Lower Bounds for Set-Blocked Clauses Proofs

Emre Yolcu
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We study propositional proof systems with inference rules that formalize restricted versions of the ability to make assumptions that hold without loss of generality, commonly used informally to shorten proofs. Each system we study is built on resolution. They are called BC^-, RAT^-, SBC^-, and GER^-, denoting respectively blocked clauses, resolution asymmetric tautologies, set-blocked clauses, and generalized extended resolution – all "without new variables." They may be viewed as weak versions of extended resolution (ER) since they are defined by first generalizing the extension rule and then taking away the ability to introduce new variables. Except for SBC^-, they are known to be strictly between resolution and extended resolution.

Several separations between these systems were proved earlier by exploiting the fact that they effectively simulate ER. We answer the questions left open: We prove exponential lower bounds for SBC^- proofs of a binary encoding of the pigeonhole principle, which separates ER from SBC^-. Using this new separation, we prove that both RAT^- and GER^- are exponentially separated from SBC^-. This completes the picture of their relative strengths.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases proof complexity, separations, resolution, extended resolution, blocked clauses

Digital Object Identifier 10.4230/LIPIcs.STACS.2024.59

Related Version *arXiv Version*: `https://arxiv.org/abs/2401.11266`

Funding This material is based upon work supported by the National Science Foundation under grant CCF-2015445.

Acknowledgements I thank Sam Buss, Marijn Heule, Jakob Nordström, and Ryan O'Donnell for useful discussions. I thank Jeremy Avigad, Ryan O'Donnell, and Bernardo Subercaseaux for feedback on an earlier version of the paper. Finally, I thank the STACS reviewers for their highly detailed comments and suggestions.

1 Introduction

When writing proofs informally, it is sometimes convenient to make assumptions that hold "without loss of generality." For instance, if we are proving a statement about real numbers x and y, we might assume without loss of generality that $x \geq y$ and continue the proof under this additional assumption. Such an assumption requires justification, for instance by arguing that the two variables are interchangeable in the statement being proved. Assumptions of this kind are not essential to proofs but they simplify or shorten the presentation.

We study propositional proof systems[1] with inference rules that allow making such assumptions. Extended resolution [21] (equivalently, Extended Frege [5]) already simulates this kind of reasoning; however, it presumably does more, and its strength is poorly understood. We thus focus on "weak" systems built on top of resolution [1, 20] and lacking the ability to introduce any new variables, while still being able to reason without loss of generality. Each system relies on a polynomial-time verifiable syntactic condition to automatically justify the assumption being made, and the exact form of this condition determines the strength of the proof system. The systems are defined by first generalizing the extension rule and then taking away the ability to introduce new variables, so, for lack of a better term, we

[1] Throughout the rest of this paper, by "proof" we mean a proof of unsatisfiability (i.e., a refutation).

will refer to these systems collectively as "weak extended resolution systems" in this section. Although we are referring to them as weak, some variants are surprisingly strong in that they admit polynomial-size proofs of the pigeonhole principle, bit pigeonhole principle, parity principle, clique–coloring principle, and Tseitin tautologies, as well as being able to undo (with polynomial-size derivations) the effects of or-ification, xor-ification, and lifting with indexing gadgets [2]. In this paper, we study the relative strengths of several variants of those systems and answer the questions left open in previous work [23, Section 1.4].

1.1 Motivation

Our interest in weak extended resolution systems is rooted in two different areas: proof complexity and satisfiability (SAT) solving.

1.1.1 Proof complexity

Proof complexity is concerned with the sizes of proofs in propositional proof systems. The notion of a proof system as accepted in proof complexity is rather general – it covers not only the "textbook" deductive systems for propositional logic but also several systems that capture different forms of mathematical reasoning. For instance, the widely studied proof systems of cutting planes [6] and polynomial calculus [3] utilize simple forms of geometric and algebraic reasoning, respectively. Weak extended resolution systems are somewhat similar, although they do not originate from any specific branch of mathematics. Instead, they capture the pervasive technique of reasoning without loss of generality, often used to shorten proofs. Since proof complexity is concerned with proof size, the limits of the degree of brevity achievable by this form of reasoning is a natural question from the perspective of proof complexity.

Moreover, the upper bounds proved by Buss and Thapen [2] show that many of the usual "hard" combinatorial principles used for proof complexity lower bounds are easy to prove in certain weak extended resolution systems. In other words, a modest amount of the ability to reason without loss of generality lends surprising strength to even a system as weak as resolution, and the full strength of extended resolution is not required for the combinatorial principles previously mentioned. Thus, many of the existing separations between extended resolution and the commonly studied proof systems can be attributed to the fact that extended resolution can reason without loss of generality while the other systems cannot. Searching for principles that separate extended resolution from the weak extended resolution systems will help us better understand other facets of the strength of extended resolution.

1.1.2 SAT solving

Another motivation for studying the weak extended resolution systems is their potential usefulness for improvements in SAT solvers, which are practical implementations of propositional theorem provers that determine whether a given formula in conjunctive normal form is satisfiable. When a solver claims unsatisfiability, it is expected to produce a proof that can be used to verify the claim efficiently. Modern SAT solvers, which are based on conflict-driven clause learning (CDCL) [17], essentially search for resolution proofs. Consequently, the well-known exponential lower bounds against resolution (e.g., [7, 22]) imply exponential lower bounds against the runtimes of CDCL-based solvers. To overcome the limitations of resolution, SAT solvers are forced to go beyond CDCL.

Many of the current solvers employ "inprocessing" techniques [11], which support inferences of the kind that we study in this paper. These techniques are useful in practice; however, they are implemented as ad hoc additions to CDCL. Weak extended resolution

systems hold the potential for improving SAT solvers in a more principled manner (e.g., through the development of a solving paradigm that corresponds to one of those systems in a manner similar to how CDCL corresponds to resolution). In comparison, proof systems such as cutting planes, polynomial calculus, DNF resolution [13], or Frege [5] appear more difficult to take advantage of, at least in part due to their richer syntax. When dealing only with clauses, it becomes possible to achieve highly efficient constraint propagation, which is an important reason for the speed of CDCL-based solvers. Extended resolution also works only with clauses; however, there are currently no widely applicable heuristics for introducing new variables during proof search. Weak extended resolution systems are relatively strong despite using only clauses without new variables. Thus, those systems are promising for practical proof search algorithms.

Earlier works [10, 8] showed that solvers based on certain weak extended resolution systems can automatically discover small proofs of some formulas that are hard for resolution, such as the pigeonhole principle and the mutilated chessboard principle. Still, those solvers fall behind CDCL-based solvers on other classes of formulas. Moreover, there appears to be a tradeoff when choosing a system for proof search: stronger systems enable smaller proofs; however, proof search in such systems is costlier with respect to proof size. To be able to choose the ideal system for proof search (e.g., the weakest system that is strong enough for one's purposes), it is important to understand the relative strengths of the systems in question. This paper is a step towards that goal.

1.2 Background

We briefly review some background, deferring formal definitions to Section 2. For comprehensive overviews of related work, see Buss and Thapen [2] and Yolcu and Heule [23].

The proof systems we study are based on the notion of "redundancy." A clause is *redundant* with respect to a formula if it can be added to or removed from the formula without affecting satisfiability. Redundancy is a generalization of logical implication: if $\Gamma \models C$ then the clause C is redundant with respect to the set Γ of clauses;[2] however, the converse is not necessarily true. When proving unsatisfiability, deriving redundant clauses corresponds to making assumptions that hold without loss of generality.[3] To ensure that proofs can be checked in polynomial time, we work with restricted versions of redundancy that rely on syntactic conditions.

Possibly the simplest interesting version is blockedness [14, 15]. We say a clause C is *blocked* with respect to a set Γ of clauses if there exists a literal $p \in C$ such that all possible resolvents of C on p against clauses from Γ are tautological (i.e., contain a literal and its negation). Kullmann [16] showed that blocked clauses are special cases of redundant clauses and thus considered an inference rule that, given a formula Γ, allows us to extend Γ with a clause that is blocked with respect to Γ. This rule, along with resolution, gives the proof system called *blocked clauses* (BC). It is apparent from the definition of a blocked clause that deleting clauses from Γ enlarges the set of clauses that are blocked with respect to Γ. With this observation, Kullmann defined a strengthening of BC called *generalized extended resolution* (GER) that allows temporary deletion of clauses from Γ. Later works [11, 12] defined more general classes of redundant clauses and proof systems based on them, called *resolution asymmetric tautologies* (RAT) and *set-blocked clauses* (SBC). Both RAT and SBC

[2] We use "set of clauses" and "formula" interchangeably.

[3] When refuting a formula Γ, deriving a redundant clause C may be viewed as stating the following: "If there exists an assignment satisfying Γ, then there also exists an assignment satisfying both Γ and C, so without loss of generality we can assume that C holds."

use relaxed versions of blocked clauses: RAT changes the word "tautological" in the definition of a blocked clause, and, in a sense, SBC considers possible resolvents on more than a single literal.

As defined, BC simulates extended resolution since extension clauses can be added in sequence as blocked clauses if we are allowed to introduce new variables (see [16, Section 6]). Thus, we disallow new variables to weaken the systems. A proof of Γ is *without new variables* if it contains only the variables that already occur in Γ. We denote a proof system variant that disallows new variables with the superscript "$-$" (e.g., BC^- is BC without new variables). We are concerned in this paper with the strengths of the systems RAT^-, SBC^-, and GER^-, each of which generalizes BC^- in different ways.

As a technical side note, the systems we study are unusual in the following respects: They are not monotonic, since a clause redundant with respect to Γ is not necessarily redundant with respect to $\Gamma' \supseteq \Gamma$. This causes *deletion* (i.e., the ability to delete clauses in the middle of a proof) to increase the strength of those systems. It also requires one to pay attention to the order of inferences when proving upper bounds. Additionally, they are not a priori closed under restrictions, which means that extra care is required when proving lower bounds against them. More specifically, a proof system P that simulates BC^- is not closed under restrictions unless P also simulates extended resolution [2, Theorem 2.4]. It follows from earlier lower bounds [16, 2] and Section 3 in this paper that BC^-, RAT^-, SBC^-, and GER^- are not closed under restrictions.

1.3 Results

This work follows up on Yolcu and Heule [23], which proved separations between the different generalizations of BC^- by exploiting the fact that, although the systems cannot introduce new variables, they nevertheless effectively simulate [18] extended resolution. Their strategy uses so-called "guarded extension variables," where we consider systems P and Q that both effectively simulate a strong system R, and we incorporate extension variables into formulas in a guarded way that allows P to simulate an R-proof while preventing Q from making any meaningful use of the extension variables to achieve a speedup. This allows using, as black-box, a separation of R from Q to separate P from Q. For more details about this strategy, we refer the reader to Yolcu and Heule [23, Section 1.3].

In this paper, we prove the following results, where each formula indexed by n has $n^{O(1)}$ variables and $n^{O(1)}$ clauses. Figure 1 summarizes the proof complexity landscape around BC^- after these results.

We first show exponential lower bounds for SBC^- proofs of a binary encoding of the pigeonhole principle called the "bit pigeonhole principle," defined in Section 3. (Note that the usual unary encoding of the pigeonhole principle admits polynomial-size proofs in SBC^- [23, Lemma 7.1].)

▶ **Theorem 1.** *The bit pigeonhole principle* BPHP_n *requires* SBC^- *proofs of size* $2^{\Omega(n)}$.

We then show, using constructions that incorporate guarded extension variables into BPHP_n, that RAT^- and GER^- are both exponentially separated from SBC^-.

▶ **Theorem 2.** *There exists an infinite sequence* $(\Gamma_n)_{n=1}^{\infty}$ *of unsatisfiable formulas such that* Γ_n *admits* RAT^- *proofs of size* $n^{O(1)}$ *but requires* SBC^- *proofs of size* $2^{\Omega(n)}$.

▶ **Theorem 3.** *There exists an infinite sequence* $(\Delta_n)_{n=1}^{\infty}$ *of unsatisfiable formulas such that* Δ_n *admits* GER^- *proofs of size* $n^{O(1)}$ *but requires* SBC^- *proofs of size* $2^{\Omega(n)}$.

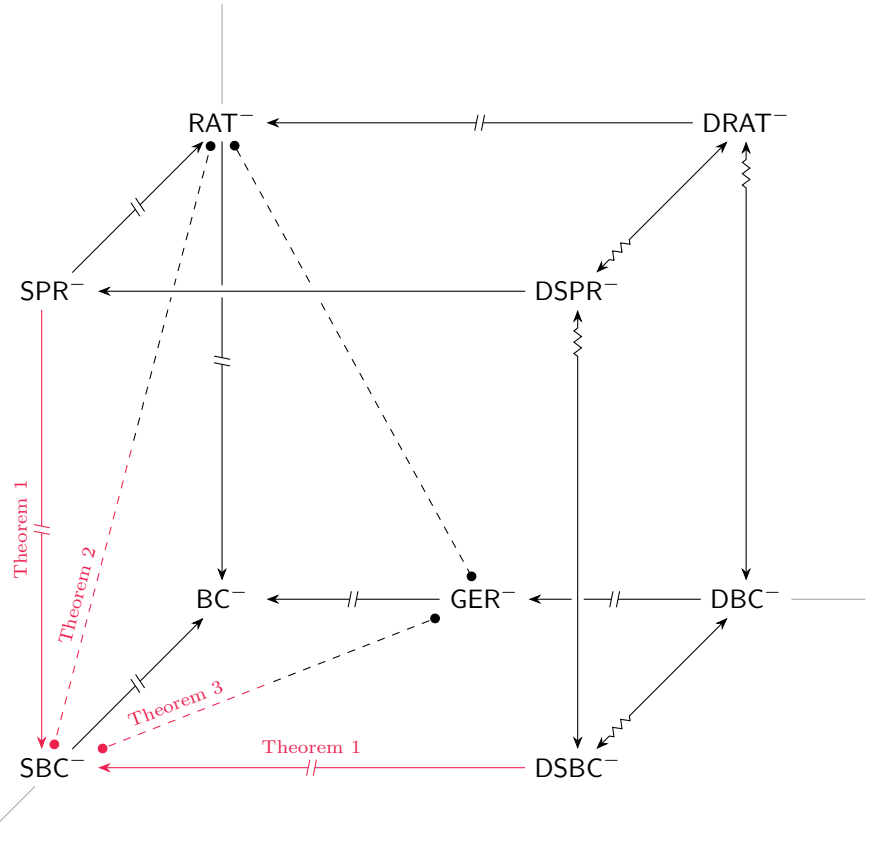

Figure 1 In the above diagram, the proof systems are placed in three-dimensional space with BC^-, the weakest system, at the origin. Moving away from the origin along each axis corresponds to a particular way of generalizing (i.e., strengthening) a proof system. The systems prefixed with "D" allow the arbitrary deletion of a clause as a proof step. For systems P and Q, we use $P \longrightarrow Q$ to denote that P simulates Q; (and $P \rightsquigarrow Q$ to indicate an "interesting" simulation, where P is not simply a generalization of Q); $P \dashrightarrow Q$ to denote that P is exponentially separated from Q (i.e., there exists an infinite sequence of formulas admitting polynomial-size proofs in P while requiring exponential-size proofs in Q); and $P \not\vdash Q$ to denote that P both simulates Q and is exponentially separated from Q. Arrows in red indicate the relationships that are new in this paper. To reduce clutter, some relationships that are implied by transitivity are not displayed (e.g., DBC^- simulates RAT^- and is exponentially separated from it through DRAT^-).

The above results, along with earlier ones, completely describe the relative strengths of the weakest generalizations of BC^- along each axis in Figure 1. For lower bounds, this pushes the frontier to the system called *set propagation redundancy* (SPR^-) [9]. We do not define SPR^- formally in this paper, although it may be thought of intuitively as combining SBC^- and RAT^-. The upper bounds proved by Buss and Thapen [2] establish SPR^- as an interesting target for proof complexity lower bounds. (Note that the binary encoding of the pigeonhole principle that we use to prove exponential lower bounds for SBC^- admits polynomial-size proofs in SPR^- [2, Theorem 4.4].)

2 Preliminaries

We assume that the reader is familiar with propositional logic, proof complexity, resolution, and extended resolution. We review some concepts to describe our notation. For notation we follow Yolcu and Heule [23] from which this section is adapted.

We denote the set of strictly positive integers by $\mathbb{N}^+$. For $n \in \mathbb{N}^+$, we let $[n] \coloneqq \{1, \dots, n\}$. For a sequence $S = (x_1, \dots, x_n)$, its *length* is n, which we denote by $|S|$.

2.1 Propositional logic

We use 0 and 1 to denote *False* and *True*, respectively. A *literal* is a propositional variable or its negation. A set of literals is *tautological* if it contains a pair of complementary literals x and $\overline{x}$. A *clause* is the disjunction of a nontautological set of literals. We use $\bot$ to denote the empty clause. We denote by $\mathbf{V}$ and $\mathbf{L}$ respectively the sets of all variables and all literals. A *conjunctive normal form formula* (CNF) is a conjunction of clauses. Throughout this paper, by "formula" we mean a CNF. We identify clauses with sets of literals and formulas with sets of clauses. In the rest of this section we use C, D to denote clauses and Γ, Δ to denote formulas.

We say D is a *weakening* of C if $C \subseteq D$. We denote by $\mathrm{var}(\Gamma)$ the set of all the variables occurring in Γ.

When we know $C \cup D$ to be nontautological, we write it as $C \vee D$. We write $C \dot{\vee} D$ to indicate a *disjoint disjunction*, where C and D have no variables in common. We take the disjunction of a clause and a formula as

$$C \vee \Delta \coloneqq \{C \vee D \colon D \in \Delta \text{ and } C \cup D \text{ is nontautological}\}.$$

An *assignment* α is a partial function $\alpha \colon \mathbf{V} \rightharpoonup \{0, 1\}$, which also acts on literals by letting $\alpha(\overline{x}) \coloneqq \overline{\alpha(x)}$. We identify α with the set $\{p \in \mathbf{L} \colon \alpha(p) = 1\}$, consisting of all the literals it satisfies. For a set L of literals, we let $\overline{L} \coloneqq \{\overline{x} \colon x \in L\}$. In particular, we use $\overline{C}$ to denote the smallest assignment that falsifies all the literals in C. We say α *satisfies* C, denoted $\alpha \models C$, if there exists some $p \in C$ such that $\alpha(p) = 1$. We say α satisfies Γ if for all $C \in \Gamma$ we have $\alpha \models C$. For C that α does not satisfy, the *restriction* of C under α is $C|_\alpha \coloneqq C \setminus \{p \in C \colon \alpha(p) = 0\}$. Extending this to formulas, the restriction of Γ under α is $\Gamma|_\alpha \coloneqq \{C|_\alpha \colon C \in \Gamma \text{ and } \alpha \not\models C\}$.

We say Γ and Δ are *equisatisfiable*, denoted $\Gamma \equiv_{\mathrm{sat}} \Delta$, if they are either both satisfiable or both unsatisfiable. With respect to Γ, a clause C is *redundant* if $\Gamma \setminus \{C\} \equiv_{\mathrm{sat}} \Gamma \equiv_{\mathrm{sat}} \Gamma \cup \{C\}$. We sometimes write $\Gamma \cup \{C\}$ as $\Gamma \wedge C$.

2.2 Proof complexity and resolution

For a proof system P and a formula Γ, we define

$$\mathrm{size}_P(\Gamma) \coloneqq \min\{|\Pi| \colon \Pi \text{ is a } P\text{-proof of } \Gamma\}$$

if Γ is unsatisfiable and $\mathrm{size}_P(\Gamma) \coloneqq \infty$ otherwise. A proof system P *simulates* Q if every Q-proof can be converted in polynomial time into a P-proof of the same formula. Proof systems P and Q are *equivalent* if they simulate each other. We say P is *exponentially separated* from Q if there exists some sequence $(\Gamma_n)_{n=1}^{\infty}$ of formulas such that $\mathrm{size}_P(\Gamma_n) = n^{O(1)}$ while $\mathrm{size}_Q(\Gamma_n) = 2^{\Omega(n)}$. We call such a sequence of formulas *easy* for P and *hard* for Q.

Let $C \dot{\vee} x$ and $D \dot{\vee} \overline{x}$ be clauses, where x is a variable, such that the set $C \cup D$ is nontautological. We call the clause $C \vee D$ the *resolvent* of $C \vee x$ and $D \vee \overline{x}$ on x. We define a resolution proof in a slightly different form than usual: as a sequence of formulas instead of a sequence of clauses.[4]

[4] The resulting proof system is equivalent to the usual version of resolution.

▶ **Definition 4.** *A* resolution proof *of a formula* Γ *is a sequence* $\Pi = (\Gamma_1, \dots, \Gamma_N)$ *of formulas such that* $\Gamma_1 = \Gamma$, $\bot \in \Gamma_N$, *and for all* $i \in [N-1]$, *we have* $\Gamma_{i+1} = \Gamma_i \cup \{C\}$, *where* C *is either the resolvent of two clauses in* Γ_i *or a weakening of some clause in* Γ_i. *The size of* Π *is* N.

We write Res to denote the resolution proof system. A well known fact is that resolution is closed under restrictions: if $(\Gamma_1, \Gamma_2, \dots, \Gamma_N)$ is a resolution proof of Γ, then for every assignment α, the sequence $(\Gamma_1|_\alpha, \Gamma_2|_\alpha, \dots, \Gamma_N|_\alpha)$ contains as a subsequence a resolution proof of $\Gamma|_\alpha$. This implies in particular the following.

▶ **Lemma 5.** *For every formula* Γ *and every assignment* α, $\mathrm{size}_{\mathsf{Res}}(\Gamma|_\alpha) \leq \mathrm{size}_{\mathsf{Res}}(\Gamma)$.

A *unit propagation proof* is a resolution proof where each use of the resolution rule has as at least one of its premises a clause that consists of a single literal. Unit propagation is not complete. With Γ a formula and $L = \{p_1, \dots, p_k\}$ a set of literals, we write $\Gamma \vdash_1 L$ to denote that there exists a unit propagation proof of $\Gamma \wedge \overline{p_1} \wedge \cdots \wedge \overline{p_k}$.

We next define *extended resolution* (ER), which is a strengthening of resolution.

▶ **Definition 6.** *Let* Γ *be a formula and* p, q *be arbitrary literals. Consider a* new *variable* x *(i.e., not occurring in any one of* Γ, p, q*). We refer to* $\{\overline{x} \vee p,\ \overline{x} \vee q,\ x \vee \overline{p} \vee \overline{q}\}$ *as a set of* extension clauses *for* Γ. *In this context, we call* x *the* extension variable.

▶ **Definition 7.** *A formula* Λ *is an* extension *for a formula* Γ *if there exists a sequence* $(\lambda_1, \dots, \lambda_t)$ *such that* $\Lambda = \bigcup_{i=1}^{t} \lambda_i$, *and for all* $i \in [t]$, *we have that* λ_i *is a set of extension clauses for* $\Gamma \cup \bigcup_{j=1}^{i-1} \lambda_j$.

▶ **Definition 8.** *An* extended resolution proof *of a formula* Γ *is a pair* (Λ, Π), *where* Λ *is an extension for* Γ *and* Π *is a resolution proof of* $\Gamma \cup \Lambda$. *The size of* (Λ, Π) *is* $|\Lambda| + |\Pi|$.

2.3 Redundancy criteria

We recall the syntactic redundancy criteria that lead to the inference rules we study. The definitions are taken from Yolcu and Heule [23, Section 3], which in turn adapted them from previous works [16, 11, 12, 9, 2].

▶ **Definition 9.** *A clause* $C = p \dot\vee C'$ *is a* blocked clause *(BC) for the literal* p *with respect to a formula* Γ *if, for every clause* D *of the form* $\overline{p} \dot\vee D'$ *in* Γ, *the set* $C' \cup D'$ *is tautological.*

▶ **Definition 10.** *A clause* $C = p \dot\vee C'$ *is a* resolution asymmetric tautology *(RAT) for the literal* p *with respect to a formula* Γ *if, for every clause* D *of the form* $\overline{p} \dot\vee D'$ *in* Γ, *we have* $\Gamma \vdash_1 C' \cup D'$.

▶ **Definition 11.** *A clause* C *is a* set-blocked clause *(SBC) for a nonempty* $L \subseteq C$ *with respect to a formula* Γ *if, for every clause* $D \in \Gamma$ *with* $D \cap \overline{L} \neq \varnothing$ *and* $D \cap L = \varnothing$, *the set* $(C \setminus L) \cup (D \setminus \overline{L})$ *is tautological.*

We say C is a BC with respect to Γ if there exists a literal $p \in C$ for which C is a BC with respect to Γ, and similarly for RAT and SBC. It was shown in previous works [16, 11, 12] that BCs, RATs, and SBCs are redundant, which makes it possible to use them to define proof systems.

▶ **Definition 12.** *A* blocked clauses proof *of a formula* Γ *is a sequence* $\Pi = (\Gamma_1, \dots, \Gamma_N)$ *of formulas such that* $\Gamma_1 = \Gamma$, $\bot \in \Gamma_N$, *and for all* $i \in [N-1]$, *we have* $\Gamma_{i+1} = \Gamma_i \cup \{C\}$, *where* C *is either the resolvent of two clauses in* Γ_i, *a weakening of some clause in* Γ_i, *or a blocked clause with respect to* Γ_i. *The size of* Π *is* N.

We write BC to denote the blocked clauses proof system. Replacing "blocked clause" by "resolution asymmetric tautology" in the above definition gives the RAT proof system. Replacing it by "set-blocked clause" gives the SBC proof system.[5] RAT and SBC are two generalizations of BC, and we now define another, called *generalized extended resolution* (GER), which reduces the dependence of the validity of BC inferences on the order of clause additions (see [16, Section 1.3]). We need to introduce the concept of a blocked extension[6] before we can proceed with the definition of GER.

▶ **Definition 13.** *A formula* Λ *is a* blocked extension *for a formula* Γ *if there exists a subset* Γ' *of* Γ *and an ordering* $(C_1, \ldots, C_r)$ *of all the clauses in* $\Lambda \cup (\Gamma \setminus \Gamma')$ *such that for all* $i \in [r]$ *the clause* C_i *is blocked with respect to* $\Gamma' \cup \bigcup_{j=1}^{i-1} \{C_j\}$.

▶ **Definition 14.** *A* generalized extended resolution proof *of a formula* Γ *is a pair* (Λ, Π), *where* Λ *is a blocked extension for* Γ *and* Π *is a resolution proof of* $\Gamma \cup \Lambda$. *The size of* (Λ, Π) *is* $|\Lambda| + |\Pi|$.

Note that the definitions in this section do not prohibit BCs, RATs, or SBCs with respect to Γ from containing variables not occurring in Γ. We study the variants of BC, RAT, SBC, and GER that disallow the use of new variables. A proof of Γ is *without new variables* if all the variables occurring in the proof are in $\mathrm{var}(\Gamma)$. In the case of GER, this constraint applies to both the blocked extension and the resolution part: a proof (Λ, Π) of Γ is without new variables if all the variables occurring in Λ or Π are in $\mathrm{var}(\Gamma)$. We use BC^-, RAT^-, SBC^-, and GER^- to denote the variants without new variables.

3 Lower bound for the bit pigeonhole principle

Let $n = 2^k$, with $k \in \mathbb{N}^+$. For a propositional variable v, let us write $v \neq 0$ and $v \neq 1$ to denote the literals v and $\overline{v}$, respectively. The *bit pigeonhole principle* is the contradiction stating that each pigeon in $[n+1]$ can be assigned a distinct binary string from $\{0,1\}^k$, where we identify strings with holes. For each pigeon $x \in [n+1]$, the variables $p_1^x, \ldots, p_k^x$ represent the bits of the string assigned to x. More formally, we write the bit pigeonhole principle as

$$\mathrm{BPHP}_n \coloneqq \bigcup_{\substack{x,y \in [n+1],\ x \neq y \\ (h_1,\ldots,h_k) \in \{0,1\}^k}} \left\{ \bigvee_{\ell=1}^{k} p_\ell^x \neq h_\ell \vee \bigvee_{\ell=1}^{k} p_\ell^y \neq h_\ell \right\},$$

which asserts that for all $x, y \in [n+1]$ such that $x \neq y$, the binary strings $p_1^x \ldots p_k^x$ and $p_1^y \ldots p_k^y$ are different.[7] We denote by P_x the set $\{p_1^x, \ldots, p_k^x, \overline{p_1^x}, \ldots, \overline{p_k^x}\}$ of all the literals concerning pigeon x.

For a set L of literals, its *pigeon-width* is the number of distinct pigeons it mentions, where a pigeon $x \in [n+1]$ is *mentioned* if there exists some $\ell \in [k]$ such that some literal of the variable p_ℓ^x is in L. We write $L(x)$ to denote the set $L \cap P_x$. In other words, $L(x)$ is the largest subset of L that mentions only the pigeon x.

---

[5] For an SBC proof to be polynomial-time verifiable, every step in the proof that adds a clause C as set-blocked is expected to indicate the subset $L \subseteq C$ for which C is set-blocked. With that said, we leave this requirement out of our definitions to reduce clutter.

[6] Instead of the original definition of a blocked extension [16, Definition 6.3], we use a convenient characterization [16, Lemma 6.5], which is simpler to state, as the definition.

[7] In our asymptotic results that use the bit pigeonhole principle, it is tacitly understood that BPHP_n could be defined for every integer $n \geq 2$ (as opposed to only powers of two) by letting BPHP_n be identical to BPHP_m, where m is the largest power of two not exceeding n.

Before proceeding with the SBC^- lower bound for the bit pigeonhole principle, we observe the result below, which will also be useful later. It is deduced in a straightforward way from the definition of a set-blocked clause, using the following fact: if a clause C is an SBC with respect to a formula Γ, then C is an SBC with respect to every subset of Γ. (Note that a similar result does not necessarily hold for RAT^-.)

▶ **Lemma 15.** *Without loss of generality, all of the set-blocked clause additions in an* SBC^- *proof are performed before any resolution or weakening steps.*

Lemma 15 allows us to reduce SBC^- lower bounds for a formula Γ to resolution lower bounds for another formula $\Gamma \cup \Sigma$, where Σ is a set of clauses derivable from Γ by a sequence of set-blocked clause additions without new variables.

A common strategy for proving resolution lower bounds is to first show that every proof contains some "complex" clause (in our case, a clause of large pigeon-width), and then argue that the existence of a small proof implies the existence of another proof where no clause is complex. The second step typically involves restricting the clauses of the proof under a suitable assignment. We work with assignments that correspond to partial matchings of pigeons to holes, as in the case of the RAT^- lower bound by Buss and Thapen [2, Section 5].

We say an assignment ρ that sets some variables of BPHP_n is a *partial matching* if ρ sets all of the bits for the pigeons it mentions in such a way that no two pigeons are in the same hole, thus representing a matching of pigeons to holes. To prove SBC^- lower bounds for BPHP_n, we will need the following pigeon-width lower bound for resolution proofs of restrictions of BPHP_n under partial matchings, which is established by a straightforward Adversary strategy in the Prover–Adversary game [19]. We define the pigeon-width of a proof as the maximum pigeon-width of any clause in the proof.

▶ **Lemma 16** ([2, Lemma 5.2])**.** *Let ρ be a partial matching of m pigeons to holes. Then every resolution proof of $(\mathrm{BPHP}_n)|_\rho$ has pigeon-width at least $n - m$.*

We will additionally need a pigeon-width lower bound for set-blocked clauses (without new variables) with respect to BPHP_n, which follows from a simple inspection.

▶ **Lemma 17.** *Every set-blocked clause with respect to BPHP_n that is without new variables has pigeon-width $n + 1$.*

Proof. Let $C = L \dot{\vee} C'$ be a set-blocked clause for L with respect to BPHP_n that is without new variables. Let x be a pigeon mentioned in L. Such a pigeon exists since L is nonempty. Let y be a pigeon different from x. We claim that C mentions y.

Let $D \in \mathrm{BPHP}_n$ be a clause that contains $\overline{L(x)} \cup C'(x)$ and mentions y. Such a clause exists since $\overline{L(x)} \cup C'(x)$ is simply a nontautological subset of P_x and, by the definition of BPHP_n, each such subset is contained in some clause in BPHP_n that mentions y. Note that every clause in BPHP_n mentions exactly two pigeons; in particular, the clause D mentions only the pigeons x and y.

Since D is a clause, it is nontautological. As a consequence, $D \cap L(x)$ is empty and $C'(x) \cup (D \setminus \overline{L})$ is nontautological. Then, since C is set-blocked for L with respect to BPHP_n, either $D \cap (L \setminus L(x))$ is nonempty or $(C' \setminus C'(x)) \cup (D \setminus \overline{L})$ is tautological. Now, neither of $L \setminus L(x)$ and $C' \setminus C'(x)$ mentions x. Since D mentions only the pigeons x and y, the pigeon y must be mentioned by L in the former case and C' in the latter. Either way, C mentions y. ◀

▶ **Theorem 18.** *The formula BPHP_n requires exponential-size proofs in SBC^-.*

Proof. Let Π be an SBC^- proof of BPHP_n of size N. By Lemma 15, we may view Π as a resolution proof of the formula $\mathrm{BPHP}_n \cup \Sigma$, where Σ is a set of clauses derivable from BPHP_n by a sequence of set-blocked clause additions without new variables. We will show by the probabilistic method that if $N < 2^{n/64}$, then there exists a partial matching ρ of $n/2$ pigeons to holes such that $(\mathrm{BPHP}_n)|_\rho$ has a resolution proof of pigeon-width strictly less than $n/2$.

Let R be a random partial matching constructed by choosing a random pigeon and assigning it to a random available hole until $n/2$ pigeons are matched to holes. We denote by r_i the random assignment performed at the ith step of this process: if pigeon x was assigned to hole $(h_1, \ldots, h_k)$, then $r_i(p^x_\ell) = h_\ell$ for all $\ell \in [k]$.

We say a clause is *wide* if it has pigeon-width at least $n/2$. Let C be a wide clause. Let x denote the ith pigeon chosen when constructing R, with $i \leq n/4$. The probability that x is mentioned by C is at least

$$\frac{n/2 - (n/4 - 1)}{n+1} \geq 1/4.$$

Suppose that x is mentioned by C through a literal p. When x is about to be assigned to a hole, there are at least $n/2-(n/4-1) \geq n/4$ available ones that would result in r_i satisfying p. Therefore, the conditional probability that r_i satisfies C given that the assignments $r_1, \ldots, r_{i-1}$ do not satisfy C is at least $1/16$. As a result,

$$\mathbf{Pr}[R \not\models C] < (1 - 1/16)^{n/4} \leq 2^{-n/64}.$$

Suppose that $N < 2^{n/64}$. Let Δ be the set of all the wide clauses appearing in Π. By the union bound, $\mathbf{Pr}[R \not\models \Delta] < 1$. Thus, there exists a partial matching ρ of $n/2$ pigeons to holes such that $\rho \models \Delta$. Also, observe that $\rho \models \Sigma$ because we have $\Sigma \subseteq \Delta$ by Lemma 17.

Since resolution is closed under restrictions, when we restrict the proof Π under ρ we obtain a resolution proof of $(\mathrm{BPHP}_n \cup \Sigma)|_\rho = (\mathrm{BPHP}_n)|_\rho$ without any wide clauses, which contradicts Lemma 16. ◀

4 Separations using guarded extension variables

From this point on, given a formula Γ, we use (Λ, Π) to denote the minimum-size ER proof of Γ,[8] where Λ is the union of a sequence of $t(\Gamma) \coloneqq |\Lambda|/3$ sets of extension clauses such that the ith set λ_i is of the form $\{\overline{x_i} \vee p_i,\ \overline{x_i} \vee q_i,\ x_i \vee \overline{p_i} \vee \overline{q_i}\}$. We thus reserve $\{x_1, \ldots, x_{t(\Gamma)}\}$ as the set of extension variables used in Λ. We assume without loss of generality that the variables of p_i and q_i are in $\mathrm{var}(\Gamma) \cup \{x_1, \ldots, x_{i-1}\}$ for all $i \in [t(\Gamma)]$.

4.1 Separation of RAT^- from SBC^-

Let Γ be a formula and (Λ, Π) be the minimum-size ER proof of Γ as described above. Consider the transformation

$$\mathcal{G}(\Gamma) \coloneqq \Gamma \cup \bigcup_{i=1}^{t(\Gamma)} \left[(x_i \vee \Gamma) \cup (\overline{x_i} \vee \Gamma)\right], \tag{1}$$

where $x_1, \ldots, x_{t(\Gamma)}$ are the extension variables used in Λ.

It becomes possible to prove $\mathcal{G}(\Gamma)$ in RAT^- by simulating the ER proof of Γ using the extension variables present in the formula, resulting in the following.

[8] We refer to *the* minimum-size proof with the assumption of having fixed some way of choosing a proof among those with minimum size.

▶ **Lemma 19** ([23, Lemma 5.1]). *For every formula* Γ, $\text{size}_{\mathsf{RAT}^-}(\mathcal{G}(\Gamma)) \leq \text{size}_{\mathsf{ER}}(\Gamma)$.

For SBC^-, the formula $\mathcal{G}(\Gamma)$ is at least as hard as Γ. We need the following definition before we prove this fact.

▶ **Definition 20.** *The* projection *of a formula* Γ *onto a literal* p *is the formula*

$$\text{proj}_p(\Gamma) \coloneqq \{C \setminus \{p\} \colon C \in \Gamma \text{ and } p \in C\}.$$

Our main tool in proving SBC^- lower bounds against the constructions that incorporate guarded extension variables is a version of the following characterization of blocked clauses, which was already observed by Kullmann (see [16, Section 4]).

▶ **Lemma 21** ([23, Lemma 3.15]). *A clause* $C = p \mathbin{\dot\vee} C'$ *is a BC for* p *with respect to a formula* Γ *if and only if the assignment* $\overline{C'}$ *satisfies* $\text{proj}_{\overline{p}}(\Gamma)$.

Lemma 21 suffices for proving BC^- lower bounds against $\mathcal{G}(\Gamma)$ (see [23, Lemma 5.2]); however, for SBC^- lower bounds we need the following result.

▶ **Lemma 22.** *If a clause* $C = L \mathbin{\dot\vee} C'$ *is an SBC for* L *with respect to a formula* Γ*, then for every* $p \in L$*, the assignment* $L \cup \overline{C'}$ *satisfies* $\text{proj}_{\overline{p}}(\Gamma)$.

Proof. Suppose that $C = L \mathbin{\dot\vee} C'$ is an SBC for L with respect to Γ, let $p \in L$, and let $D' \in \text{proj}_{\overline{p}}(\Gamma)$. Then the clause $D = \overline{p} \mathbin{\dot\vee} D'$ is in Γ. Note that $D \cap \overline{L}$ is nonempty. Then, since C is an SBC for L, either $D \cap L$ is nonempty or $C' \cup (D \setminus \overline{L})$ is tautological.

Case 1: $D \cap L \neq \varnothing$. Since $\overline{p} \notin L$, the set $D' \cap L$ also is nonempty; therefore $L \models D'$.

Case 2: $C' \cup (D \setminus \overline{L})$ *is tautological.* Since neither of C' and $D \setminus \overline{L}$ is tautological, their union is tautological if and only if $\overline{C'} \cap (D \setminus \overline{L})$ is nonempty. This implies in particular that $\overline{C'} \cap D'$ is nonempty; therefore $\overline{C'} \models D'$. ◀

Lemma 22 implies that if the projection of a formula onto a literal p is unsatisfiable, then, with respect to the formula, no clause is set-blocked for any set that contains $\overline{p}$.

The intuition behind the construction of $\mathcal{G}(\Gamma)$ is as follows. We incorporate the extension variables into the formula while having Γ be the projection for each added literal. Thus, if Γ is unsatisfiable, we render the extension variables useless in set-blocked clause additions with respect to $\mathcal{G}(\Gamma)$ while still allowing RAT^- to take advantage of them. In particular, it becomes unnecessary for a set-blocked clause C with respect to $\mathcal{G}(\Gamma)$ to include any of the extension variables present in the formula. This is because every such clause C has some subset C' without any of the extension variables that is still set-blocked with respect to $\mathcal{G}(\Gamma)$. Moreover, since $\Gamma \subseteq \mathcal{G}(\Gamma)$, the clause C' is set-blocked also with respect to Γ. The alternative way to use the extension variables in $\mathcal{G}(\Gamma)$ is to derive x_i from $x_i \vee \Gamma$, but this involves proving Γ. When Γ is hard for SBC^-, we leave no way for SBC^- to make any meaningful use of the extension variables to achieve a speedup. In the end, an SBC^- proof of $\mathcal{G}(\Gamma)$ might as well ignore the extension variables present in the formula, falling back to an SBC^- proof of Γ.

▶ **Lemma 23.** *For every formula* Γ, $\text{size}_{\mathsf{SBC}^-}(\mathcal{G}(\Gamma)) \geq \text{size}_{\mathsf{SBC}^-}(\Gamma)$.

Proof. When Γ is satisfiable, the inequality holds trivially, so suppose that Γ is unsatisfiable.

Suppose that $\mathcal{G}(\Gamma)$ has an SBC^- proof of size N. By Lemma 15, we may view such a proof as a resolution proof of the formula $\mathcal{G}(\Gamma) \cup \Sigma$, where Σ is a set of clauses derivable from $\mathcal{G}(\Gamma)$ by a sequence of set-blocked clause additions without new variables. Let $X = \{x_1, \ldots, x_{t(\Gamma)}\}$ denote the set of extension variables incorporated into $\mathcal{G}(\Gamma)$, and consider an assignment α defined as

$$\alpha(v) = \begin{cases} 1 & \text{if } v \in X \\ \text{undefined} & \text{otherwise.} \end{cases}$$

By Lemma 5, there exists a resolution proof of the formula $(\mathcal{G}(\Gamma) \cup \Sigma)|_\alpha = \Gamma \cup \Sigma|_\alpha$ of size at most $N - |\Sigma|$. We claim that the clauses in $\Sigma|_\alpha$ can be derived in sequence from Γ by set-blocked clause additions, which implies that there exists an SBC^- proof of Γ of size at most N.

Let $S = (C_1, \ldots, C_r)$ be the ordering in which the clauses of Σ are derived from $\mathcal{G}(\Gamma)$. We will show that if we restrict each clause in S under α and remove the satisfied clauses, then the remaining sequence of clauses can be derived from Γ in the same order by set-blocked clause additions. More specifically, the goal is to prove that for all $i \in [r]$ such that α does not satisfy C_i, the clause $C_i|_\alpha$ is set-blocked with respect to $\Gamma \cup \Phi_{i-1}|_\alpha$, where

$$\Phi_{i-1} \coloneqq \bigcup_{j \in [i-1]} \{C_j\}.$$

Let $i \in [r]$, and consider the clause C_i, which we write as C from this point on. Suppose that α does not satisfy C, so the variables from X can occur only negatively in C. Let $L \subseteq C$ be a subset for which C is set-blocked with respect to $\mathcal{G}(\Gamma) \cup \Phi_{i-1}$. We will prove that $C|_\alpha$ is set-blocked for $L|_\alpha$ with respect to both Γ and $\Phi_{i-1}|_\alpha$, which, by the definition of a set-blocked clause, implies that $C|_\alpha$ is set-blocked with respect to $\Gamma \cup \Phi_{i-1}|_\alpha$.

Before proceeding, observe that L cannot contain any variables from X: If some $\overline{x_i}$ is in L, then the assignment $L \cup \overline{(C \setminus L)}$ satisfies $\mathrm{proj}_{x_i}(\mathcal{G}(\Gamma)) = \Gamma$ by Lemma 22. Since Γ is unsatisfiable, no such assignment exists. Therefore, L cannot contain $\overline{x_i}$, which implies that $L|_\alpha = L$.

$C|_\alpha$ is set-blocked for L with respect to Γ: Since $\Gamma \subseteq \mathcal{G}(\Gamma)$, the clause C is set-blocked for L in particular with respect to Γ. Noting that the variables from X do not occur in Γ, we conclude that $C|_\alpha$ also is set-blocked for L with respect to Γ.

$C|_\alpha$ is set-blocked for L with respect to $\Phi_{i-1}|_\alpha$: Consider an arbitrary $D' \in \Phi_{i-1}|_\alpha$, which is the restriction under α of some clause $D \in \Phi_{i-1}$ that α does not satisfy. Suppose $D' \cap \overline{L} \neq \varnothing$ and $D' \cap L = \varnothing$. We need to show that $(C|_\alpha \setminus L) \cup (D' \setminus \overline{L})$ is tautological.

Since $D' \subseteq D$, we immediately have $D \cap \overline{L} \neq \varnothing$. Now, recall that the variables from X do not occur in L, and observe that D' is simply D with the variables from X removed. We thus have $D \cap L = \varnothing$. Then, because C is set-blocked for L with respect to Φ_{i-1}, the set $E = (C \setminus L) \cup (D \setminus \overline{L})$ must be tautological. A variable that occurs both positively and negatively in E cannot be from X, since in that case α would satisfy C or D. Therefore, the set $(C|_\alpha \setminus L) \cup (D' \setminus \overline{L})$ also is tautological. ◀

Invoking Lemmas 19 and 23 with Γ as the bit pigeonhole principle gives us the separation.

▶ **Theorem 24.** *The formula $\mathcal{G}(\mathrm{BPHP}_n)$ admits polynomial-size proofs in RAT^- but requires exponential-size proofs in SBC^-.*

Proof. Buss and Thapen [2, Theorem 4.4] gave polynomial-size proofs of BPHP_n in SPR^-, which ER simulates.[9] By Lemma 19, we have $\mathrm{size}_{\mathsf{RAT}^-}(\mathcal{G}(\mathrm{BPHP}_n)) = n^{O(1)}$. Theorem 18 and Lemma 23 give $\mathrm{size}_{\mathsf{SBC}^-}(\mathcal{G}(\mathrm{BPHP}_n)) = 2^{\Omega(n)}$. Thus, the bit pigeonhole principle with $\mathcal{G}$ applied to it exponentially separates RAT^- from SBC^-. ◀

[9] It is also possible to deduce the existence of polynomial-size ER proofs of BPHP_n from the fact that the pigeonhole principle (PHP_n) is easy for ER [4], combined with the observation that PHP_n can be derived from BPHP_n in polynomial size in ER.

4.2 Separation of GER$^-$ from SBC$^-$

We proceed in a similar way to the previous section. Let Γ be a formula and (Λ, Π) be the minimum-size ER proof of Γ. Let $m \in \mathbb{N}^+$, and let

$$\{y_1, \dots, y_m, z_1, \dots, z_m\} \subseteq \mathbf{V} \setminus \operatorname{var}(\Gamma \cup \Lambda)$$

be a set of $2m$ distinct variables. Consider

$$\begin{aligned} V_m(\Gamma) &\coloneqq \bigcup_{i=1}^{t(\Gamma)} \bigcup_{j=1}^{m} \{x_i \vee y_j \vee \overline{z_j},\ \overline{x_i} \vee y_j \vee \overline{z_j}\}, \\ W_m(\Gamma) &\coloneqq \bigcup_{j=1}^{m} \left[\{\overline{y_j} \vee z_j\} \cup (y_j \vee \Gamma) \cup (\overline{z_j} \vee \Gamma)\right], \\ \mathcal{I}_m(\Gamma) &\coloneqq \Gamma \cup V_m(\Gamma) \cup W_m(\Gamma), \end{aligned} \tag{2}$$

where $x_1, \dots, x_{t(\Gamma)}$ are the extension variables used in Λ.

To prove $\mathcal{I}_m(\Gamma)$ in GER$^-$ by simulating the ER proof of Γ, we essentially remove the clauses in $V_m(\Gamma)$, derive the extension clauses, and rederive $V_m(\Gamma)$ by a sequence of blocked clause additions.

▶ **Lemma 25.** *For every formula Γ and every $m \in \mathbb{N}^+$, $\operatorname{size}_{\mathsf{GER}^-}(\mathcal{I}_m(\Gamma)) \leq \operatorname{size}_{\mathsf{ER}}(\Gamma)$.*

Proof. Let (Λ, Π) be the minimum-size ER proof of Γ. We will show that the clauses in $\Lambda \cup V_m(\Gamma)$ can be derived from $\Gamma \cup W_m(\Gamma)$ in some sequence by blocked clause additions, which implies by Definition 13 that Λ is a blocked extension for $\mathcal{I}_m(\Gamma)$.

Recall that extension clauses can be derived in sequence by blocked clause additions. The formula Λ is an extension for $\Gamma \cup W_m(\Gamma)$, so we derive Λ by such a sequence. Next, from $\Gamma \cup W_m(\Gamma) \cup \Lambda$, we derive the clauses in $V_m(\Gamma)$ in any order. Let V' be a proper subset of $V_m(\Gamma)$, and let C be a clause in $V_m(\Gamma) \setminus V'$. For some $i \in [t(\Gamma)]$ and $j \in [m]$, the clause C is of the form $p \vee y_j \vee \overline{z_j}$, where p is either x_i or $\overline{x_i}$. With respect to $\Gamma \cup W_m(\Gamma) \cup \Lambda \cup V'$, the clause C is blocked for y_j since the only earlier occurrence of $\overline{y_j}$ is the clause $\overline{y_j} \vee z_j$ and $\{p, \overline{z_j}, z_j\}$ is tautological. It follows by induction that we can derive $V_m(\Gamma)$ from $\Gamma \cup W_m(\Gamma) \cup \Lambda$. Thus, Λ is a blocked extension for $\mathcal{I}_m(\Gamma)$.

Noting that Π is a resolution proof of $\Gamma \cup \Lambda$ and that $\mathcal{I}_m(\Gamma)$ contains Γ as a subset, we conclude that there exists a GER$^-$ proof of $\mathcal{I}_m(\Gamma)$ of size $|\Lambda| + |\Pi| = \operatorname{size}_{\mathsf{ER}}(\Gamma)$. ◀

For SBC$^-$, the formula $\mathcal{I}_m(\Gamma)$ stays at least as hard as Γ if fewer than 2^m set-blocked clauses are derived. As before, our goal is to render the added variables useless in set-blocked clause additions.

We will eventually choose Γ to be hard for SBC$^-$, which makes the literals $\overline{y_j}$ and z_j useless in set-blocked clause additions. Moreover, the presence of the clause $\overline{y_j} \vee z_j$ ensures that if a clause is set-blocked for a set containing y_j or $\overline{z_j}$, then the clause is a weakening of $y_j \vee \overline{z_j}$. Such clauses are killed by assignments that set y_j and z_j to the same value.

We also need to consider the clauses that are set-blocked for sets containing the variables x_i. The projection of $\mathcal{I}_m(\Gamma)$ onto x_i or $\overline{x_i}$ is the formula $\bigcup_{j=1}^{m} \{y_j \vee \overline{z_j}\}$, which has 2^m minimal satisfying assignments. Without deriving clauses that rule out all of those assignments, SBC$^-$ proofs cannot use the variables x_i in any meaningful way. Since the variables y_j and z_j are rendered useless in set-blocked clause additions, the assignments can only be ruled out one at a time, which forces SBC$^-$ proofs of $\mathcal{I}_m(\Gamma)$ to either derive at least 2^m clauses or ignore the variables x_i.

▶ **Lemma 26.** *For every formula* Γ *and every* $m \in \mathbb{N}^+$,

$$\text{size}_{\mathsf{SBC}^-}(\mathcal{I}_m(\Gamma)) \geq \min\{2^m, \text{size}_{\mathsf{SBC}^-}(\Gamma)\}.$$

Proof. Fix some $m \in \mathbb{N}^+$. When Γ is satisfiable, the inequality holds trivially, so suppose that Γ is unsatisfiable.

Suppose that $\mathcal{I}_m(\Gamma)$ has an SBC^- proof of size N. By Lemma 15, we may view such a proof as a resolution proof of the formula $\mathcal{I}_m(\Gamma) \cup \Sigma$, where Σ is a set of clauses derivable from $\mathcal{I}_m(\Gamma)$ by a sequence of set-blocked clause additions without new variables. We claim that if $|\Sigma| < 2^m$, then there exists an SBC^- proof of Γ of size at most N. This implies the desired lower bound.

Let $X = \{x_1, \ldots, x_{t(\Gamma)}\}$ and $U = \{y_1, \ldots, y_m, z_1, \ldots, z_m\}$ denote the two sets of variables incorporated into $\mathcal{I}_m(\Gamma)$. Consider a clause $C = L \dot{\vee} C'$ that is set-blocked for L with respect to $\mathcal{I}_m(\Gamma)$. We start by inspecting the ways in which the variables from $X \cup U$ can occur in L:

- We first consider the variables from U. If either $\overline{y_j}$ or z_j for some $j \in [m]$ occurs in L, then, by Lemma 22, the assignment $L \cup \overline{C'}$ satisfies Γ since Γ is contained in the projections of $\mathcal{I}_m(\Gamma)$ onto the negations of these literals. Thus, since Γ is unsatisfiable, neither of the literals $\overline{y_j}$ and z_j for any $j \in [m]$ can occur in L. Moreover, if the literal y_j occurs in L, then, by Lemma 22, the assignment $L \cup \overline{C'}$ satisfies z_j since the clause $\overline{y_j} \vee z_j$ is in $\mathcal{I}_m(\Gamma)$. In that case, since z_j cannot occur in L, the literal z_j must occur in $\overline{C'}$, which implies that C contains the literal $\overline{z_j}$. Thus, if the literal y_j occurs in L, then C contains the literal $\overline{z_j}$. By a similar argument, if the literal $\overline{z_j}$ occurs in L, then C contains the literal y_j. To summarize, if some variable from U is in L, then C is a weakening of some clause in $\bigcup_{j=1}^{m} \{y_j \vee \overline{z_j}\}$.
- Next, we consider the variables from X. For all $j \in [m]$, define $A_j \coloneqq \{y_j, \overline{z_j}\}$ (intended to be viewed as an assignment). Let $\mathbf{A} = A_1 \times \cdots \times A_m$. Suppose that a literal p of some x_i is in L. Then, by Lemma 22, the assignment $L \cup \overline{C'}$ satisfies the formula

$$\text{proj}_{\overline{p}}(\mathcal{I}_m(\Gamma)) = \bigcup_{j=1}^{m} \{y_j \vee \overline{z_j}\}.$$

  This implies that if no variable from U occurs in L, then there exists some assignment $\beta \in \mathbf{A}$ such that $\beta \subseteq \overline{C'}$. We say C is a *good* clause if some variable from X occurs in L but no variable from U occurs in L.

From this point on, suppose $|\Sigma| < 2^m$. For each good clause E in Σ, choose a single subset $F \subseteq E$ such that $\overline{F} \in \mathbf{A}$. Let Δ be the collection of those subsets. Since $|\Delta| < 2^m$, there exists some $\beta \in \mathbf{A}$ such that $\overline{\beta} \notin \Delta$. Recall that for each $j \in [m]$, the assignment β sets exactly one of the variables y_j and z_j. Let β' be the smallest assignment extending β such that $\beta'(y_j) = \beta'(z_j)$ for all $j \in [m]$.

▷ Claim 27. Let C be a clause in Σ, and let $L \subseteq C$ be a subset for which C is set-blocked with respect to $\mathcal{I}_m(\Gamma)$. If some variable from $X \cup U$ occurs in L, then β' satisfies C.

Proof. Let C be a clause in Σ set-blocked for $L \subseteq C$ with respect to $\mathcal{I}_m(\Gamma)$. Suppose that some variable from $X \cup U$ occurs in L. Then either $\text{var}(L) \cap U \neq \varnothing$ or C is a good clause.

Case 1: $\text{var}(L) \cap U \neq \varnothing$. Since C is a weakening of some clause in $\bigcup_{j=1}^{m} \{y_j \vee \overline{z_j}\}$ and $\beta'(y_j) = \beta'(z_j)$ for all $j \in [m]$, the assignment β' satisfies C.

Case 2: C is a good clause. Let F be a subset of C such that $F \in \Delta$. Since $\overline{\beta} \notin \Delta$, there exists some $j \in [m]$ such that either $\overline{y_j} \in \overline{\beta}$ and $z_j \in F$ or $z_j \in \overline{\beta}$ and $\overline{y_j} \in F$. We have $\beta'(z_j) = 1$ in the former case and $\beta'(y_j) = 0$ in the latter. Either way, β' satisfies F and hence it also satisfies C. ◁

Now, let α be the assignment defined as

$$\alpha(v) = \begin{cases} 1 & \text{if } v \in X \\ \beta'(v) & \text{if } v \in U \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The point of α is to set all of the variables from $X \cup U$ in such a way that kills all of the clauses in Σ that are set-blocked for sets containing those variables, leaving behind only the clauses that could also be derived in an SBC^- proof of Γ.

The rest of the argument is similar at a high level to the proof of Lemma 23, so we will be relatively brief. By Lemma 5, there exists a resolution proof of the formula $(\mathcal{I}_m(\Gamma) \cup \Sigma)|_\alpha = \Gamma \cup \Sigma|_\alpha$ of size at most $N - |\Sigma|$. We claim that the clauses in $\Sigma|_\alpha$ can be derived in sequence from Γ by set-blocked clause additions.

As before, let $S = (C_1, \dots, C_r)$ be the ordering in which the clauses of Σ are derived from $\mathcal{I}_m(\Gamma)$. We will prove that for all $i \in [r]$ such that α does not satisfy C_i, the clause $C_i|_\alpha$ is set-blocked with respect to $\Gamma \cup \Phi_{i-1}|_\alpha$, where

$$\Phi_{i-1} \coloneqq \bigcup_{j \in [i-1]} \{C_j\}.$$

Let $i \in [r]$, and consider the clause C_i, which we write as C from this point on. Let $L \subseteq C$ be a subset for which C is set-blocked with respect to $\mathcal{I}_m(\Gamma) \cup \Phi_{i-1}$. Suppose that α does not satisfy C. Noting that α extends β', by Claim 27, no variable from $X \cup U$ is in L. As a result, $L|_\alpha = L$. We will prove that $C|_\alpha$ is set-blocked for L with respect to both Γ and $\Phi_{i-1}|_\alpha$.

$C|_\alpha$ is set-blocked for L with respect to Γ: Since $\Gamma \subseteq \mathcal{I}_m(\Gamma)$, the clause C is set-blocked for L in particular with respect to Γ. No variable from $X \cup U$ occurs in Γ, so the clause $C|_\alpha$ also is set-blocked for L with respect to Γ.

$C|_\alpha$ is set-blocked for L with respect to $\Phi_{i-1}|_\alpha$: Consider an arbitrary $D' \in \Phi_{i-1}|_\alpha$, which is the restriction under α of some clause $D \in \Phi_{i-1}$ that α does not satisfy. Suppose $D' \cap \overline{L} \neq \varnothing$ and $D' \cap L = \varnothing$. We need to show that $(C|_\alpha \setminus L) \cup (D' \setminus \overline{L})$ is tautological.

We have $D \cap \overline{L} \neq \varnothing$ because $D' \subseteq D$. Recall that no variable from $X \cup U$ is in L, and observe that D' is simply D with the variables from $X \cup U$ removed. This implies $D \cap L = \varnothing$. Now, because C is set-blocked for L with respect to Φ_{i-1}, the set $E = (C \setminus L) \cup (D \setminus \overline{L})$ must be tautological. A variable that occurs both positively and negatively in E cannot be from $X \cup U$, since in that case α would satisfy C or D. Therefore, the set $(C|_\alpha \setminus L) \cup (D' \setminus \overline{L})$ also is tautological. ◀

Invoking Lemmas 25 and 26 with a suitable choice of m and with Γ as the bit pigeonhole principle gives us the separation.

▶ **Theorem 28.** *For every unsatisfiable formula Γ, let $m(\Gamma) \coloneqq \lceil \log(\text{size}_{\mathsf{SBC}^-}(\Gamma)) \rceil$ and define $\mathcal{K}(\Gamma) \coloneqq \mathcal{I}_{m(\Gamma)}(\Gamma)$. The formula $\mathcal{K}(\text{BPHP}_n)$ admits polynomial-size proofs in GER^- but requires exponential-size proofs in SBC^-.*

Proof. Buss and Thapen [2, Theorem 4.4] gave polynomial-size proofs of BPHP_n in SPR^-, which ER simulates. By Lemma 25, we have $\text{size}_{\mathsf{GER}^-}(\mathcal{K}(\text{BPHP}_n)) = n^{O(1)}$. Theorem 18 and Lemma 26 give $\text{size}_{\mathsf{SBC}^-}(\mathcal{K}(\text{BPHP}_n)) = 2^{\Omega(n)}$. Thus, the bit pigeonhole principle with $\mathcal{K}$ applied to it exponentially separates GER^- from SBC^-. ◀

References

1 Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, The University of Chicago, 1937.

2 Sam Buss and Neil Thapen. DRAT and propagation redundancy proofs without new variables. *Logical Methods in Computer Science*, 17(2):12:1–12:31, 2021. doi:10.23638/LMCS-17(2:12)2021.

3 Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Symposium on Theory of Computing (STOC)*, pages 174–183. Association for Computing Machinery, 1996. doi:10.1145/237814.237860.

4 Stephen A. Cook. A short proof of the pigeon hole principle using extended resolution. *ACM SIGACT News*, 8(4):28–32, 1976. doi:10.1145/1008335.1008338.

5 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979. doi:10.2307/2273702.

6 William Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987. doi:10.1016/0166-218X(87)90039-4.

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

8 Marijn J. H. Heule, Benjamin Kiesl, and Armin Biere. Encoding redundancy for satisfaction-driven clause learning. In *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, number 11427 in Lecture Notes in Computer Science, pages 41–58. Springer, 2019. doi:10.1007/978-3-030-17462-0_3.

9 Marijn J. H. Heule, Benjamin Kiesl, and Armin Biere. Strong extension-free proof systems. *Journal of Automated Reasoning*, 64(3):533–554, 2020. doi:10.1007/s10817-019-09516-0.

10 Marijn J. H. Heule, Benjamin Kiesl, Martina Seidl, and Armin Biere. PRuning through satisfaction. In *Proceedings of the 13th Haifa Verification Conference (HVC)*, number 10629 in Lecture Notes in Computer Science, pages 179–194. Springer, 2017. doi:10.1007/978-3-319-70389-3_12.

11 Matti Järvisalo, Marijn J. H. Heule, and Armin Biere. Inprocessing rules. In *Proceedings of the 6th International Joint Conference on Automated Reasoning (IJCAR)*, number 7364 in Lecture Notes in Computer Science, pages 355–370. Springer, 2012. doi:10.1007/978-3-642-31365-3_28.

12 Benjamin Kiesl, Martina Seidl, Hans Tompits, and Armin Biere. Local redundancy in SAT: Generalizations of blocked clauses. *Logical Methods in Computer Science*, 14(4:3):1–23, 2018. doi:10.23638/LMCS-14(4:3)2018.

13 Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1–2):123–140, 2001.

14 Oliver Kullmann. Worst-case analysis, 3-SAT decision and lower bounds: Approaches for improved SAT algorithms. In *Satisfiability Problem: Theory and Applications*, number 35 in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 261–313. American Mathematical Society, 1997. doi:10.1090/dimacs/035.

15 Oliver Kullmann. New methods for 3-SAT decision and worst-case analysis. *Theoretical Computer Science*, 223(1–2):1–72, 1999. doi:10.1016/S0304-3975(98)00017-6.

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

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

18 Toniann Pitassi and Rahul Santhanam. Effectively polynomial simulations. In *Proceedings of the 1st Innovations in Computer Science (ICS)*, pages 370–382. Tsinghua University Press, 2010.

19 Pavel Pudlák. Proofs as games. *The American Mathematical Monthly*, 107(6):541–550, 2000. doi:10.2307/2589349.

20 John A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965. doi:10.1145/321250.321253.

21 Grigori S. Tseitin. On the complexity of derivation in propositional calculus. *Zapiski Nauchnykh Seminarov LOMI*, 8:234–259, 1968.

22 Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987. doi:10.1145/7531.8928.

23 Emre Yolcu and Marijn J. H. Heule. Exponential separations using guarded extension variables. In *Proceedings of the 14th Innovations in Theoretical Computer Science (ITCS)*, number 251 in Leibniz International Proceedings in Informatics, pages 101:1–101:22. Schloss Dagstuhl, 2023. doi:10.4230/LIPIcs.ITCS.2023.101.