs.ITCS.2023.101.