The Power of Small Symmetries

Nikita Gaevoy
The Faculty of Data and Decision Sciences, Technion, Haifa, Israel

Abstract

Resolution with symmetries is a natural extension of the Resolution proof system that allows to use symmetries of the formula to simplify the proof. Symmetries can be global (applied to the whole input formula), local (applied to a subformula), or dynamic (applied to newly derived clauses as well). The framework of Resolution with (global) symmetries was introduced by Krishnamurthy [16] and further extended by Arai and Urquhart [1] to local symmetries. Later, Szeider [20] generalized this approach to homomorphisms and introduced the notion of Resolution with dynamic symmetries. While proving superpolynomial proof-size lower bounds for Resolution with dynamic symmetries remains an open problem already for two decades, the power of proof systems with global and local symmetries is well studied: exponential lower bounds have been proven for these proof systems, as well as exponential separations between all of them. However, these systems are too general to reflect practical applications since it is computationally too hard to find and efficiently exploit arbitrary symmetries.

In this work, we introduce the notion of small symmetries: symmetries that can operate on a limited number of variables at the same time. Resolution with small symmetries gives hopes both for practical applications and for theoretical study of dynamic symmetries. We show that proof systems with both local and global small symmetries form strict hierarchies w.r.t. the size of symmetries. We prove exponential separations between proof systems with symmetries of different sizes and types. It turns out that even lower levels of these hierarchies are exponentially separated from Resolution and stronger proof systems, such as constant-depth Frege. As a byproduct of our constructions, we obtain an exponential separation between the classical systems SRC-I and SR-II that was not known before.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases proof complexity, complexity lower bounds, resolution with symmetries, small symmetries

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.97

Related Version *Full Version*: `https://arxiv.org/abs/2606.25895` [6]

Acknowledgements The author is very grateful to Edward A. Hirsch and Ofer Strichman, who supervised this work, in particular, introduced the author to the use of symmetries in proof search, guided him through numerous discussions, and gave invaluable comments on the presentation of the results. The author also thanks Arthur Riazanov for fruitful discussions.

1 Introduction

Many families of formulas based on combinatorial principles that are hard for Resolution, including pigeonhole and Tseitin formulas [2, 7, 21], have a very symmetric nature. One of the first theoretical approaches to use these symmetries to optimize the length of Resolution-like proofs was Krishnamurthy's symmetry rule [16]. This rule allows one to infer a clause symmetric to a previously deduced clause in one step, if the formula itself is symmetric under the same symmetry. Resolution with this additional rule is called Resolution with global (SR-I) or local (SR-II) symmetries, depending on whether the whole input formula or only a part must be symmetric. In both cases, symmetries are applied only to the original formula (not inferred clauses), so we call them *static*. Urquhart [22] generalized these proof systems by introducing Resolution with *complementary* symmetries (SRC-I and SRC-II) that additionally allowed to interchange literals with their negations.

Table 1 Resolutions with symmetries of varying degree of generality and corresponding exponential lower bounds. The strength of proof systems increases from top to bottom and from left to right.

	global	local	dynamic
symmetries	SR-I [22]	SR-II [1]	SR-III (open)
symmetries + complementation	SRC-I [22]	SRC-II [1]	SRC-III (open)
homomorphisms	HR-I [20]	HR-II [20]	HR-III (open)

These systems were further extended by Szeider [20] in two important directions: Resolution with homomorphisms (HR-I and HR-II) and Resolution with *dynamic* symmetries (SR-III, SRC-III, and HR-III). Resolution with homomorphisms allows the usage of any homomorphism of a formula in place of a symmetry, which makes HR-I and HR-II even more general than their SR and SRC counterparts. Resolution with dynamic symmetries (homomorphisms) is an extension that allows using inferred clauses in the symmetry rule. This yields three new proof systems: SR-III, SRC-III, and HR-III, depending on whether we are allowed to use symmetries, complementary symmetries, or homomorphisms.

1.1 Previous work

For all six proof systems with global and local symmetries and homomorphisms, exponential lower bounds on the length of their proofs are known (see Table 1). Moreover, for Resolution with local homomorphisms (HR-II), the strongest proof system among these six, Szeider [20] showed that every formula could be transformed so that the lower bounds for the original formula in Resolution could be translated into lower bounds for the new version of the formula in HR-II. However, it remains an open problem to prove superpolynomial lower bounds on proof length even in SR-III, the weakest of the three proof systems with dynamic symmetries. In fact, Resolution with dynamic symmetries is so much stronger than Resolution with static (i.e., global or local) symmetries that, for some formulas, it is interesting whether using static symmetries only is sufficient to achieve polynomial proof length [19]. The same question of what weaker symmetries can achieve motivates studying proof systems that lie even below SR-I [18].

Despite the theoretical nature of this approach, some restricted versions of these proof systems could be found in practical SAT solvers. Several such methods are jointly called dynamic symmetry handling [5] or dynamic symmetry exploitation [12], as opposed to symmetry breaking, the collection of methods that can add clauses to a formula that may affect the set of satisfying assignments as long as the satisfiability stays intact. Symmetry breaking approaches also could be formalized and are promising for studying in the framework of proof systems [3]. Although most solvers that use dynamic symmetry handling lose the competition to solvers with modern symmetry-breaking heuristics [5], the latter approach requires the symmetries to be essentially global. In contrast, dynamic symmetry handling heuristics may efficiently work with local symmetries, which is crucial for almost symmetric formulas. Moreover, in some practical cases, such as problems that emerge from bounded model checking, local symmetries arise naturally and could be generated together with the input instance of the satisfiability problem. It allows the solver to have a sufficient amount of symmetries to work with without spending much time finding those symmetries, which can significantly improve the performance [12]. In Ramsey theory, many state-of-the-art results are obtained using SAT solvers [8, 9, 13, 14]. Although these solvers frequently employ symmetry breaking, the input formulas naturally contain many local symmetries, suggesting that their use could be a promising direction for further improvement.

1.2 Our contribution

In this work, we introduce a notion of small symmetries, that is, symmetries operating only on a small number (this number is a parameter of the proof system) of variables at the same time. This defines a set of new proof systems that play an intermediate role between Resolution and the original proof systems with symmetries and homomorphisms. A system with k-symmetries (where k could be a constant or a function) is obtained from a system with arbitrary symmetries by restricting all applications of the symmetry rule to symmetries that nontrivially permute at most k variables at a time (for formal definitions refer to Section 2). We use the notation k-Π to denote the proof system Π with the restriction on k-symmetries (e.g., k-SR-I, k-SRC-II). The notion of small symmetries arises naturally in the context of practical SAT solving, where it is much harder to detect large symmetries than small ones. Additionally, many formulas have very symmetric nature, with many small symmetries, but almost no large symmetries. Examples include formulas constructed from the bounded model checking instances and formulas encoding statements from Ramsey theory. In the latter case, one can often find symmetries comparable in the size to the forbidden structure, which is often much smaller than the number of variables.

For dynamic symmetries, the notion of small symmetries allows us to define natural restricted versions of the proof system, which could be easier to prove lower bounds for. It is important to note that even the most restricted system 1-SRC-III (1-SR-III is trivially equivalent to Resolution) is probably strictly stronger than Resolution, even though we don't have any nontrivial upper or lower bounds on it. In all three cases of global, local, and dynamic symmetries, it is natural to ask how the power of Resolution with small symmetries depends on the number of variables that the symmetries are allowed to operate on.

We focus on the case of small static symmetries. We consider two different families of proof systems: Resolution with constant-size symmetries and Resolution with logarithmic-size symmetries. In the case of logarithmic symmetries, we prove that $(2\log n)$-SR-I has a polynomial proof for the formula bPHP_n representing the binary encoding of the pigeonhole principle, which is hard for Resolution. We also use this bound to show that $(k+1)$-SR-I has exponentially shorter proofs for some formulas than k-SRC-II, where k is odd and at least logarithmic in n. In the case of constant symmetries, we use Tseitin formulas to obtain an exponential separation between 3-SRC-I and Res to show that for any k, $\mathcal{O}(k)$-SR-I has exponentially shorter proofs for some formulas than k-SRC-II (see Figure 6 for the precise relations between proof systems).

The hierarchy of proof systems with small symmetries is three-dimensional: we may increase the size of symmetries, add complementary symmetries, or generalize the type of used symmetries. Therefore, there are three different types of separations that we can obtain, showing that the increase in one parameter makes the system stronger on some formulas, even if the other two decrease. In our case, we use the separations that increase the size of symmetries to obtain the results of the other two types, and show that these separations hold even if the resulting system is allowed to operate with constant symmetries only (see Figure 8 for the diagram). As a byproduct of these results, we also obtain an exponential separation between classical systems SRC-I and SR-II that was not known before (see Corollary 35).

1.3 Our methods

All previous separations between proof systems with symmetries share the same structure: take a formula with a symmetric structure and then modify it to break one type of symmetries, but maintain another. The base formulas in these separations are the propositional pigeonhole

principle or perfect matching [20, 22], or formulas that use them as a gadget [1]. Unfortunately, these formulas allow one to construct proofs with large (polynomial-size) symmetries only. The second problem arises from the fact that, although they allow using very broad families of formulas, the transformations produce only formulas with large symmetries.

Our separations follow the same path. However, to overcome the limitations of the previous separations, we consider different families of formulas. For each family, we construct polynomial-size refutations using only small symmetries. We then apply transformations designed specifically for these formula types, which allow us to "lift" the lower bounds for the Resolution proof system without a significant growth in the size of the required symmetries.

Upper bounds. Our basic formulas are the binary encoding of the pigeonhole principle and Tseitin formulas. For both families of formulas we construct refutations that have a dynamic programming-like nature.

For the refutation of bPHP_n we exploit symmetries swapping rows to assume without any loss in the generality that the first column starts with either $\frac{n+1}{2}$ zeros or $\frac{n+1}{2}$ ones. This is sufficient to reduce bPHP_n to a smaller formula $\mathsf{bPHP}_{n/2}$, which we solve recursively.

The refutation of Tseitin formulas on grids has a more complex structure, resembling the folklore technique of dynamic programming on broken profile. We reduce the problem on toroidal grids to the problem on the planar grid, where we structure our proof similarly to a dynamic programming solution to the problem of counting the number of satisfying assignments. It allows us to partition all possible assignments into a constant number of symmetric classes, which leads to the polynomial-size proof. Although we consider grids, as they produce the best constants, this technique is applicable to Tseitin formulas built on top of any family of graphs that have a lot of small and well connected cycles.

Separations. For the separations, we transform our basic formulas by embedding extra variables in the formula structure, thereby increasing the size of the smallest "useful" symmetry. Although the size of the symmetries required to efficiently refute bPHP_n is logarithmic in n, the transformation that works with this family of formulas allows us to separate systems where the symmetry sizes differ by as little as 1. At the same time, Tseitin formulas have the advantage of being polynomially solved using only constant symmetries, but the transformation lifting the lower bounds for local symmetries leads to a multiplicative blow-up in the size of symmetries in the upper bound. However, this still allow us to separate k-SRC-I from $(k+1)$-SRC-I and separate bigger global symmetries from smaller local ones, which would not be possible using previous approaches. Our separations of proof systems with small symmetries from systems with larger symmetries of different type are also based on the Tseitin formulas. They share the same core idea, but use more complex constructions.

1.4 Organization of the paper

In Section 3, we consider formulas based on binary encoding of the pigeonhole principle. We show that Resolution with global logarithmic symmetries admits polynomial-size proofs for the formula bPHP, which has exponential lower bounds in Resolution and in stronger systems such as constant-depth Frege. Then, we use this bound to study the hierarchy of proof systems with symmetries of at least logarithmic size (see Figure 1).

Section 4 is dedicated to formulas built on top of Tseitin contradictions on two types of grids, rectangular and triangular. Using these formulas, we show that Resolution with static symmetries may have short proofs for formulas hard for systems like constant-depth Frege even when the bound on the size of symmetries is as small as the constant three. Then, we

use these bounds to obtain the results on the hierarchy of proof systems with symmetries that are smaller than logarithmic (see Figure 6). For the sake of simplicity, we construct some of our proofs for the case of rectangular grid first, obtaining worse parameters than it is possible to obtain in the triangular case (see Figure 5), and then generalize them to the triangular grid.

In Section 5, we consider separations in the directions opposite to those proven in the previous sections and obtain the relations between proof systems with the smallest symmetry constraints (see Figure 8).

2 Preliminaries

▶ **Definition 1.** *A mapping φ of literals into literals is a* k-symmetry *if and only if it satisfies the following two conditions:*

1. $\varphi(\neg x) = \neg\varphi(x)$.
2. *All variables except for at most k are mapped into themselves (i.e. $\varphi(x) = x$).*

We call a k-symmetry positive *if it maps variables into variables without negation. Symmetries that are not necessarily positive we call* complementary *symmetries. In case k is equal to the number of variables, we omit the parameter k and obtain the classical definitions of symmetries [16, 22].*

▶ **Definition 2.** *Let $k \geq 1$ be a constant or a function of the number of variables.* Resolution with dynamic k-symmetries (k-SRC-III) *is a proof system with the following rules:*

1. *Resolution rule.*

$$\frac{A \vee x \qquad B \vee \neg x}{A \vee B}$$

The variable x is called the pivot *of the inference.*

2. *Symmetry rule.*

$$\frac{\varphi(B_1), \varphi(B_2), \ldots, \varphi(B_l)}{\varphi(A)}\ \varphi$$

where φ is a k-symmetry and A was inferred from $B_1, B_2, \ldots, B_l$ before. Additionally, each time this rule is applied, we also write down φ, for the purposes of verification. We call the inference of A from $B_1, B_2, \ldots, B_l$ the justification of the symmetry rule, $\varphi(B_1), \varphi(B_2), \ldots, \varphi(B_l) \vdash \varphi(A)$ the result of the symmetry rule, and $\varphi(A)$ the resulting clause.

Similarly to Resolution, the proof is a sequence of clauses, where each clause is either an input clause or is inferred from previous clauses by one of the rules above, and the last clause is the empty clause.

The motivation behind the symmetry rule is as follows. If A was inferred from $B_1, \ldots, B_l$, and there is a k-symmetry φ mapping all premises to already derived clauses, then applying φ to the proof yields an inference of $\varphi(A)$ from $\varphi(B_1), \ldots, \varphi(B_l)$. Hence, instead of repeating the proof, we may infer $\varphi(A)$ in a single step.

▶ Remark 3. It is possible to define k-homomorphisms and the corresponding proof systems. Moreover, some of results could be generalized to homomorphisms – either directly or with some additional technical work. In this work, however, we restrict our attention to symmetries.

We define Resolution with k-symmetries as a system with dynamic symmetries. That is, the proof is not restricted to symmetries of the original formula and may instead use symmetries that appear after several inference steps, or in a more complex manner. However, it is easy to define versions of Resolution with k-symmetries that use only static (i.e. global or local) symmetries.

▶ **Definition 4.** Resolution with local k-symmetries (k-SRC-II) *is a Resolution with k-symmetries with an additional requirement that, in the symmetry rule, all B_i and all $\varphi(B_i)$ are present in the input formula.*

Resolution with global k-symmetries (k-SRC-I) *is a Resolution with k-symmetries with an additional requirement that, in the symmetry rule, both sets $\{B_i\}$ and $\{\varphi(B_i)\}$ are equal to the set of all input clauses.*

If n is the number of variables, n-SRC-III is SRC-III as defined in [20] and n-SRC-I and n-SRC-II correspond to SRC-I and SRC-II as defined in [22], respectively.

▶ **Definition 5.** k-SR-I, k-SR-II, and k-SR-III *are the proof systems k-SRC-I, k-SRC-II, and k-SRC-III, respectively, with an additional requirement for all symmetries to be positive.*

Analogously, if k is set to n, we get the original definitions of SR-I, SR-II, and SR-III [16, 20]. Now we need to show that all the proof systems above are valid proof systems in the sense of the following definition of Cook and Reckhow [4].

▶ **Definition 6.** *A* proof system *for CNF is a polynomial-time algorithm* Π *that accepts a pair of a CNF-formula* Φ *and a binary string (i.e. refutation or proof)* ξ *and satisfies the following conditions.*

- *Completeness: for every unsatisfiable* Φ *there exists a refutation* ξ *such that* $\Pi(\Phi, \xi) = 1$.
- *Soundness: if* Φ *does not encode an unsatisfiable formula, then* $\Pi(\Phi, \xi) = 0$ *for any* ξ.

▶ **Lemma 7.** *All defined systems are Cook-Reckhow proof systems.*

Proof. The completeness follow from the fact that all systems are extensions of Resolution, which is a Cook-Reckhow proof system. Polynomial-time verification is possible since we write down the used symmetry function each time the symmetry rule is applied. And the soundness follows from the requirement for the justification to be inferred before the application of the symmetry rule, so the result of the symmetry rule always encodes a valid inference. ◀

Similarly to Resolution, the size of a proof in systems with symmetries could be measured as both the number of steps and the total length of the proof. However, those metrics differ only by a polynomial factor, so we will consider only the number of steps.

▶ **Definition 8.** $S_\Pi(\Phi)$ *is the number of steps in the shortest proof of an unsatisfiable formula* Φ *in proof system* Π.

Note that due to the symmetry rule, the size of a proof may be even smaller than the number of clauses in the formula. An example of such a situation will be given in Theorem 15. Another classical notion we operate with is the notion of p-simulation.

▶ **Definition 9.** *A proof system A* p-simulates *a proof system B if there exists a polynomial-time function f such that $f(\Phi, \xi)$ is a valid A-proof of Φ whenever ξ is a valid B-proof of Φ. In other words, $B(\Phi, \xi) = 1 \Rightarrow A(\Phi, f(\Phi, \xi)) = 1$, for all Φ, ξ.*

Two systems simulating each other are called p-equivalent. *Proof system A is* exponentially separated *from B if there exists a family of formulas Φ_n such that $S_A(\Phi_n) = \mathcal{O}(\mathsf{poly}(n))$ and $S_B(\Phi_n) = 2^{\Omega(n)}$.*

▶ Remark 10. A system with small symmetries of a given type is trivially p-simulated by systems with the larger symmetries of the same type. However, separating such systems is nontrivial and will be one of our main goals in this work.

In our definition, Resolution with k-symmetries does not have the weakening rule:

$$\frac{A}{A \vee x}.$$

Addition of the weakening rule may potentially make Resolution with small dynamic symmetries stronger. However, in the static case, it was proven by Szeider [20] that we can eliminate all applications of the weakening rule similarly to Resolution, and this result can be directly extended to all versions of Resolution with small static symmetries.

▶ Remark 11. For all k, the addition of the weakening rule to k-SR-I, k-SRC-I, k-SR-II, and k-SRC-II does not change the length of the shortest proof.

Since we prove only upper bounds on systems with dynamic symmetries, we do not include the weakening rule in the definition.

▶ **Lemma 12.** *1-SRC-I is p-equivalent to Res.*

Proof. See the full version of the paper. ◀

The approach in this lemma cannot be extended to 1-SRC-II because removal of variables may break some symmetries. Moreover, it is likely that 1-SRC-II is strictly stronger than Resolution.

3 Binary PHP

In this section, we consider formulas based on the binary pigeonhole principle and prove the relations between proof systems shown in Figure 1.

▶ **Definition 13.** *Let n be a power of 2.* bPHP_n *is the following CNF-formula in variables $x_{1,1}, x_{1,2}, \dots, x_{1,\log n}, x_{2,1}, \dots, x_{n+1,\log n}$:*

$$\bigwedge_{\substack{i<j \\ 0\le y<n}} (\overline{x_{i,1} \dots x_{i,\log n}} \neq y) \vee (\overline{x_{j,1} \dots x_{j,\log n}} \neq y).$$

Here $\overline{x_1 x_2 \dots x_k}$ denotes the integer number whose binary representation is $x_1 x_2 \dots x_k$ and log *denotes the binary logarithm. Note that the non-equalities in the formula above can be expressed by disjunctions, so the formula is in CNF already.*

Informally, bPHP_n encodes the statement that $n+1$ pigeons cannot be placed into n holes, where each pigeon i is assigned a hole by its binary string $\overline{x_{i,1} \dots x_{i,\log n}}$. Each clause encodes the constraint that two distinct pigeons i and j cannot both be mapped to the same hole y.

bPHP_n consists of $\mathcal{O}(n \log n)$ variables and $\mathcal{O}(n^3)$ clauses. The formula is unsatisfiable and does not change when we swap any two rows of the matrix $(x_{i,j})$. bPHP is hard not only for Resolution, but also for stronger proof systems. For example, bPHP is hard for bounded-depth Frege, since the pigeonhole principle with functional and onto axioms (ofPHP) is hard for it [15, 17], and it is easy to derive axioms of bPHP from ofPHP in bounded depth.

▶ **Theorem 14** ([15, 17]). $S_{\textit{depth-d Frege}}(\mathsf{ofPHP}_n) = \exp\left(n^{\exp(-\mathcal{O}(d))}\right)$

Table 2 Variables of bPHP_8.

$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$
$x_{5,1}$	$x_{5,2}$	$x_{5,3}$
$x_{6,1}$	$x_{6,2}$	$x_{6,3}$
$x_{7,1}$	$x_{7,2}$	$x_{7,3}$
$x_{8,1}$	$x_{8,2}$	$x_{8,3}$
$x_{9,1}$	$x_{9,2}$	$x_{9,3}$

▶ **Theorem 15.** $S_{2\log n\text{-}SR\text{-}I}(\mathsf{bPHP}_n) = \mathcal{O}(n^2)$

Proof. We only use symmetries to swap rows of the matrix $(x_{i,j})$ in the formula. It is clear that each such symmetry step is a valid step in our proof system because the formula itself is symmetrical on swapping rows.

Our plan is to infer two clauses stating that the first column of the matrix $(x_{i,j})$ (green part in Table 2) cannot start with $n/2+1$ zeros nor $n/2+1$ ones and then use symmetries to obtain a contradiction from them. The last step is possible because every assignment to the first column contains either at least $n/2+1$ zeros or at least $n/2+1$ ones, and after swapping some rows we can put these values to the first $n/2+1$ rows. The construction is recursive.

First of all, we want to infer clauses $(x_{1,1} \vee x_{2,1} \vee \cdots \vee x_{n/2+1,1})$ and $(\neg x_{1,1} \vee \neg x_{2,1} \vee \cdots \vee \neg x_{n/2+1,1})$. In order to do that, we recursively refute the similar $\mathsf{bPHP}_{n/2}$ subformula on variables $x_{1,2}, x_{1,3}, \ldots, x_{1,\log n}, x_{2,2}, \ldots, x_{n/2+1,\log n}$ (cyan part in Table 2). This operation should be done twice, once per each desired clause. Then, we construct a contradiction in the first column (green part in Table 2) using these two clauses. Consider all partial assignments on the first column. We call a partial assignment *sorted* if all variables assigned to zero precede all variables assigned to one. Similarly, a clause is *sorted* if and only if it encodes a negation of a sorted assignment.

It is easy to see that all partial assignments of the size $n+1$ contradict one of two derived clauses due to symmetry. By Remark 11, we can therefore use the weakening rule to explicitly infer all sorted clauses of size $n+1$. This results in $O(n)$ clauses, each obtained using $O(n)$ symmetry steps, yielding a total size of $O(n^2)$ for this part of the proof.

Next, each sorted clause of size $k \leq n$ can be derived by a single resolution step applied to two (not necessarily sorted) clauses of size $k+1$, and every unsorted assignment is symmetric to a sorted one. More formally, each sorted clause is derived via the following resolution step (we omit column indices for clarity, since all variables lie in the same first column):

$$\frac{\neg x_1 \vee \cdots \vee \neg x_k \vee x_{k+1} \vee \cdots \vee x_{t+1} \qquad \neg x_1 \vee \cdots \vee \neg x_k \vee x_{k+1} \vee \cdots \vee x_t \vee \neg x_{t+1}}{\neg x_1 \vee \cdots \vee \neg x_k \vee x_{k+1} \vee \cdots \vee x_t}.$$

Each unsorted clause in this part of the proof is obtained by applying a symmetry step to the proof of $\neg x_1 \vee \neg x_2 \vee \cdots \vee \neg x_k \vee x_{k+1} \vee \cdots \vee x_t \vee x_{t+1}$. Namely, we apply the symmetry $\varphi_{k+1,t+1}$ that swaps rows $k+1$ and $t+1$ in the matrix $(x_{i,j})$:

$$\frac{\mathsf{bPHP}_n}{\neg x_1 \vee \neg x_2 \vee \cdots \vee \neg x_{k+1} \vee x_{k+2} \vee \cdots \vee x_t \vee \neg x_{t+1}}\ \varphi_{k+1,t+1}.$$

Proceeding in decreasing order of size, we refute all sorted assignments, eventually deriving the empty sorted assignment, which yields a contradiction.

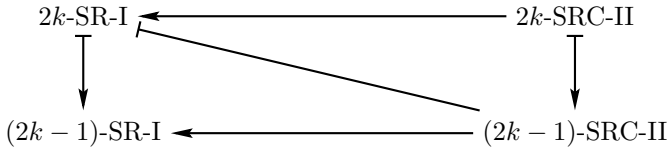

Figure 1 Relations between proof systems for the case when k is at least logarithmic in number of variables, obtained from Corollary 20. Arrows with heads indicate p-simulation and arrows with tails indicate exponential separation.

There are $O(n^2)$ sorted assignments. Therefore, the overall size of the proof satisfies $S_{2\log n\text{-SR-I}}(\mathsf{bPHP}_n) = 2S_{2\log n\text{-SR-I}}(\mathsf{bPHP}_{n/2}) + O(n^2)$, and hence $S_{2\log n\text{-SR-I}}(\mathsf{bPHP}_n) = O(n^2)$. ◀

Our next step is, having an upper bound for bPHP, construct a formula that separates proof systems with different sizes of allowed symmetries.

▶ **Definition 16.** *Let n be a power of 2. $\mathsf{bPHP}_n^{(t)}$ is the formula obtained by adding extra variables $z_{i,j}$ and $w_{i,j}$ to all clauses of bPHP_n in the following way:*

$$(\overline{x_{i,1}\dots x_{i,\log n}} \neq y) \vee (\overline{x_{j,1}\dots x_{j,\log n}} \neq y) \vee \left(\bigvee_{k<t} z_{i,k} \vee z_{j,k}\right) \vee \left(\bigvee_{k<2(t+\log n)} w_{y,k}\right),$$

and also by adding extra clauses that are unit clauses falsifying all the extra variables. We call the non-unit clauses main clauses.

▶ **Theorem 17.** *Let $K = 2(t+\log n)$. Then, $S_{K\text{-SR-I}}(\mathsf{bPHP}_n^{(t)}) = \mathcal{O}(n^3(t+\log n))$.*

Proof. A symmetry swapping two rows a and b is a K-symmetry, since it swaps $x_{a,k}$ with $x_{b,k}$ and $z_{a,k}$ with $z_{b,k}$ and no other variables. Thus, after propagating all unit-clauses (it takes $\mathcal{O}(n^3(t+\log n))$ steps), we can use the proof constructed in Theorem 15. ◀

For proving the lower bounds for systems with small symmetries, we need the following technical statement, which we prove in the full version of the paper.

▶ **Theorem 18.** *Let Π be one of proof systems k-SR-I, k-SR-II, k-SRC-I, or k-SRC-II for some parameter k (i.e. a proof system with static symmetries). Consider a formula Φ over two sets of variables X and Y. Suppose that every symmetry of Φ available in Π maps each clause containing variables from X to itself. Then, for every substitution ρ to Y, $S_\Pi(\Phi) \geq S_{Res}(\Phi[\rho])$.*

▶ **Theorem 19.** *Let $K = 2(t+\log n)$. Then, $S_{(K-1)\text{-SRC-II}}(\mathsf{bPHP}_n^{(t)}) \geq S_{Res}(\mathsf{bPHP}_n)$.*

Proof. Variables $w_{i,j}$ ensure that there are no symmetries that map main clauses corresponding to different values of y to each other. Therefore, the only symmetries of the main clauses are permutations of rows. The smallest nontrivial permutation of rows, which is a swap of two rows, is a K-symmetry. Hence, there are no nontrivial symmetries of size $K-1$ on main clauses. Applying Theorem 18 concludes the proof. ◀

Combining Theorems 17 and 19 together, we obtain a separation for systems with at least logarithmic symmetries.

▶ **Corollary 20.** *Let $k = o(N)$ and $k \geq \log N$, where N is the number of variables. Then, $2k$-SR-I is exponentially separated from $(2k-1)$-SRC-II.*

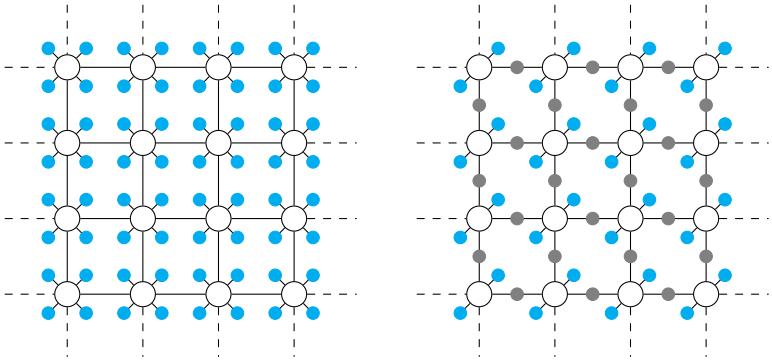

Figure 2 Parts of the graphs of $\eta_n^{(4)}$ (on the left) and $\xi_n^{(2,2,2)}$ (on the right). Gray and cyan vertices are new. Cyan vertices participate in the constraints only with the positive sign.

The requirement $k = o(N)$ here emerges from the fact that we need the parameter n of our formula to grow, and $N = \Theta(nk)$.

4 Tseitin Formulas

In this section, we consider formulas based on Tseitin contradictions on two types of grids: rectangular and triangular. For the sake of clarity, we demonstrate our construction on the rectangular grid first, obtaining simpler but more restricted separations (see Figure 5) and then generalize them to the triangular grid.

We start with the definition of a Tseitin contradiction. We follow the notation in [10].

▶ **Definition 21.** *Let $G = \langle V, E\rangle$ be a graph and α be a mapping from variables to $\mathbb{F}_2$ such that $\sum_{v\in V} \alpha(v) = 1$. A formula on the set of Boolean variables $\{x_e \mid e \in E\}$ is called a* Tseitin formula *on graph G if, for all v, it encodes the following linear equality over $\mathbb{F}_2$:*

$$\sum_{e \ni v} x_e = \alpha(v) \pmod 2.$$

The encoding of each linear equality is done by adding $\deg(v)$-clauses excluding all $2^{\deg(v)-1}$ assignments on $\{x_e \mid v \in e\}$ that contradict the constraint.

A Tseitin formula is a Tseitin contradiction *if $|V|$ is odd and $\alpha(v) = 1$ for all v.*

4.1 Rectangular Grid

Let T_n be the Tseitin contradiction on a toroidal rectangular grid of the size $n \times n$. This formula has known lower bounds for the bounded-depth Frege proof system [10] (see also [11]).

▶ **Theorem 22** ([10]). *For any sufficiently large odd n and $d \le \frac{\log n}{59 \log\log n}$, the following holds. Any depth-d Frege refutation of T_n requires size $\exp(\Omega(n^{\frac{1}{58(d+1)}}))$.*

▶ **Theorem 23.** $S_{3\text{-}SRC\text{-}II}(T_n) = \mathcal{O}(n^2)$

Proof. Informally, we refute T_n as follows. As a first step, we reduce our problem on the toroidal grid to the problem on a planar grid by resolving everything in one row and one column. Then, we solve the problem inductively chipping off one node at the time, row by

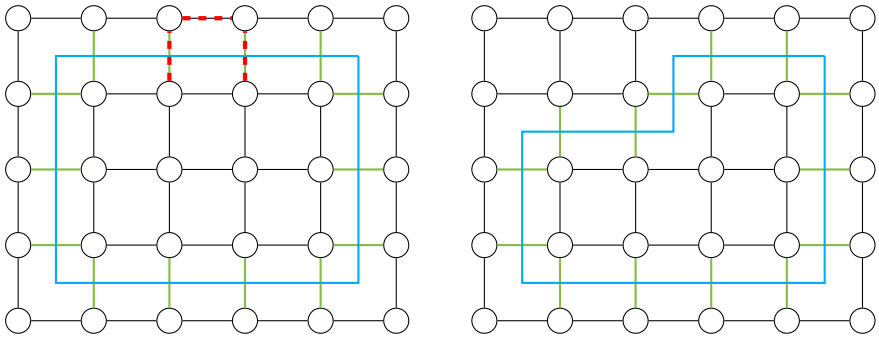

Figure 3 Border assignment (on the left) and broken border assignment (on the right). Edges corresponding to variables in the clause are painted green and the borders are painted cyan.

row, left to right. The order of nodes is important as it is crucial that the border of the remaining part always consists of constant number of "straight edges" (see Figure 3), inside which we can exploit symmetries.

More formally, we start by resolving clauses corresponding to the first row and the first column on the grid. The result of such inference will be a clause consisting of variables corresponding to two sets of vertical edges that we call the top and bottom borders and two sets of horizontal edges that we call left and right borders. We call such clauses *border assignments* (see example in Figure 3). Clauses corresponding to vertices enclosed by borders are not used in the inference. Thus, we call those vertices *unused*. There are exponential number of border assignments. However, we are interested only in those that set all edges to zero, except for one edge for each border (the leftmost edges for top and bottom borders, and the topmost edges for left and right borders). We call such border assignments *basic clauses*. There are only eight basic clauses that could be inferred from the clauses of the first row and the first column, so we infer all of them. This can be done in linear number of steps.

Flipping edges along a U-shaped path (i.e., a path that forms a rectangle without one edge) that starts and ends on a border is a local 3-symmetry of a border assignment, because each used clause has even number of flipped edges. An example of such U-shaped path is shown in Figure 3 by dashed red paint. Due to the structure of border assignments, eight basic clauses are sufficient to produce any other border assignment with the same parity using only local 3-symmetries.

Our next goal is to use the topmost unused row to construct all basic clauses for a border enclosing the same vertices without this row. In order to achieve it, we consider broken border assignments, i.e. assignments to a set of variables that consist of three regular borders and the top border consisting of vertical edges of two neighboring rows and one horizontal edge as shown by colored edges in Figure 4. Similarly to regular border assignments, we define basic clauses for broken border assignments by setting all edges to zero, except for one edge per regular border and three edges closest to the horizontal edge on a broken border (colored red in Figure 4). Again, there are only constant number of basic clauses for each broken border. And similarly, we can use local 3-symmetries to infer any broken-border assignment from basic clauses.

Now, it only remains to observe that all basic clauses of a broken border enclosing t vertices could be obtained from the basic clauses of a broken border enclosing $t+1$ vertices in a constant number of steps. Repeating this process several times, we obtain all basic clauses for the border enclosing a rectangle of the size $1 \times (n-1)$. Then, we can use the similar process to obtain all basic clauses for a single vertex. The basic clauses for a single vertex encode the fact that the parity of edges incident to the vertex is even, while clauses of that vertex encode the fact that it is odd. Resolving all these clauses we obtain a contradiction. ◀

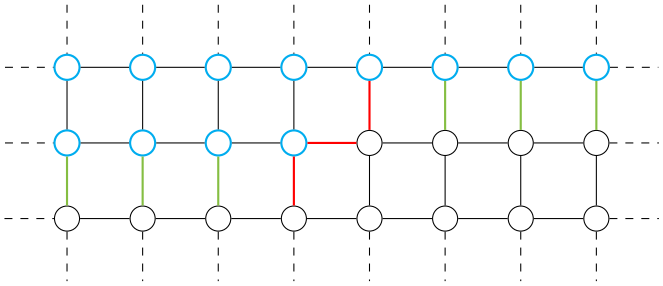

Figure 4 Representation of a basic clause for a broken border assignment.

▶ **Corollary 24.** $S_{4\text{-}SRC\text{-}I}(T_n) = \mathcal{O}(n^2)$

Proof. The proof above used only symmetries that flipped variables along the U-shaped paths in the graph. Each such path can be extended to a cycle of length 4 making a local symmetry global. ◀

Now, having upper bounds for the size of Tseitin formulas on toroidal grid in Resolution with small symmetries, we are ready to construct formulas for the separations. We start with the separation of larger global and local symmetries (with different parameters) from smaller local symmetries. Let $\eta_n^{(k)}$ be a formula T_n with k additional unique variables added to each clause and unit clauses falsifying all additional variables.

▶ **Theorem 25.**

$$S_{(k+1)\text{-}SRC\text{-}II}(\eta_n^{(k)}) \geq S_{Res}(T_n) \tag{1}$$

$$S_{(32k+4)\text{-}SRC\text{-}I}(\eta_n^{(k)}) = \mathcal{O}(n^2 k) \tag{2}$$

$$S_{(16k+3)\text{-}SRC\text{-}II}(\eta_n^{(k)}) = \mathcal{O}(n^2 k) \tag{3}$$

Proof. Since T_n has no 1-symmetries, all $(k+1)$-symmetries of $\eta_n^{(k)}$ satisfy the conditions of Theorem 18. This gives us the lower bound.

On the other hand, we can still repeat the proof from Theorem 23 after using Resolution steps to eliminate all additional variables. The only obstacle is to make sure that our system is able to perform the symmetry step that flips two variables along a U-shaped path, only two vertices of which (the ones that are in the base of the letter U) were used in the inference. In the case of local symmetries, it is sufficient to add $16k$ additional nontrivially mapped variables to the symmetry, while in the case of global symmetries we need to permute clauses of all four vertices which results in a $(32k+4)$-symmetry. ◀

Now we want to construct formula $\xi_n^{(a,b,k)}$ that separate only systems global symmetries, but with much better parameters than $\eta_n^{(k)}$. Consider formula T_n. In order to separate k-SRC-I from $(k+1)$-SRC-I, we make the following modifications to this formula.

- Split each horizontal edge into a smaller edges and each vertical edge into b smaller edges. We refer to all newly added vertices as *small vertices* and to all original vertices as *big vertices*. We set the charge of all small vertices to zero to enforce the constraint that each edge is split into smaller edges of the same value.
- Add k new variables to each clause corresponding to each big vertex and add unit-clauses falsifying all such variables. We refer to these variables as *auxiliary variables*.

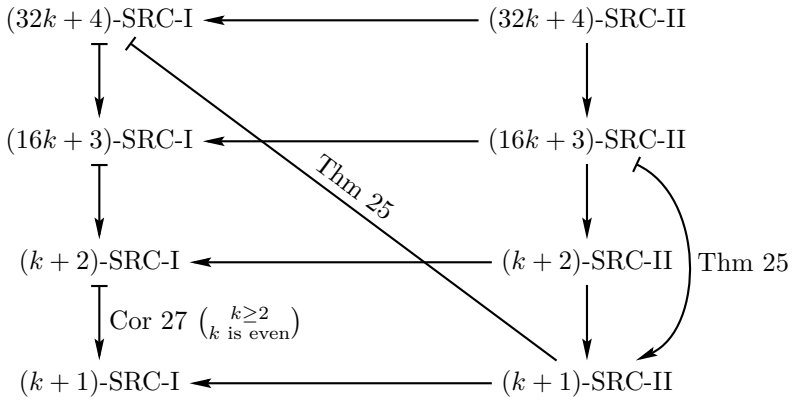

Figure 5 Relations between proof systems that follow from the hierarchy theorems for the rectangular grid.

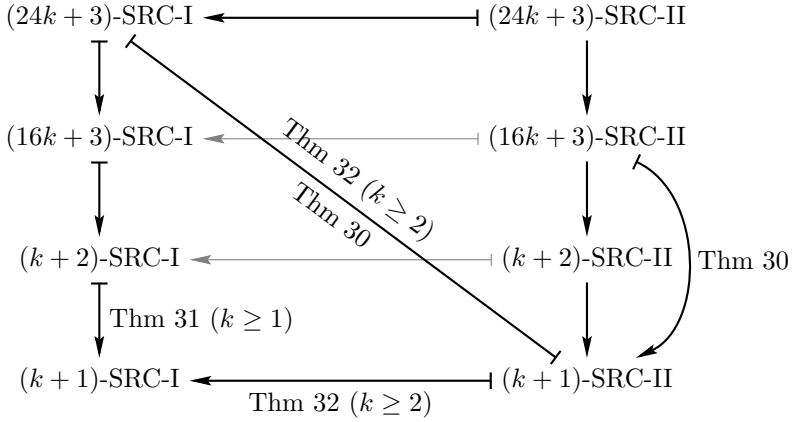

Figure 6 Relations between proof systems that follow from the hierarchy theorems for the triangular grid.

Let $\xi_n^{(a,b,k)}$ be the resulting formula. Note that the original formula is $\xi_n^{(1,1,0)}$. Clearly, $\xi_n^{(a,b,k)}$ consists of $\mathcal{O}(n^2(k+a+b))$ variables and clauses.

▶ **Theorem 26.**

$$S_{(2a+2b-1)\text{-SRC-I}}(\xi_n^{(a,b,a+b)}) \geq S_{Res}(T_n)$$
$$S_{2(a+b)\text{-SRC-I}}(\xi_n^{(a,b,a+b)}) = \mathcal{O}(n(a+b)+n^2)$$

Proof. See the full version of the paper. ◀

▶ **Corollary 27.** *For every even $k \geq 4$, there exists a family of formulas Φ_n such that the size of the shortest k-SRC-I-proof of Φ_n is polynomial, but the size of the shortest $(k-1)$-SRC-I-proof is exponential.*

4.2 Triangular Grid

Let T'_n be a Tseitin contradiction on a graph constructed as a rectangular toroidal grid with diagonals of the type $(i,j)-(i+1,j+1)$ added to each rectangle (see Figure 7).

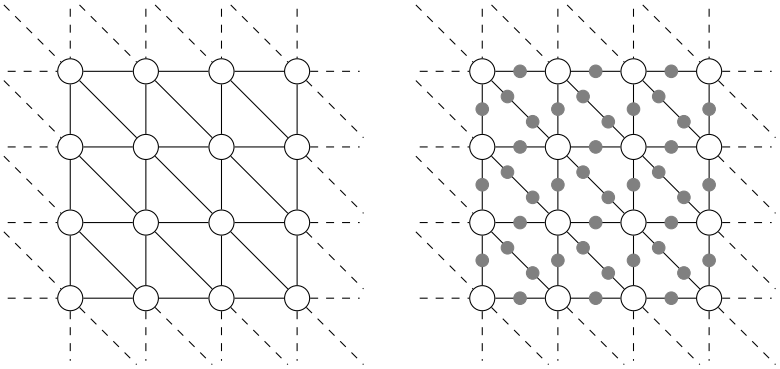

Figure 7 Parts of the graphs of T'_n (on the left) and $\xi_n^{(2,2,3,0)}$ (on the right).

▶ **Lemma 28.** $S_{\textit{depth-d Frege}}(T'_n) \geq S_{\textit{depth-d Frege}}(T_n)$.

Proof. Setting all diagonals to zero maps T'_n to T_n and any proof of T'_n to a proof of T_n. ◀

▶ Remark 29. The proof of T'_n in Resolution with global symmetries analogous to the proof in Corollary 24 uses 3-symmetries. Thus, $S_{\text{3-SRC-I}}(T'_n) = \mathcal{O}(n^2)$.

We want to construct a formula similar to $\eta_n^{(k)}$, but with the base formula T'_n. A naive approach – adding k new variables to each clause together with unit clauses falsifying them – leads to worse constants because each vertex constraint in T'_n consists of 32 clauses, compared to 8 in T_n. Hence, we need to change the way we add the variables. For each vertex, we arbitrarily choose 3 out of 4 vertical or horizontal edges and split all 32 clauses into 8 groups of 4 clauses with respect to the signs of those three edges. In each group, we add k new variables to each clause and k unit clauses falsifying them. Added variables are unique for each group, but they are the same for the clauses inside the group. As a result, each vertex constraint contains only $8k$ additional variables. We call the constructed formula $\eta_n'^{(k)}$.

▶ **Theorem 30.**

$$S_{(k+1)\textit{-SRC-II}}(\eta_n'^{(k)}) \geq S_{\textit{Res}}(T_n) \tag{4}$$

$$S_{(24k+3)\textit{-SRC-I}}(\eta_n'^{(k)}) = \mathcal{O}(n^2 k) \tag{5}$$

$$S_{(16k+3)\textit{-SRC-II}}(\eta_n'^{(k)}) = \mathcal{O}(n^2 k) \tag{6}$$

Proof. Similar to the proof of Theorem 25. Only the lower bound requires some additional work. See the full version of the paper for more details. ◀

Let $\xi_n^{(a,b,c,k)}$ be the formula constructed similarly to $\xi_n^{(a,b,k)}$, but with formula T'_n used as a base formula, where c is the number of parts each diagonal edge is split into and all other parameters having the same meaning as before.

▶ **Theorem 31.** *For every $k \geq 3$, there exists a family of formulas Φ_n such that the size of the shortest k-SRC-I-proof of Φ_n is polynomial, but the size of the shortest $(k-1)$-SRC-I-proof is exponential.*

Proof. Similar to Corollary 27. See the full version of the paper for more details. ◀

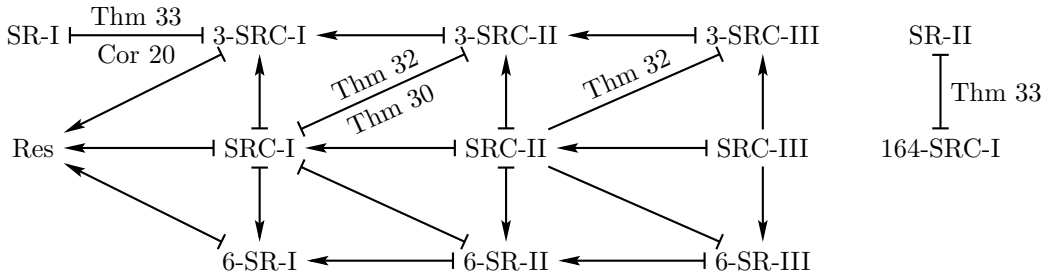

Figure 8 Relation between proof systems with the smallest symmetry constraints.

5 Separating Small Symmetries from Large Symmetries

There are three possible metrics on which one system with small symmetries can have an advantage over another. Each of these metrics leads to a different type of separation:

1. Increasing the size of the symmetries. An example of such statement is: for appropriate values of k, $k+1$-SR-I has polynomial proofs for formulas hard for k-SRC-II.
2. Generalizing the type of the symmetries. For example, 6-SR-II has polynomial proofs for formulas hard for SRC-I (Theorem 32).
3. Adding complementary symmetries. For example, 164-SRC-I has polynomial proofs for formulas hard for SR-II (Theorem 33).

In previous sections, we focused on the separations of the first type. In this section, we consider the other two types, where the system with small symmetries has an advantage over the system with large symmetries.

We start with the separations of small symmetries of more general type from large symmetries of less general type. To do it, we construct four new families of formulas on top of T'_n from the previous section. Full constructions appear in the full version of the paper.

▶ **Theorem 32.** *There exist families of formulas Φ_n, Ψ_n, Φ'_n, Ψ'_n such that:*
- *$S_{3\text{-}SRC\text{-}III}(\Phi_n)$ is polynomial, but $S_{SRC\text{-}II}(\Phi_n)$ is exponential.*
- *$S_{3\text{-}SRC\text{-}II}(\Psi_n)$ is polynomial, but $S_{SRC\text{-}I}(\Psi_n)$ is exponential.*
- *$S_{6\text{-}SR\text{-}III}(\Phi'_n)$ is polynomial, but $S_{SRC\text{-}II}(\Phi'_n)$ is exponential.*
- *$S_{6\text{-}SR\text{-}II}(\Psi'_n)$ is polynomial, but $S_{SRC\text{-}I}(\Psi'_n)$ is exponential.*

Combining this theorem with Theorem 30 and Corollary 20, we obtain all separations shown in Figure 8, except for the half of the top left edge and the separation between SR-II and 164-SRC-I. The remaining separations (proved in the full version of the paper) are obtained by generalizing the SRC-II lower bound construction of Arai and Urquhart [1].

▶ **Theorem 33.** *There exist families of formulas τ_n and τ'_n such that:*
- *$S_{3\text{-}SRC\text{-}I}(\tau_n)$ is polynomial, but $S_{SR\text{-}I}(\tau_n)$ is exponential.*
- *$S_{164\text{-}SRC\text{-}I}(\tau'_n)$ is polynomial, but $S_{SR\text{-}II}(\tau'_n)$ is exponential.*

▶ Remark 34. The number 164 in the previous theorem is not optimal, and it can be reduced to at most 54 by a more careful construction. However, doing this would require a lot of technical work, and it would not add much to the understanding of the problem, so we do not pursue this direction.

As an immediate corollary, we obtain the following separation that was not known before.

▶ **Corollary 35.** *There exists a family of formulas τ'_n such that the size of the shortest SRC-I-proof of τ'_n is polynomial, but the size of the shortest SR-II-proof is exponential.*

6 Conclusion and Open Problems

We have shown that the systems 3-SRC-I and 6-SR-I are stronger than Resolution. The question of whether systems with smaller constraints are stronger than Resolution is open. It seems plausible that a separation exists even for 1-SRC-II and Resolution. Separating 2-SRC-III from Resolution would also be of independent interest: while 2-symmetries may efficiently encode parities of some variables, an efficient decoding of those parities from a polynomial-size formula often requires larger symmetries.

For proof systems with global symmetries and for systems with at least logarithmic symmetries, we have shown that the addition of a constant to the size of allowed symmetries makes the system stronger. However, for systems with sublogarithmic local symmetries, we only have a separation result with respect to a multiplicative factor, but not an additive constant. It seems natural for this case to also have the separation for the additive constant, so it would be interesting to find it.

Although in this paper we focused on static symmetries, it would be interesting to study the case of small dynamic symmetries as well. Even the simplest case of dynamic symmetries, 1-SRC-III, seems to be very nontrivial. 1-symmetries grant the system the power to express, although with some restrictions, conjunctions of sets of variables. For instance, $(x \wedge y)$ could be expressed as a *single* clause $(x \vee y)$ with two symmetries negating x and y. It raises the potential power of the system to Res(k) (a version of the Resolution proof system that works with k-DNFs instead of clauses). However, due to the structure of the proof with symmetries, it is not possible to operate on the conjunctions freely, so both upper and lower bounds for 1-SRC-III are interesting open problems.

Additionally, the notion of small symmetries could also be naturally applied to systems that are stronger than Resolution. Examples of such systems are: Cutting Planes, Res(k), or even constant-depth Frege. It looks hard to prove lower bounds for these systems augmented with large symmetries, but proving lower bounds for systems with small symmetries only seems more feasible.

References

1 Noriko H. Arai and Alasdair Urquhart. Local symmetries in propositional logic. In Roy Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 40–51, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. doi:10.1007/10722086_3.

2 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *J. ACM*, 48(2):149–169, March 2001. doi:10.1145/375827.375835.

3 Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified symmetry and dominance breaking for combinatorial optimisation. *Journal of Artificial Intelligence Research*, 77:1539–1589, March 2022. doi:10.1613/jair.1.14296.

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

5 Jo Devriendt, Bart Bogaerts, and Maurice Bruynooghe. Symmetric explanation learning: Effective dynamic symmetry handling for SAT. In Serge Gaspers and Toby Walsh, editors, *Theory and Applications of Satisfiability Testing – SAT 2017*, pages 83–100, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-66263-3_6.

6 Nikita Gaevoy. The power of small symmetries, 2026. arXiv:2606.25895.

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

8 Marijn Heule. Schur number five. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. doi:10.1609/aaai.v32i1.12209.

9 Marijn J. H. Heule, Oliver Kullmann, and Victor W. Marek. Solving and verifying the boolean pythagorean triples problem via cube-and-conquer. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing – SAT 2016*, pages 228–245, Cham, 2016. Springer International Publishing. doi:10.1007/978-3-319-40970-2_15.

10 Johan Håstad. On small-depth Frege proofs for Tseitin for grids. *J. ACM*, 68(1), November 2020. doi:10.1145/3425606.

11 Johan Håstad and Kilian Risse. On bounded depth proofs for Tseitin formulas on the grid; revisited. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1138–1149, 2022. doi:10.1109/FOCS54457.2022.00110.

12 Alexander Ivrii and Ofer Strichman. Exploiting isomorphic subgraphs in SAT. *Proceedings of the 21st Formal Methods in Computer-Aided Design, FMCAD 2021*, pages 204–211, 2021. doi:10.34727/2021/isbn.978-3-85448-046-4_29.

13 Boris Konev and Alexei Lisitsa. Computer-aided proof of Erdős discrepancy properties. *Artificial Intelligence*, 224:103–118, 2015. doi:10.1016/j.artint.2015.03.004.

14 Michal Kouril and Jerome L. Paul. The van der Waerden number W(2, 6) is 1132. *Experimental Mathematics*, 17(1):53–61, 2008. doi:10.1080/10586458.2008.10129025.

15 Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Structures & Algorithms*, 7(1):15–39, 1995. doi:10.1002/rsa.3240070103.

16 Balakrishnan Krishnamurthy. Short proofs for tricky formulas. *Acta Inf.*, 22(3):253–275, August 1985. doi:10.1007/BF00265682.

17 Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3(2):97–140, 1993. doi:10.1007/BF01200117.

18 Artur Riazanov. On the decision trees with symmetries. In Fedor V. Fomin and Vladimir V. Podolskii, editors, *Computer Science – Theory and Applications*, pages 282–294, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-90530-3_24.

19 Pascal Schweitzer and Constantin Seebach. Resolution with symmetry rule applied to linear equations. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, volume 187 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2021.58.

20 Stefan Szeider. The complexity of resolution with generalized symmetry rules. *Theory of Computing Systems*, 38:171–188, February 2005. doi:10.1007/s00224-004-1192-0.

21 Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, January 1987. doi:10.1145/7531.8928.

22 Alasdair Urquhart. The symmetry rule in propositional logic. *Discrete Applied Mathematics*, 96-97:177–193, 1999. doi:10.1016/S0166-218X(99)00039-6.