

Resolution with Symmetry Rule Applied to Linear Equations

Pascal Schweitzer

TU Kaiserslautern, Germany

Constantin Seebach

TU Kaiserslautern, Germany

Abstract

This paper considers the length of resolution proofs when using Krishnamurthy’s classic symmetry rules. We show that inconsistent linear equation systems of bounded width over a fixed finite field \mathbb{F}_p with p a prime have, in their standard encoding as CNFs, polynomial length resolutions when using the local symmetry rule (SRC-II).

As a consequence it follows that the multipede instances for the graph isomorphism problem encoded as CNF formula have polynomial length resolution proofs. This contrasts exponential lower bounds for individualization-refinement algorithms on these graphs.

For the Cai-Fürer-Immerman graphs, for which Torán showed exponential lower bounds for resolution proofs (SAT 2013), we also show that already the global symmetry rule (SRC-I) suffices to allow for polynomial length proofs.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases Logical Resolution, Symmetry Rule, Linear Equation System

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.58

Related Version *Full Version:* <https://arxiv.org/abs/2101.05142> [15]

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (EngageS: grant agreement No. 820148).

1 Introduction

Refutation via logical resolution is one of the most basic and fundamental methods in theorem proving used to argue the validity of statements in propositional logic. It is famously sound and complete for proving that formulas in conjunctive normal form (CNF) are unsatisfiable. In automated theorem proving, resolution is in particular used for various primitive backtracking algorithms for the satisfiability problem (SAT) such as the DPLL algorithm.

However, resolution is primitive in that we know simple unsatisfiable CNF formulas that admit only resolution refutations of superpolynomial length. This was first proven by Haken [11] who showed that a canonical encoding of the pigeonhole principle into a CNF formula provides formulas whose shortest refutations are superpolynomial in length. Other examples and exponential bounds were given by Chvátal and Szemerédi [5] as well as Urquhart who used formulas based on Tseitin tautologies [19]. Investigating the resolution complexity of the graph non-isomorphism problem, Torán [17] constructed CNF formulas from so-called CFI-graphs (see [4]) and showed the shortest resolution proofs of the arising formulas have exponential length.

As observed by Krishnamurthy, many simple examples without short resolution refutations exhibit symmetries. This prompted the introduction of Krishnamurthy’s symmetry rule [12] which intuitively allows the deduction of a clause symmetric to a previously deduced



© Pascal Schweitzer and Constantin Seebach;

licensed under Creative Commons License CC-BY 4.0

38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021).

Editors: Markus Bläser and Benjamin Monmege; Article No. 58; pp. 58:1–58:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



clause in one step (formal definitions are given in Section 2). For various formulas, Krishnamurthy argued polynomial bounds when the symmetry-rule is used, leading to exponential improvements. Further examples with this effect, including another analysis for pigeonhole principle formulas, were provided by Urquhart [20].

Krishnamurthy in fact introduced two rules, each of them arises from permutations of the variables. The *global* rule allows only symmetries of the entire original formula, while the *local* one allows us to use symmetries of a subset of the clauses. These rules led to the proof systems SR-I (*symmetric resolution*) and SR-II (*locally symmetric resolution*), respectively. Urquhart [20] introduced *complementation symmetries* in addition to the variable permutations. This allows us to interchange literals with their negations and leads to the proof systems SRC-I and SRC-II. In [20] Urquhart also showed that there are exponential-to-polynomial improvements regarding proof length from the system SR-I to SRC-I. Arai and Urquhart [1] showed exponential-to-polynomial improvements from SR-I to SR-II and also provided exponential lower bounds for SRC-II.

Szeider [16], who actually focuses on homomorphisms, describes another strengthening of the symmetry rule. In his extension we are allowed the use of symmetries within clauses that have been resolved, rather than only allowing clauses of the original formula. This is called resolution with *dynamic* symmetries and leads to the proof systems SR-III and SRC-III, depending on whether complementation is allowed. However, to date it remains an open problem to find superpolynomial lower bounds on proof length in SR-III and SRC-III.

1.1 Contribution

In this paper we are concerned with proof systems obtained by extending resolution with additional symmetry rules. We prove that the CNF formulas arising from the CFI-graphs have refutations polynomially bounded in length in the SR-I calculus. With Torán's exponential lower bounds [17] mentioned above, this gives an exponential-to-polynomial improvement for the resolution complexity of non-isomorphism when introducing the symmetry rule. To those familiar with the details of the CFI-construction this may not come as a surprise, since the CFI-graphs exhibit many global symmetries. However, this is not the case for multipede graphs, these arise from a construction related to the CFI-graphs [10]. Crucially these graphs are asymmetric. That is, they have no symmetries at all. They provide exponential lower bounds for all individualization-refinement algorithms for the graph isomorphism problem. This includes all tools currently viable in practice, such as nauty/traces [13]. The initial intuition might therefore be that the CNF formulas arising from multipedes provide exponential lower bounds for SRC-III. However, this turns out not to be the case. In fact, maybe surprisingly, we show that even when using only local symmetries rather than dynamic symmetries (i.e., in SCR-II rather than SCR-III) there are polynomial bounds on the respective formulas. In some sense this shows that the multipedes have substructures with symmetries that allow them to be distinguished concisely.

To prove this statement, we reduce the statement to one concerning linear equation systems. It is known that isomorphism of CFI and multipede graphs are related to solvability of linear equation systems. (This is also the case for Tseitin tautologies.) We show that this relation can be exploited. Specifically, we show that there is a resolution transforming the CNFs arising from the graph isomorphism instances to CNFs arising from linear equation systems. We then show our main theorem which says that inconsistent linear equation systems with equations of bounded width (i.e., the maximum number of non-zero coefficients in an equation is bounded) have polynomial resolutions using the local symmetry rule.

	none	global	local	dynamic
without complementation	classical resol.[5, 19]	SR-I [20]	SR-II [1]	SR-III (open)
with complementation	–	SRC-I [20]	SRC-II [1]	SRC-III (open)

■ **Figure 1** Resolution calculi with symmetries rules of varying degree of generality and references with formulas proving exponential lower bounds on resolution length.

► **Theorem 1.** *Inconsistent linear equation systems of bounded width over a fixed finite field \mathbb{F}_p with p a prime have, in their standard encoding as CNFs, polynomial length resolutions when using the local symmetry rule (i.e., in SRC-II).*

Structure of the paper. Section 3 shows that the CNF formulas arising from CFI-graph pairs have polynomial length proofs in SR-I. Due to space restrictions, the proof of this result (Theorem 18) was omitted from this version of the paper. It can be found in the full version [15]. Section 4 shows that linear equation systems of bounded width have polynomial length proofs in SRC-II. Section 5 shows that the formulas arising from (bounded degree) multipede graphs can be transformed in the resolution calculus (without using symmetry) to linear equation systems of bounded width.

1.2 Related Work

Figure 1 gives an overview of resolution calculi with symmetry and references to lower bound constructions. A proof system p -simulates another proof system if shortest proofs in the latter are polynomially bounded in the length of shortest proofs in the former. We should remark that the extended resolution system introduced by Tseitin [18] can p -simulate proof systems with symmetries [20]. See [2] for an implementation using Krishnamurthy’s symmetry rule. Symmetry rules have of course also been introduced for other proof systems [3, 8]. See also [7] for another way to incorporate symmetries into resolution.

Connection to the graph isomorphism problem. The results of our paper are connected to the graph isomorphism problem in two conceptually very different ways. First, finding valid literal permutations (with or without complementation) for the global symmetry rule is equivalent to the graph isomorphism problem itself (e.g., [4]). Therefore isomorphism solvers such as nauty/traces [13], which are highly efficient in practice, can be used to find the symmetries (see [6]). Symmetry detection is one of the standard applications of graph isomorphism solvers, for example there is a tool integrating nauty into Prolog [9] for this purpose.

Second, our results relate to the proof complexity of the graph isomorphism problem itself, which explains why we are interested in CNF formulas arising from non-isomorphism instances. Torán [17] describes a canonical way to encode the isomorphism problem as a CNF formula (see Subsection 2.2). The resolution complexity of graph non-isomorphism is related to the complexity of the graph isomorphism problem. After all isomorphism solvers need to prove, some way or another, that the inputs are non-isomorphic, if they are. A crucial feature of isomorphism solvers is that they are able to exploit already detected symmetries (i.e., automorphisms) of the underlying instances during run-time [13]. Vaguely, this translates into a symmetry rule that they apply already during the process of computing the symmetries of the instance. Current tools basically only exploit local symmetries. Our new insights into

the resolution complexity of multipedes thus shows a combinatorial possibility to solve their isomorphism problem. It brings up the question how to exploit local symmetries in graph isomorphism solvers.

It remains unknown whether graph non-isomorphism has polynomial resolution complexity in any of the proof systems with symmetry rule we have discussed.

2 Preliminaries

2.1 Resolution and the Symmetry Rule

We are interested in unsatisfiability proofs of Boolean formulas. The basic resolution proof system works with formulas in conjunctive normal form.

Let Γ be a finite set of variables. $\text{Lit}(\Gamma) := \Gamma \cup \bar{\Gamma}$ is the set of literals, where $\bar{\Gamma} := \{\bar{x} \mid x \in \Gamma\}$. A *clause* is a disjunction of literals. We also represent clauses as sets of literals. A Boolean formula is in *conjunctive normal form (CNF)* if it is a conjunction of clauses. We may treat such a formula as a set of clauses. \perp is the empty clause, i.e. the disjunction of the empty set, which is unsatisfiable. For sets of clauses C_1 and C_2 define $C_1 \sqsubseteq C_2 : \iff \forall c_1 \in C_1 \exists c_2 \in C_2 : c_1 \supseteq c_2$. Since we will treat clauses as sets of literals, we do not care for their order, i.e. we do not differentiate between $x \vee y$ and $y \vee x$. The same applies to CNF formulas, which we interpret as sets of clauses.

► **Definition 2.** Resolution is a proof system in propositional logic. It operates on CNF formulas, employing a single inference rule:

$$\frac{x \vee A, \bar{x} \vee B}{A \vee B}.$$

The clause produced by the resolution rule is called *resolvent*.

Let $A = \{a_1, \dots, a_m\}$ and B be sets of clauses. We write $A \vdash_n B$ if there exists a sequence of clauses $a_1, \dots, a_m, c_1, \dots, c_n$ such that every c_i is a resolvent of two earlier clauses and $B \subseteq A \cup \{c_1, \dots, c_n\}$. Such a sequence is called *derivation of B from A* . When the length of the sequence is irrelevant, we write $A \vdash B$, meaning $A \vdash_n B$ for some n . Given a clause b , we also write $A \vdash_n b$ for $A \vdash_n \{b\}$.

For a CNF formula F with $F \vdash_n \perp$, we say F has a *resolution refutation of size n* .

We write $A \vdash_n^w B$ if there exists a set of clauses B' such that $B \sqsubseteq B'$ and $A \vdash_n B'$. This is a *weaker requirement than $A \vdash_n B$* .

Resolution is sound and complete, i.e. $F \vdash_n \perp$ if and only if F is unsatisfiable. We examine the proof complexity of formulas in this proof system, i.e., the length of the shortest possible resolution refutation of a given formula, in relation to the formula size. There exist classes of formulas with exponential lower bounds on the resolution proof complexity [5, 17, 19].

In the following we define the symmetry rule, which is an extension to resolution, aiming to reduce the proof complexity of some of these hard formulas.

► **Definition 3.** Let L be a finite set of literals. A bijection $\sigma : L \rightarrow L$ is called *renaming* if for every $\ell \in L$ we have $\overline{\sigma(\ell)} = \sigma(\bar{\ell})$.

A renaming is essentially a permutation of the variables that may also negate some of them. We can apply renamings to clauses (i.e., sets of literals) and CNF formulas (i.e., sets of clauses). In either case we define $\sigma(C) := \{\sigma(x) \mid x \in C\}$.

► **Definition 4** (The Symmetry Rule [1] [16]). Consider a derivation S from a formula F and a subsequence S' of S which derives a clause C from a subset $F' \subseteq F$. If there exists a renaming σ with $\sigma(F') \subseteq F$, then the local symmetry rule allows derivation of $\sigma(C)$.

With the restriction $F = F'$, we obtain the global symmetry rule. Adding the global or local symmetry rule to the resolution system yields the proof systems **SRC-I** and **SRC-II**, respectively.

We write $A \vdash_n^{\text{SRC-II}} B$ to indicate that B can be derived from A using resolution and the local symmetry rule, with a derivation of length at most n .

Note that in order to apply σ via the local symmetry rule to some clause C in a derivation, we must look at the entire history of how C was derived, and find out which part $F' \subseteq F$ of the original formula was used. Then we need to check that $\sigma(F') \subseteq F$.

This means that in general we cannot chain derivations that use the symmetry rule together, because such an operation changes the history for some of the clauses. Still, we can combine **SRC-II** derivations in the following ways:

► **Lemma 5.** Let A, B, C and D be sets of clauses and $n, m \in \mathbb{N}$.

- (a) $A \vdash_n^{\text{SRC-II}} B$ and $A \subseteq C$ implies $C \vdash_n^{\text{SRC-II}} B$
- (b) $A \vdash_n^{\text{SRC-II}} B$ and $B \vdash_m C$ implies $A \vdash_{n+m}^{\text{SRC-II}} C$
- (c) $A \vdash_n^{\text{SRC-II}} B$ and $C \vdash_m^{\text{SRC-II}} D$ implies $A \cup C \vdash_{n+m}^{\text{SRC-II}} B \cup D$

► **Lemma 6.** Let A and B be sets of clauses and d a clause.

- (a) $A \vdash \perp$ and $\{c \vee d \mid c \in A\} \sqsubseteq B$ implies $B \vdash^w d$

2.2 Encoding Graph Isomorphism

Our interest in the graph isomorphism problem is twofold: First, finding valid literal permutations for the symmetry rule is equivalent to finding certain graph isomorphisms. Secondly, we examine the proof complexity of the problem by translating it into propositional logic and applying resolution with symmetry rule.

A *graph* is a tuple (V, E) of a set of vertices V and edges E . Each edge is a two element subset of V . A *colored graph* is a graph (V, E) together with a function $f : V \rightarrow C$, called *coloring*, assigning to every vertex a color from some set C . Let $G = (V, E)$ be a graph and $v \in V$. $\mathcal{E}_G(v) := \{e \in E \mid v \in e\}$ are the edges *incident* with v . $N_G(v) := \{u \in V \mid \{u, v\} \in E\}$ is the *neighborhood* of v . $\deg_G(v) := |N_G(v)| = |\mathcal{E}_G(v)|$ is the *degree* of v .

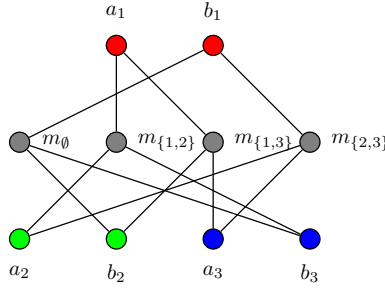
Given a colored graph $G = (V, E)$ with coloring f and a vertex $v \in V$, we can *individualize* v by creating a new coloring f' such that $f'(v) := (f(v), 1)$ and setting $f'(v') := (f(v'), 0)$ for all $v' \in V \setminus \{v\}$. We write the individualized graph as G_v .

► **Definition 7.** Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs.

- A graph isomorphism from G_1 to G_2 is a bijection $\varphi : V_1 \rightarrow V_2$ such that for all $v, v' \in V_1$ we have $\{v, v'\} \in E_1$ if and only if $\{\varphi(v), \varphi(v')\} \in E_2$.
- We say G_1 and G_2 are isomorphic, written $G_1 \cong G_2$, if there exists a graph isomorphism from G_1 to G_2 .
- An automorphism of a graph G is a graph isomorphism from G to itself.
- $\text{Aut}(G)$ is the automorphism group of G .

The automorphisms of a graph constitute its inherent combinatorial symmetries. We will use the terms automorphism and symmetry synonymously.

Given two graphs G_1 and G_2 , one can construct a Boolean formula that is satisfiable if and only if there is an isomorphism between G_1 and G_2 [17]. This is commonly done by constraining variables of the form $x_{u,v}$ such that each satisfying assignment corresponds to an isomorphism: 1 is assigned to $x_{u,v}$ if and only if the isomorphism maps u to v .



■ **Figure 2** The CFI-gadget $X_{\{1,2,3\}}$.

► **Definition 8.** For a pair of graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ with $|V_1| = |V_2|$, define

$F(G_1, G_2) := T_1 \wedge T_2 \wedge T_3$, where

$$T_1 := \bigwedge_{v_1 \in V_1} \bigvee_{v_2 \in V_2} x_{v_1, v_2},$$

$$T_2 := \bigwedge_{v_2 \in V_2} \bigwedge_{\substack{v_1, v'_1 \in V_1 \\ v_1 \neq v'_1}} (\overline{x_{v_1, v_2}} \vee \overline{x_{v'_1, v_2}}),$$

$$T_3 := \bigwedge_{\{u_1, v_1\} \in E_1 \oplus \{u_2, v_2\} \in E_2} (\overline{x_{u_1, u_2}} \vee \overline{x_{v_1, v_2}}).$$

We refer to the clauses of this CNF formula as being “of Type i ”, depending on which T_i they come from. The clause types naturally encode the concept of a graph isomorphism in propositional logic. Specifically, Type 1 and Type 2 clauses ensure that we have a bijection from V_1 to V_2 ; Type 3 clauses make the function preserve edges.

If the graphs G_1 and G_2 are colored by some functions l_1 and l_2 respectively, then an isomorphism between them should respect the colors. To represent this in the formula $F(G_1, G_2)$, we simply assign 0 to all variables $x_{u,v}$ for which $l_1(u) \neq l_2(v)$.

2.3 The CFI Graphs

In this section, we look at the graphs by Cai, Fürer and Immerman [4], which were constructed to prove lower bounds for the Weisfeiler-Lehman method in isomorphism testing. These graphs are also challenging when we use resolution to decide isomorphism. They are built from gadget graphs which are defined as follows (see Figure 2).

► **Definition 9** (CFI-gadget [4, 6]). Given a finite set N , define: $X_N := (V, E, \gamma)$, where $V := A \cup B \cup M$ consists of $A := \{a_w \mid w \in N\}$, $B := \{b_w \mid w \in N\}$ as well as $M := \{m_S \mid S \subseteq N, |S| \text{ even}\}$ and $E := \{\{m_S, a_w\} \mid w \in S\} \cup \{\{m_S, b_w\} \mid w \in N \setminus S\}$. Also define the coloring $\gamma : V \rightarrow C : v \mapsto \gamma(v) := \begin{cases} c_w & \text{if } v \in A \cup B \text{ with } v = a_w \text{ or } v = b_w, \\ m & \text{if } v \in M. \end{cases}$

The most important feature of the CFI-gadgets are their automorphisms:

► **Lemma 10** ([4, 6.1]). There are $2^{|N|-1}$ automorphisms of X_N . Each is uniquely determined by interchanging the vertices a_w and b_w for all w in some subset $S \subseteq N$ of even cardinality.

► **Definition 11** (CFI graph). From a graph $G = (V, E)$ construct $X(G)$ by connecting the CFI gadgets $\{X_{\mathcal{E}_G^v} \mid v \in V\}$ with edges $E' := \{\{a_e^u, a_e^v\} \mid e = \{u, v\} \in E\} \cup \{\{b_e^u, b_e^v\} \mid e = \{u, v\} \in E\}$.

► **Definition 12.** *Given a graph $G = (V, E)$ with $E \neq \emptyset$, construct $\tilde{X}(G)$ from $X(G)$ by choosing some edge $e = \{u, v\} \in E$ and replacing the edges $\{a_e^u, a_e^v\}, \{b_e^u, b_e^v\}$ with the edges $\{a_e^u, b_e^v\}, \{b_e^u, a_e^v\}$. We say that the edges corresponding to e have been twisted.*

Note that Definition 12 does not specify how to choose the edge which is to be twisted, so there are in fact multiple graphs that we could call $\tilde{X}(G)$. If G is connected however, these graphs are isomorphic. On the other hand, for any graph G with at least one edge, $\tilde{X}(G)$ and $X(G)$ are not isomorphic [[4, see Lemma 6.2]]. The CFI graphs have been used to prove the following lower bound for resolution:

► **Theorem 13** ([17, Corollary 5.2]). *There exists a family of graphs $\mathcal{G} = (G_n)_{n \in \mathbb{N}}$ such that for every n , G_n has n vertices and the resolution refutation of the formula $F(X(G_n), \tilde{X}(G_n))$ requires size $\exp(\Omega(n))$. The graphs $X(G_n)$ and $\tilde{X}(G_n)$ have color multiplicity at most 4.*

This exponential lower bound motivates the use of a more efficient proof system to prove non-isomorphism of CFI graphs. Because of the symmetric nature of the CFI-gadgets, the symmetry rule is expected to reduce the proof length significantly. With symmetry rule, short proofs exist, as we show in Section 3.

In order to obtain examples for which the symmetry rule is not able to produce short proofs, it is a natural idea to consider asymmetric graphs instead.

2.4 Multipede Graphs

In [10], the so-called Multipedes were defined - a method to construct asymmetric structures. Combining this construction with CFI-gadgets, one obtains a family of asymmetric graphs which provide exponential lower bounds for individualization-refinement algorithms [14].

► **Definition 14** (Multipede graph [14]). *From a bipartite graph $G = (V, W, E)$, we construct the Multipede graph $MP(G)$ as follows: For every $w \in W$ create a pair of vertices a_w, b_w , colored with c_w . We call these pairs feet. Then for every $v \in V$ take a CFI-gadget $X_{N_G(v)}^v$ and identify the vertices a_w^v and b_w^v with a_w and b_w respectively.*

► **Theorem 15** ([14]). *There exists a family of bipartite graphs $\mathcal{G} = (G_n)_{n \in \mathbb{N}}$ such that for each n the graph G_n has $\mathcal{O}(n)$ vertices, $MP(G_n)$ is asymmetric and individualization-refinement algorithms take $\exp(\Omega(n))$ steps to verify $MP(G_n)_{a_w} \not\cong MP(G_n)_{b_w}$.*

The Multipede graphs are of particular interest, because they are a generalization of the CFI graphs, and thus also hard for resolution, and additionally they can be constructed to be asymmetric. Hence the global symmetry rule is insufficient to get short proofs concerning Multipedes. However, as we will prove, the local symmetry of the CFI-gadgets can be used by the local symmetry rule.

The automorphism group of a Multipede graph is closely related to the solution set of a linear equation system. As a consequence of Lemma 10, any automorphism φ of a Multipede can be uniquely specified by the set of feet $Y := \{w \in W \mid \varphi(a_w) = b_w\}$ for which the a - b -pairs are swapped. The set Y represents a valid automorphism exactly if for every CFI-gadget in the graph, an even number of incident feet is swapped.

Using linear algebra, we can encode a subset $Y \subseteq W = \{w_1, \dots, w_n\}$ uniquely as a vector $\mathbf{y} \in \mathbb{F}_2^n$, by setting $\mathbf{y}_i = 1$ if and only if $w_i \in Y$ for all i . Then the evenness-condition, which the CFI-gadgets require, can be expressed as a set of linear equations:

$$\text{for all } v \in V: \sum_{w_i \in N_G(v)} \mathbf{y}_i = 0.$$

We can write the equations in matrix form: Let $G = (\{v_1, \dots, v_m\}, \{w_1, \dots, w_n\}, E)$ be a bipartite graph. Define $\mathbf{M}(G) \in \mathbb{F}_2^{m \times n}$ as follows: $\mathbf{M}(G)_{i,j} := \begin{cases} 1 & \text{if } \{v_i, w_j\} \in E \\ 0 & \text{otherwise.} \end{cases}$

The solutions of the linear equation system $\mathbf{M}(G)\mathbf{y} = \mathbf{0}$ correspond to the automorphisms of $MP(G)$. We will show how to apply resolution and the symmetry rule to linear equations, and extend our results to Multipedes.

2.5 Encoding Linear Equations

Linear equations over finite fields have been used to show lower bounds in Proof Complexity. For example, the Tseitin formulas are constructed from graphs, representing a system of linear equations over \mathbb{F}_2 , and are hard for resolution [19]. In the next section we will show that by adding the symmetry rule to resolution, we get short proofs for linear equations.

To work with linear equations, some basic definitions and notations from linear algebra are needed. Let K be a field and $n, m \in \mathbb{N}$. We write $K^{m \times n}$ for the set of all m by n matrices over K . Symbols for matrices will be written in boldface. Given a matrix $\mathbf{A} \in K^{m \times n}$ and numbers $i \in [1, m]$ and $j \in [1, n]$, we write $\mathbf{A}_{i,j}$ for the element at the i -th row and j -th column of \mathbf{A} . We write K^n for the set of all n -element vectors. For our purposes they can be treated like single-column matrices, i.e., $K^n = K^{n \times 1}$.

We write $\mathbf{0}$ for a vector consisting of zeros, where its size is clear from context. Similarly $\mathbf{1}$ is a vector filled with ones.

Applying resolution to a linear equation system means providing a refutation certifying that the system cannot be solved, if that is indeed the case. The following lemma is essential in proving an equation system unsolvable:

► **Lemma 16.** *Let $\mathbf{A} \in \mathbb{F}_p^{m \times n}$ and $\mathbf{b} \in \mathbb{F}_p^m$. If the equation system $\mathbf{A}\mathbf{x} = \mathbf{b}$ does not have a solution $\mathbf{x} \in \mathbb{F}_p^n$, then there exists some $\mathbf{v} \in \mathbb{F}_p^m$ such that $\mathbf{v}\mathbf{A} = \mathbf{0}$ and $\mathbf{v} \cdot \mathbf{b} = 1$.*

Proof. Since the equation system does not have a solution, applying the Gaussian elimination algorithm yields the equation $0 = 1$. Writing the row operations used by the algorithm as a vector, we get the sought-after \mathbf{v} . ◀

We require some notation for standard operations from linear algebra.

Let $\mathbf{r} = (r_1, \dots, r_n) \in K^n$ and $\mathbf{A} \in K^{m \times n}$. $\text{supp}(\mathbf{r}) := \{i \in \{1, \dots, n\} \mid r_i \neq 0\}$ is the *support* of \mathbf{r} . $\mathbf{A}_{i,*} := (\mathbf{A}_{i,1}, \dots, \mathbf{A}_{i,n}) \in K^n$ is the i -th row of \mathbf{A} . $\text{diag}(\mathbf{r}) \in K^{n \times n}$ is the diagonal matrix with diagonal entries equal to \mathbf{r} . $\Sigma\mathbf{A} := \sum_{i=1}^m \mathbf{A}_{i,*} = \mathbf{1} \cdot \mathbf{A}$ is the row sum of \mathbf{A} . Let $\mathbf{v} = (v_1, \dots, v_n) \in K^n$. Then $\mathbf{r}|_{\mathbf{v}} \in K^n$ is the *restriction* of \mathbf{r} to the support of \mathbf{v} , defined by $(\mathbf{r}|_{\mathbf{v}})_i := \begin{cases} r_i & \text{if } v_i \neq 0 \\ 0 & \text{if } v_i = 0 \end{cases}$ for $i \in \{1, \dots, n\}$.

To encode linear equations as CNF formulas, we first introduce variables which correspond to the solution vector of the linear equation system: $\text{Vars} := \{\xi_{i,k} \mid i \in [1, n] \text{ and } k \in \mathbb{F}_p\}$. For a given vector $\mathbf{x} \in \mathbb{F}_p^n$, the corresponding assignment to the variables would set $\xi_{i,k}$ to *true* if and only if $\mathbf{x}_i = k$.

Our CNF formula has a clause for every \mathbf{x} with $\mathbf{A}\mathbf{x} \neq \mathbf{b}$, ensuring the formula is *false* under the assignment corresponding to \mathbf{x} . For every row (\mathbf{a}, b) of the equation system, we consider all \mathbf{x} with $\mathbf{a} \cdot \mathbf{x} \neq b$. We can restrict \mathbf{x} to the components for which \mathbf{a} is nonzero.

$$P(\mathbf{a}, b) := \{\mathbf{x} \in \mathbb{F}_p^n \mid \mathbf{a} \cdot \mathbf{x} \neq b \text{ and } \text{supp}(\mathbf{x}) \subseteq \text{supp}(\mathbf{a})\}.$$

The formula for the row (\mathbf{a}, b) is then defined as follows:

$$F(\mathbf{a}, b) := \bigwedge_{\mathbf{x} \in P(\mathbf{a}, b)} C_{\mathbf{a}}(\mathbf{x}), \text{ where } C_{\mathbf{a}}(\mathbf{x}) := \bigvee_{i \in \text{supp}(\mathbf{a})} \overline{\xi_{i, \mathbf{x}_i}}.$$

We extend this definition to whole systems of equations: $F(\mathbf{A}, \mathbf{b}) := \bigwedge_{i=1}^m F(\mathbf{A}_{i,*}, \mathbf{b}_i)$. Notice that assigning *false* to every variable satisfies $F(\mathbf{A}, \mathbf{b})$, but this assignment does not represent a vector. For this reason, we additionally need the clauses $V := \bigwedge_{i=1}^n \bigvee_{k \in \mathbb{F}_p} \xi_{i,k}$.

► **Lemma 17.** *Let $\mathbf{A} \in \mathbb{F}_p^{m \times n}$ and $\mathbf{b} \in \mathbb{F}_p^m$. There exists an $\mathbf{x} \in \mathbb{F}_p^n$ with $\mathbf{A}\mathbf{x} = \mathbf{b}$ if and only if $F(\mathbf{A}, \mathbf{b}) \wedge V$ is satisfiable.*

Proof. \implies : Assume $\mathbf{A}\mathbf{x} = \mathbf{b}$. Define an assignment $\varphi : \text{Vars} \rightarrow \mathbb{B}$ such that $\varphi(\xi_{i,k}) = 1$ if and only if $\mathbf{x}_i = k$. It is easy to see that $\varphi(V) = 1$. Let $j \in [1, m]$ and $\mathbf{x}' \in P(\mathbf{A}_{j,*}, \mathbf{b}_j)$. Then $\mathbf{A}_{j,*} \cdot \mathbf{x}' \neq \mathbf{b}_j = \mathbf{A}_{j,*} \cdot \mathbf{x}$. Hence there exists $i \in \text{supp}(\mathbf{A}_{j,*})$ such that $\mathbf{x}_i \neq \mathbf{x}'_i$. Therefore $\varphi(\xi_{i, \mathbf{x}'_i}) = 0$, so $\varphi(C_{\mathbf{A}_{j,*}}(\mathbf{x}')) = 1$. Then $\varphi(F(\mathbf{A}_{j,*}, \mathbf{b}_j)) = 1$ and thus $F(\mathbf{A}, \mathbf{b})$ is satisfied by φ .

\impliedby : Assume that we have an assignment φ with $\varphi(F(\mathbf{A}, \mathbf{b})) = 1$ and $\varphi(V) = 1$. For all $i \in [1, n]$ there exists a $k \in \mathbb{F}_p$ such that $\varphi(\xi_{i,k}) = 1$. Define $\mathbf{x}_i := k$. Towards a contradiction, assume there exists $j \in [1, m]$ with $\mathbf{b}_j \neq \mathbf{A}_{j,*} \cdot \mathbf{x}$. Then $\mathbf{x}|_{\mathbf{A}_{j,*}} \in P(\mathbf{A}_{j,*}, \mathbf{b}_j)$ and $\varphi(C_{\mathbf{A}_{j,*}}(\mathbf{x})) = 1$. Hence there must exist an $i \in \text{supp}(\mathbf{A}_{j,*})$ such that $\varphi(\xi_{i, \mathbf{x}_i}) = 0$, which contradicts our construction of \mathbf{x} . Therefore $\mathbf{A}\mathbf{x} = \mathbf{b}$. ◀

3 Linear-sized Refutations for Non-Isomorphism of CFI graphs

Due to the symmetric nature of the CFI graphs, using the symmetry rule gives us linear-sized resolution proofs of non-isomorphism for a pair of these graphs.

► **Theorem 18.** *Let G be a graph with at least one edge. Then*

$$F(X(G), \tilde{X}(G)) \vdash_{\mathcal{O}(|F(X(G), \tilde{X}(G))|)}^{\text{SRC-I}} \perp.$$

For a full proof of Theorem 18, see [15]. Here we only sketch the main ideas of the proof.

Proof sketch. The resolution refutation is created recursively. We remove a special vertex or edge from G to obtain a smaller graph G' , get a short proof for $F(X(G'), \tilde{X}(G')) \vdash^{\text{SRC-I}} \perp$, and then use this to build a short proof for $F(X(G), \tilde{X}(G)) \vdash^{\text{SRC-I}} \perp$.

One of the following cases surely holds for G : It has an edge e on a cycle, or it has a vertex v of degree 1. In the latter case, there is only one way to map the CFI-gadgets of $X(G)$ and $\tilde{X}(G)$ around v to each other. This can be proved using standard resolution in a constant number of steps.

In the case of a cycle, we have a certain symmetry in $X(G)$, which the symmetry rule can exploit: Twisting every edge along the cycle is an automorphism. Hence, if $X(G)$ and $\tilde{X}(G)$ are isomorphic, then also this twisted version of $X(G)$ is isomorphic to $\tilde{X}(G)$. To show non-isomorphism, it is then sufficient to prove non-isomorphism for one of these versions, and the rest follows by symmetry - a symmetry which can be written in SRC-I. Choosing one of the versions is equivalent to having a fixed mapping of e .

In both cases, we fix the mapping of a vertex or an edge, which makes it possible to build up the resolution refutation from recursively smaller proofs. Each time we only add a constant number of steps, in sum yielding a proof of linear size. ◀

This result stands in contrast to the exponential lower bound of Theorem 13.



■ **Figure 3** A visualization of $\vartheta(\mathbf{x}, \mathbf{y})$.

4 Polynomial-sized Refutations for Linear Equations

4.1 Linear Combinations

The usual approach to showing that a system of linear equations is inconsistent, is to build a linear combination of the equations to derive the obvious contradiction $0 = 1$. This method is complete by Lemma 16. We recreate this process in the resolution proof system. However we need to ensure that the support of equations we create along the way is not excessively large. We will do so by using the the symmetry rule.

Note that the formula $F(\mathbf{a}, b)$ is invariant under linear scaling of the inputs: For any $k \in \mathbb{F}_p \setminus \{0\}$ we have $\text{supp}(\mathbf{a}) = \text{supp}(k\mathbf{a})$ and $P(\mathbf{a}, b) = P(k\mathbf{a}, kb)$. Hence $F(\mathbf{a}, b) = F(k\mathbf{a}, kb)$.

For the computation of linear combinations, we use the following definition. For $\theta \subseteq [1, n]$ define $\Omega(\theta) := \{\bigvee_{i \in \theta} \overline{\xi_{i, \mathbf{x}_i}} \mid \mathbf{x} \in \mathbb{F}_p^n \text{ with } \text{supp}(\mathbf{x}) \subseteq \theta\}$. Together, the clauses in $\Omega(\theta)$ forbid all possible assignments to the components in range θ . In a sense, $\Omega(\theta)$ is our basic building block for contradictions.

► **Lemma 19.** *Let $\theta \subseteq [1, n]$. Then $\Omega(\theta) \wedge V \vdash_{\frac{p^{|\theta|+1}-p}{p-1}} \perp$.*

Proof. Induction over $|\theta|$. If $\theta = \emptyset$ then $\Omega(\theta) = \{\bigvee \emptyset\} = \{\perp\}$, so $\Omega(\theta) \vdash_0 \perp$.

Induction step: $\theta = \theta' \cup \{j\}$. It holds:

$$\begin{aligned} \Omega(\theta) &= \left\{ \bigvee_{i \in \theta} \overline{\xi_{i, \mathbf{x}_i}} \mid \mathbf{x} \in \mathbb{F}_p^n \text{ with } \text{supp}(\mathbf{x}) \subseteq \theta \right\} \\ &= \left\{ \overline{\xi_{j, \mathbf{x}_j}} \vee \bigvee_{i \in \theta'} \overline{\xi_{i, \mathbf{x}_i}} \mid \mathbf{x} \in \mathbb{F}_p^n \text{ with } \text{supp}(\mathbf{x}) \subseteq \theta \right\} \\ &= \left\{ \overline{\xi_{j, k}} \vee \bigvee_{i \in \theta'} \overline{\xi_{i, \mathbf{x}_i}} \mid \mathbf{x} \in \mathbb{F}_p^n \text{ with } \text{supp}(\mathbf{x}) \subseteq \theta' \text{ and } k \in \mathbb{F}_p \right\} \\ &= \{ \overline{\xi_{j, k}} \vee c' \mid c' \in \Omega(\theta'), k \in \mathbb{F}_p \} \end{aligned}$$

For each $c' \in \Omega(\theta')$, we can derive the clause c' by resolving $\bigvee_{k \in \mathbb{F}_p} \xi_{j, k}$ from V with the clauses from $\Omega(\theta)$. Doing this for all $c' \in \Omega(\theta')$ takes $p \cdot |\Omega(\theta')| = |\Omega(\theta)| = p^{|\theta|}$ resolution steps. By induction, we can then derive \perp from $\Omega(\theta')$ and V in $\frac{p^{|\theta'+1|-p}}{p-1} = \frac{p^{|\theta|-p}}{p-1}$ steps. The total number of steps taken is $\frac{p^{|\theta|-p}}{p-1} + p^{|\theta|} = \frac{p^{|\theta|+1}-p}{p-1}$. ◀

When we sum two vectors \mathbf{x} and \mathbf{y} , some components may become zero which were nonzero before. The following definition captures this phenomenon: $\vartheta(\mathbf{x}, \mathbf{y}) := (\text{supp}(\mathbf{x}) \cup \text{supp}(\mathbf{y})) \setminus \text{supp}(\mathbf{x} + \mathbf{y})$. If a coefficient vanishes in a sum, it has to appear in both summands: $\vartheta(\mathbf{x}, \mathbf{y}) \subseteq \text{supp}(\mathbf{x}) \cap \text{supp}(\mathbf{y})$ (see Figure 3).

With these ingredients, we can finally explain the process of building sums using resolution.

► **Theorem 20 (Sum Resolution).** *Let $\mathbf{a} \in \mathbb{F}_p^{2 \times n}$ and $\mathbf{b} \in \mathbb{F}_p^2$. Define $\theta := \vartheta(\mathbf{a}_1, \mathbf{a}_2)$, where \mathbf{a}_i is the i -th row of \mathbf{a} . For all $c \in F(\Sigma \mathbf{a}, \Sigma \mathbf{b})$ it holds: $F(\mathbf{a}_1, \mathbf{b}_1) \cup F(\mathbf{a}_2, \mathbf{b}_2) \cup V \vdash_{\frac{w}{2(p^{|\theta|-1})}} c$.*

Proof. Let $c \in F(\Sigma\mathbf{a}, \Sigma\mathbf{b})$. By definition of F , there exists some $\mathbf{x} \in P(\Sigma\mathbf{a}, \Sigma\mathbf{b})$ such that $c = \bigvee_{i \in \text{supp}(\Sigma\mathbf{a})} \overline{\xi_{i, \mathbf{x}_i}}$. The resolution derivation will have to get rid of all variables corresponding to components in θ . For $\kappa \in \{1, 2\}$ define $R^\kappa := \bigvee_{i \in \text{supp}(\mathbf{a}^\kappa) \setminus \theta} \overline{\xi_{i, \mathbf{x}_i}}$. From this we will build the desired clause c . It holds: $(\text{supp}(\mathbf{a}_1) \cup \text{supp}(\mathbf{a}_2)) \setminus \theta = (\text{supp}(\mathbf{a}_1) \cup \text{supp}(\mathbf{a}_2)) \cap \text{supp}(\Sigma\mathbf{a}) = \text{supp}(\Sigma\mathbf{a})$. Thus

$$\begin{aligned} R^1 \vee R^2 &= \bigvee_{i \in \text{supp}(\mathbf{a}_1) \setminus \theta} \overline{\xi_{i, \mathbf{x}_i}} \vee \bigvee_{i \in \text{supp}(\mathbf{a}_2) \setminus \theta} \overline{\xi_{i, \mathbf{x}_i}} \\ &= \bigvee_{i \in (\text{supp}(\mathbf{a}_1) \cup \text{supp}(\mathbf{a}_2)) \setminus \theta} \overline{\xi_{i, \mathbf{x}_i}} \\ &= \bigvee_{i \in \text{supp}(\Sigma\mathbf{a})} \overline{\xi_{i, \mathbf{x}_i}} = c \end{aligned}$$

Consider an arbitrary $\mathbf{y} \in \mathbb{F}_p^n$ with $\text{supp}(\mathbf{y}) \subseteq \theta$. Since $\text{supp}(\mathbf{y}) \cap \text{supp}(\Sigma\mathbf{a}) = \emptyset$, we have $\Sigma\mathbf{a} \cdot \mathbf{y} = 0$. There exists κ with $\mathbf{a}_\kappa \cdot (\mathbf{x} + \mathbf{y}) \neq b_\kappa$, because otherwise we would have $\Sigma\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2 = \mathbf{a}_1 \cdot (\mathbf{x} + \mathbf{y}) + \mathbf{a}_2 \cdot (\mathbf{x} + \mathbf{y}) = (\mathbf{a}_1 + \mathbf{a}_2) \cdot (\mathbf{x} + \mathbf{y}) = \Sigma\mathbf{a} \cdot \mathbf{x} + \Sigma\mathbf{a} \cdot \mathbf{y} = \Sigma\mathbf{a} \cdot \mathbf{x}$, which contradicts $\mathbf{x} \in P(\Sigma\mathbf{a}, \Sigma\mathbf{b})$.

Hence $(\mathbf{x} + \mathbf{y})|_{\mathbf{a}_\kappa} \in P(\mathbf{a}_\kappa, \mathbf{b}_\kappa)$ and we have a clause $c'(\mathbf{y}) := \bigvee_{i \in \text{supp}(\mathbf{a}_\kappa)} \overline{\xi_{i, (\mathbf{x} + \mathbf{y})_i}} \in F(\mathbf{a}_\kappa, \mathbf{b}_\kappa)$. It holds:

$$\begin{aligned} c'(\mathbf{y}) &= \bigvee_{i \in \text{supp}(\mathbf{a}_\kappa) \cap \theta} \overline{\xi_{i, (\mathbf{x} + \mathbf{y})_i}} \vee \bigvee_{i \in \text{supp}(\mathbf{a}_\kappa) \setminus \theta} \overline{\xi_{i, (\mathbf{x} + \mathbf{y})_i}} \\ &= \bigvee_{i \in \text{supp}(\mathbf{a}_\kappa) \cap \theta} \overline{\xi_{i, \mathbf{y}_i}} \vee \bigvee_{i \in \text{supp}(\mathbf{a}_\kappa) \setminus \theta} \overline{\xi_{i, \mathbf{x}_i}} \\ &= \bigvee_{i \in \theta} \overline{\xi_{i, \mathbf{y}_i}} \vee R^\kappa \end{aligned}$$

Now, looking at the set $C := \{c'(\mathbf{y}) \mid \mathbf{y} \in \mathbb{F}_p^n\} \subseteq F(\mathbf{a}_1, \mathbf{b}_1) \cup F(\mathbf{a}_2, \mathbf{b}_2)$, note that for every clause $d \in \Omega(\theta)$, we have $d \vee R^1 \in C$ or $d \vee R^2 \in C$. By Lemma 19 and Lemma 6 we can resolve the clauses in C together with V to obtain $R^1 \vee R^2 = c$ or stronger. Since $p \geq 2$, this takes at most $\frac{p^{|\theta|+1}-p}{p-1} \leq \frac{p^{|\theta|+1}-p}{p/2} = 2(p^{|\theta|} - 1)$ resolution steps. \blacktriangleleft

By applying Theorem 20 iteratively, we can construct the formulas for linear combinations with an arbitrary number of summands. This method, however, is inefficient since the produced intermediate equations may accumulate more and more variables, leading to an exponential growth of the number of required clauses.

We solve this problem by deriving only a single representative clause for intermediate results, and using the local symmetry rule to derive more clauses as necessary.

4.2 Local Symmetry in Equations

We want to understand which symmetries the formulas corresponding to linear equations have. For $\mathbf{d} \in \mathbb{F}_p^n$ define $\Delta_{\mathbf{d}} : \text{Vars} \rightarrow \text{Vars} : \xi_{i,k} \mapsto \Delta_{\mathbf{d}}(\xi_{i,k}) := \xi_{i, k + \mathbf{d}_i}$. This bijective map is a translation by \mathbf{d} of the vector corresponding to the variables.

► Lemma 21. *Let $b \in \mathbb{F}_p$ and $\mathbf{a}, \mathbf{d} \in \mathbb{F}_p^n$. Then $\Delta_{\mathbf{d}} \in \text{Sym}(F(\mathbf{a}, b))$ if and only if $\mathbf{a} \cdot \mathbf{d} = 0$.*

Proof. \Leftarrow : Assume $\mathbf{a} \cdot \mathbf{d} = 0$. Let $c = C_{\mathbf{a}}(\mathbf{x}) = \bigvee_{i \in \text{supp}(\mathbf{a})} \overline{\xi_{i, \mathbf{x}_i}} \in F(\mathbf{a}, b)$ for some $\mathbf{x} \in P(\mathbf{a}, b)$. We have $\mathbf{a} \cdot (\mathbf{x} + \mathbf{d}) = \mathbf{a} \cdot \mathbf{x} \neq b$. Hence $(\mathbf{x} + \mathbf{d})|_{\mathbf{a}} \in P(\mathbf{a}, b)$ and thus $\Delta_{\mathbf{d}}(c) = \bigvee_{i \in \text{supp}(\mathbf{a})} \overline{\xi_{i, \mathbf{x}_i + \mathbf{d}_i}} = C_{\mathbf{a}}(\mathbf{x} + \mathbf{d}) = C_{\mathbf{a}}((\mathbf{x} + \mathbf{d})|_{\mathbf{a}}) \in F(\mathbf{a}, b)$.

58:12 Resolution with Symmetry Rule Applied to Linear Equations

\implies : Assume $\mathbf{a} \cdot \mathbf{d} \neq 0$. Then $\mathbf{a} \neq \mathbf{0}$ and hence there exists a vector $\mathbf{x} \in P(\mathbf{a}, b)$ with $\mathbf{a} \cdot \mathbf{x} = b - \mathbf{a} \cdot \mathbf{d} \neq b$. But then $\mathbf{a} \cdot (\mathbf{x} + \mathbf{d}) = b$ and thus $\Delta_{\mathbf{d}}(C_{\mathbf{a}}(\mathbf{x})) = C_{\mathbf{a}}(\mathbf{x} + \mathbf{d}) \notin F(\mathbf{a}, b)$. Hence $\Delta_{\mathbf{d}} \notin \text{Sym}(F(\mathbf{a}, b))$. \blacktriangleleft

► **Corollary 22.** *Let $\mathbf{A} \in \mathbb{F}_p^{m \times n}$, $\mathbf{b} \in \mathbb{F}_p^m$ and $\mathbf{d} \in \mathbb{F}_p^n$. If $\mathbf{A} \cdot \mathbf{d} = \mathbf{0}$, then $\Delta_{\mathbf{d}} \in \text{Sym}(F(\mathbf{A}, \mathbf{b}))$.*

Note the following: If $\mathbf{d}, \mathbf{d}' \in \mathbb{F}_p^n$ such that $\mathbf{d}|_{\mathbf{a}} = \mathbf{d}'|_{\mathbf{a}}$, then for all $c \in F(\mathbf{a}, b)$ it holds: $\Delta_{\mathbf{d}}(c) = \Delta_{\mathbf{d}'}(c)$. In particular, $\Delta_{\mathbf{d}} \in \text{Sym}(F(\mathbf{a}, b))$ implies $\Delta_{\mathbf{d}'} \in \text{Sym}(F(\mathbf{a}, b))$. The condition $\mathbf{d}|_{\mathbf{a}} = \mathbf{d}'|_{\mathbf{a}}$ can equivalently be expressed using matrix algebra: $\text{diag}(\mathbf{a})\mathbf{d} = \text{diag}(\mathbf{a})\mathbf{d}'$.

To make use of the symmetry rule, we want to apply the symmetries of $F(\mathbf{A}, \mathbf{b})$ to derive clauses of $F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$. From the statements and Corollary 22, we conclude the following relation between $\text{Sym}(F(\mathbf{A}, \mathbf{b}))$ and $\text{Sym}(F(\Sigma\mathbf{A}, \Sigma\mathbf{b}))$.

► **Lemma 23.** *Let $\mathbf{A} \in \mathbb{F}_p^{m \times n}$, $\mathbf{b} \in \mathbb{F}_p^m$ and $\mathbf{d} \in \mathbb{F}_p^n$ with $\Delta_{\mathbf{d}} \in \text{Sym}(F(\Sigma\mathbf{A}, \Sigma\mathbf{b}))$. If there exists $\mathbf{d}' \in \mathbb{F}_p^n$ such that $\mathbf{A}\mathbf{d}' = \mathbf{0}$ and $\text{diag}(\Sigma\mathbf{A})\mathbf{d}' = \text{diag}(\Sigma\mathbf{A})\mathbf{d}$, then $\Delta_{\mathbf{d}'} \in \text{Sym}(F(\mathbf{A}, \mathbf{b}))$ and $\Delta_{\mathbf{d}'}(c) = \Delta_{\mathbf{d}}(c)$ for all $c \in F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$.*

Concerning V , the symmetries are simpler: For any $\mathbf{d} \in \mathbb{F}_p^n$ we have $\Delta_{\mathbf{d}} \in \text{Sym}(V)$.

We will assume that the coefficient matrices \mathbf{A} in the following have at most L nonzero entries in each row. In other words, the width of \mathbf{A} is at most L .

► **Theorem 24.** *Let $\mathbf{A} \in \mathbb{F}_p^{m \times n}$ and $\mathbf{b} \in \mathbb{F}_p^m$. For any $H \subseteq F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$ it holds: $F(\mathbf{A}, \mathbf{b}) \wedge V \vdash_{\mathcal{O}(m^{\Theta(p)}p^{L+1})+|H|}^{\text{SRC-II}} H$.*

Proof. Define $\lambda := \frac{\log(2)}{\log(p/(p-1))}$ and $f(x) := Cp^{L+1}x^\lambda$ for some constant C chosen later. Regarding the relationship between λ and p we have $\lambda \sim \log(2)p$ (i.e., $\lim_{n \rightarrow \infty} \lambda / \log(2)p = 1$)).

We prove the following by induction over the number of equations m : For any $H \subseteq F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$ it holds: $F(\mathbf{A}, \mathbf{b}) \wedge V \vdash_{f(m)+|H|}^{\text{SRC-II}} H$.

Induction basis: $m = 0$. In this case, we have $\Sigma\mathbf{A} = \mathbf{0}$ and $\Sigma\mathbf{b} = \mathbf{0}$. Then $F(\Sigma\mathbf{A}, \Sigma\mathbf{b}) = \emptyset$; hence $H = \emptyset$ and we have nothing to prove.

Induction step: $m - 1 \rightarrow m$. Here we have two cases:

Case 1: Symmetric sum. For this case we assume that for all \mathbf{d} with $\Delta_{\mathbf{d}} \in \text{Sym}(F(\Sigma\mathbf{A}, \Sigma\mathbf{b}))$ we have a \mathbf{d}' with $\mathbf{d}|_{\Sigma\mathbf{A}} = \mathbf{d}'|_{\Sigma\mathbf{A}}$ and $\Delta_{\mathbf{d}'} \in \text{Sym}(F(\mathbf{A}, \mathbf{b}))$. Thanks to this property, all the symmetries of $F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$ are already present in $F(\mathbf{A}, \mathbf{b})$ and can be used by the symmetry rule. So we only need to derive a few clauses of $F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$ to obtain a set allowing us to generate all clauses via symmetries. This which can be done by inductively applying Theorem 20 as follows.

Define \mathbf{A}' and \mathbf{b}' to be the first $m - 1$ rows of \mathbf{A} and \mathbf{b} respectively. Define $\theta := \vartheta(\Sigma\mathbf{A}', \mathbf{A}_{m,*})$. We have $|\theta| \leq |\text{supp}(\mathbf{A}_{m,*})| \leq L$. If $\Sigma\mathbf{A} \neq \mathbf{0}$, then for each $k \neq \Sigma\mathbf{b}$ there exists a vector \mathbf{z}^k with $\text{supp}(\mathbf{z}^k) \subseteq \text{supp}(\Sigma\mathbf{A})$ such that $\Sigma\mathbf{A} \cdot \mathbf{z}^k = k$. Define $G := \{C_{\Sigma\mathbf{A}}(\mathbf{z}^k) \mid k \in \mathbb{F}_p \setminus \{\Sigma\mathbf{b}\}\} \subseteq F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$. Then $|G| = p - 1$.

Using Theorem 20, we get $F(\Sigma\mathbf{A}', \Sigma\mathbf{b}') \wedge F(\mathbf{A}_{m,*}, \mathbf{b}_m) \wedge V \vdash_{|G| \cdot \mathcal{O}(p^L)}^w G$. This derivation only uses a subset $H' \subseteq F(\Sigma\mathbf{A}', \Sigma\mathbf{b}')$ of at most $|H'| \leq \mathcal{O}(p^{L+1})$ clauses. By induction, it holds that $F(\mathbf{A}', \mathbf{b}') \wedge V \vdash_{f(m-1)+|H'|}^{\text{SRC-II}} H'$. We can combine these derivations by Lemma 5 to obtain $F(\mathbf{A}, \mathbf{b}) \wedge V \vdash_{f(m-1)+\mathcal{O}(p^{L+1})}^{\text{SRC-II}} G$. Using $\lambda \geq 1$, we take in total $f(m-1) + Cp^{L+1} = Cp^{L+1}(m-1)^\lambda + Cp^{L+1} \leq Cp^{L+1}m^\lambda = f(m)$ steps, for some constant C .

Now we show that G is a generator for H : Let $c \in F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$. Then $c = C_{\Sigma\mathbf{A}}(\mathbf{x})$ for some $\mathbf{x} \in P(\Sigma\mathbf{A}, \Sigma\mathbf{b})$. Define $\mathbf{d} := \mathbf{x} - \mathbf{z}^{\Sigma\mathbf{A} \cdot \mathbf{x}}$. Then $\Sigma\mathbf{A} \cdot \mathbf{d} = 0$, so $\Delta_{\mathbf{d}} \in \text{Sym}(F(\Sigma\mathbf{A}, \Sigma\mathbf{b}))$. Hence there exists a $\varphi \in \text{Sym}(F(\mathbf{A}, \mathbf{b}))$ such that $\varphi(C_{\Sigma\mathbf{A}}(\mathbf{z}^{\Sigma\mathbf{A} \cdot \mathbf{x}})) = \Delta_{\mathbf{d}}(C_{\Sigma\mathbf{A}}(\mathbf{z}^{\Sigma\mathbf{A} \cdot \mathbf{x}})) = C_{\Sigma\mathbf{A}}(\mathbf{x}) = c$. We can apply the local symmetry rule to derive c from G in a single step, using the symmetries of $F(\mathbf{A}, \mathbf{b})$. Repeating this for every $c \in H$ yields $F(\mathbf{A}, \mathbf{b}) \wedge V \vdash_{f(m)+|H|}^{\text{SRC-II}} H$.

If $\Sigma\mathbf{A} = \mathbf{0}$ and $\Sigma\mathbf{b} \neq \mathbf{0}$ then $F(\Sigma\mathbf{A}, \Sigma\mathbf{b}) = \{C_0(\mathbf{0})\} = \{\perp\} =: G$, which can be derived in at most Cp^{L+1} steps, again using Theorem 20.

If $\Sigma\mathbf{A} = \mathbf{0}$ and $\Sigma\mathbf{b} = \mathbf{0}$ then $F(\Sigma\mathbf{A}, \Sigma\mathbf{b}) = \emptyset$. We treat this the same way as the case $m = 0$.

Case 2: Composite. If Case 1 does not apply, the following must hold by Lemma 23: For some \mathbf{d} with $\Sigma\mathbf{A} \cdot \mathbf{d} = 0$, the equations $\mathbf{A}\mathbf{d}' = \mathbf{0}$ and $\text{diag}(\Sigma\mathbf{A})\mathbf{d}' = \text{diag}(\Sigma\mathbf{A})\mathbf{d}$ have no common solution \mathbf{d}' .

Applying Lemma 16 to the combined inconsistent equations, we have \mathbf{v}, \mathbf{w} such that $\mathbf{v}\mathbf{A} + \mathbf{w}\text{diag}(\Sigma\mathbf{A}) = \mathbf{0}$ and $\mathbf{w}\text{diag}(\Sigma\mathbf{A})\mathbf{d} \neq \mathbf{0}$. We will use the vector \mathbf{v} to decompose \mathbf{A} into two smaller matrices, each contributing independently to the derivation of H . First we show that \mathbf{v} has special properties which make this *divide and conquer* approach work. Then we need to ensure that the sub-problems are not too large for our proof length bound f .

It holds: $\mathbf{v}\mathbf{A} = -\mathbf{w}\text{diag}(\Sigma\mathbf{A})$; thus $\mathbf{v}\mathbf{A}\mathbf{d} = -\mathbf{w}\text{diag}(\Sigma\mathbf{A})\mathbf{d} \neq \mathbf{0}$. For all $i \in [1, n]$, if $(\Sigma\mathbf{A})_i = 0$, then $(\mathbf{v}\mathbf{A})_i = (-\mathbf{w}\text{diag}(\Sigma\mathbf{A}))_i = -\mathbf{w}_i(\Sigma\mathbf{A})_i = 0$. Hence $\text{supp}(\mathbf{v}\mathbf{A}) \subseteq \text{supp}(\Sigma\mathbf{A})$. We show that $\mathbf{v}\mathbf{A}$ and $\Sigma\mathbf{A}$ are linearly independent: Let $\alpha_1, \alpha_2 \in \mathbb{F}_p$ such that $\alpha_1\mathbf{v}\mathbf{A} + \alpha_2\Sigma\mathbf{A} = \mathbf{0}$. Then $0 = \alpha_1\mathbf{v}\mathbf{A}\mathbf{d} + \alpha_2\Sigma\mathbf{A}\mathbf{d} = \alpha_1\mathbf{v}\mathbf{A}\mathbf{d}$, which implies $\alpha_1 = 0$. Since $\Sigma\mathbf{A} \neq \mathbf{0}$, we also have $\alpha_2 = 0$.

Let $k_1 \in \arg \max_{k \in \mathbb{F}_p} |\{i \mid \mathbf{v}_i = k\}|$ be the most common component of \mathbf{v} . Let $k_2 \in \arg \max_{k \in \mathbb{F}_p, k \neq k_1} |\{i \mid \mathbf{v}_i = k\}|$ be the second most common component of \mathbf{v} . Since $\mathbf{v}\mathbf{A}$ is linearly independent from $\Sigma\mathbf{A}$, we have $\mathbf{v} \neq k \cdot \mathbf{1}$ for all $k \in \mathbb{F}_p$, so there are at least two different components in \mathbf{v} . Hence k_1 and k_2 exist. Define m_i to be the number of times k_i occurs in \mathbf{v} . We have $m_i \geq 1$ for $i \in \{1, 2\}$. Furthermore $m_1 \geq m/p$ and $m_2 \geq (m - m_1)/(p - 1)$.

Define $\mathbf{v}^1 := \mathbf{v} - k_1\mathbf{1}$ and $\mathbf{v}^2 := k_2\mathbf{1} - \mathbf{v}$. It holds: $\mathbf{v}^1 + \mathbf{v}^2 = (k_2 - k_1)\mathbf{1}$. By subtracting k_i from every component, we get exactly m_i zeros in \mathbf{v}^i , i.e. $|\text{supp}(\mathbf{v}^i)| = m - m_i$.

Towards a contradiction, assume there is some $j \in \vartheta(\mathbf{v}^1\mathbf{A}, \mathbf{v}^2\mathbf{A})$. Then $0 = (\mathbf{v}^1\mathbf{A} + \mathbf{v}^2\mathbf{A})_j = ((k_2 - k_1)\Sigma\mathbf{A})_j$, so $0 = (\Sigma\mathbf{A})_j$. Thus $0 = -\mathbf{w}_j(\Sigma\mathbf{A})_j = (\mathbf{v}\mathbf{A})_j = (\mathbf{v}^1\mathbf{A} + k_1\Sigma\mathbf{A})_j = (\mathbf{v}^1\mathbf{A})_j + k_1(\Sigma\mathbf{A})_j = (\mathbf{v}^1\mathbf{A})_j$. This contradicts the assumption. Hence $\vartheta(\mathbf{v}^1\mathbf{A}, \mathbf{v}^2\mathbf{A}) = \emptyset$. By Theorem 20 we can derive the sum clauses of $\mathbf{v}^1\mathbf{A} + \mathbf{v}^2\mathbf{A}$ in 0 steps, so they are already implied by the summand clauses: $F((\mathbf{v}^1 + \mathbf{v}^2)\mathbf{A}, (\mathbf{v}^1 + \mathbf{v}^2)\mathbf{b}) \subseteq F(\mathbf{v}^1\mathbf{A}, \mathbf{v}^1\mathbf{b}) \cup F(\mathbf{v}^2\mathbf{A}, \mathbf{v}^2\mathbf{b})$. It follows that

$$\begin{aligned} F(\Sigma\mathbf{A}, \Sigma\mathbf{b}) &= F((k_2 - k_1)\Sigma\mathbf{A}, (k_2 - k_1)\Sigma\mathbf{b}) \\ &= F((\mathbf{v}^1 + \mathbf{v}^2)\mathbf{A}, (\mathbf{v}^1 + \mathbf{v}^2)\mathbf{b}) \\ &\subseteq F(\mathbf{v}^1\mathbf{A}, \mathbf{v}^1\mathbf{b}) \cup F(\mathbf{v}^2\mathbf{A}, \mathbf{v}^2\mathbf{b}). \end{aligned}$$

Hence we can partition $H \subseteq F(\Sigma\mathbf{A}, \Sigma\mathbf{b})$ into H_1 and H_2 such that $H_i \subseteq F(\mathbf{v}^i\mathbf{A}, \mathbf{v}^i\mathbf{b})$ for $i \in \{1, 2\}$.

Note that $F(\mathbf{v}^i\mathbf{A}, \mathbf{v}^i\mathbf{b}) = F(\Sigma\text{diag}(\mathbf{v}^i)\mathbf{A}, \Sigma\text{diag}(\mathbf{v}^i)\mathbf{b})$. Since $|\text{supp}(\mathbf{v}^i)| \leq m - 1$, we have at least one zero row each in $\text{diag}(\mathbf{v}^1)\mathbf{A}$ and $\text{diag}(\mathbf{v}^2)\mathbf{A}$. This makes it possible to apply the induction hypothesis, yielding $F(\text{diag}(\mathbf{v}^i)\mathbf{A}, \text{diag}(\mathbf{v}^i)\mathbf{b}) \wedge V \vdash_{f(|\text{supp}(\mathbf{v}^i)|)+|H_i|}^{\text{SRC-II}} H_i$.

Scaling the equations does not produce different clauses, so we have $F(\text{diag}(\mathbf{v}^1)\mathbf{A}, \text{diag}(\mathbf{v}^1)\mathbf{b}) \cup F(\text{diag}(\mathbf{v}^2)\mathbf{A}, \text{diag}(\mathbf{v}^2)\mathbf{b}) \subseteq F(\mathbf{A}, \mathbf{b})$. Then we can combine the derivations of H_1 and H_2 to obtain $F(\mathbf{A}, \mathbf{b}) \wedge V \vdash_{f(|\text{supp}(\mathbf{v}^1)|)+f(|\text{supp}(\mathbf{v}^2)|)+|H|}^{\text{SRC-II}} H$. It holds:

58:14 Resolution with Symmetry Rule Applied to Linear Equations

$f(|\text{supp}(\mathbf{v}^1)|) + f(|\text{supp}(\mathbf{v}^2)|) = f(m - m_1) + f(m - m_2) \leq f(m - m_1) + f(m - (m - m_1)/(p - 1)) =: T(m_1)$. By applying standard calculus techniques to the function T , we find that $T(m_1) \leq f(m)$ for all possible values of m_1 . ◀

► **Corollary 25.** Let $\mathbf{A} \in \mathbb{F}_p^{m \times n}$ and $\mathbf{b} \in \mathbb{F}_p^m$, such that there is no $\mathbf{x} \in \mathbb{F}_p^n$ satisfying $\mathbf{A}\mathbf{x} = \mathbf{b}$. Then there exists a resolution refutation of $F(\mathbf{A}, \mathbf{b}) \wedge V$ using the local symmetry rule, with its length bounded by $\mathcal{O}(m^{\Theta(p)} p^{L+1})$.

5 Linear-sized Refutations for Non-Isomorphism of Multipedes

We can use the result on linear equations to show that there are short resolution proofs for the non-isomorphism of Multipede graphs.

► **Theorem 26.** Let $G = (V, W, E)$ be a connected bipartite graph such that $MP(G)$ is asymmetric, and $\omega \in W$. Then $F(MP(G)_{a_\omega}, MP(G)_{b_\omega})$ has a linear-sized resolution refutation using the local symmetry rule.

Proof. Let $G = (\{v_1, \dots, v_m\}, \{w_1, \dots, w_n\}, E)$ be a connected bipartite graph and $\omega := w_k$ for some k . Our goal is to apply the techniques of the previous section to the formula $F_0 := F(MP(G)_{a_\omega}, MP(G)_{b_\omega})$.

We first inspect the simpler formula $F_1 := F(MP(G), MP(G))$. The solutions of this formula correspond to the automorphisms of $MP(G)$. By applying resolution to F_1 , we can derive the formula $F(\mathbf{M}(G), \mathbf{0})$: Let $i \in [1, m]$. Then

$$\begin{aligned} F(\mathbf{M}(G)_{i,*}, 0) &= \bigwedge_{\mathbf{x} \in P(\mathbf{M}(G)_{i,*}, 0)} \bigvee_{j \in \text{supp}(\mathbf{M}(G)_{i,*})} \overline{\xi_{j, \mathbf{x}_j}} \\ &= \bigwedge_{\mathbf{x} \in P(\mathbf{M}(G)_{i,*}, 0)} \bigvee_{\{v_i, w_j\} \in E} \overline{\xi_{j, \mathbf{x}_j}} \\ &= \bigwedge_{\substack{B \subseteq N_G(v_i) \\ |B| \text{ odd}}} \left(\bigvee_{w_j \in B} \overline{\xi_{j,1}} \vee \bigvee_{w_j \in N_G(v_i) \setminus B} \overline{\xi_{j,0}} \right) \end{aligned}$$

Define $N := N_G(v_i)$. We define $\mathcal{P}_{\text{even}}(N)$ to be the subsets of N with even cardinality. For all $B \subseteq N$ with odd $|B|$ there exists a surjective function $\gamma : \mathcal{P}_{\text{even}}(N) \rightarrow N$ such that for all $S \in \mathcal{P}_{\text{even}}(N) : \gamma(S) \in S \setminus B \cup B \setminus S$. We can make the following resolution derivation from F_1 :

$$\begin{aligned} &\text{Type 1:} && \bigvee_{S \in \mathcal{P}_{\text{even}}(N)} z_{\emptyset, S}^v \\ \forall S \in \mathcal{P}_{\text{even}}(N) \text{ with } w := \gamma(S) \in B \setminus S : &\text{Type 3:} && \overline{z_{\emptyset, S}^v} \vee \overline{y_{a_w, b_w}} \\ \forall S \in \mathcal{P}_{\text{even}}(N) \text{ with } w := \gamma(S) \in S \setminus B : &\text{Type 3:} && \overline{z_{\emptyset, S}^v} \vee \overline{y_{a_w, a_w}} \\ &\implies && \bigvee_{w \in B} \overline{y_{a_w, b_w}} \vee \bigvee_{w \in N \setminus B} \overline{y_{a_w, a_w}}, \end{aligned}$$

taking $|\mathcal{P}_{\text{even}}(N)| \leq 2^{|N|}$ steps. Repeating this process for every B and i takes $\sum_{v \in V} |\mathcal{P}_{\text{odd}}(N_G(v))| \cdot 2^{|N_G(v)|} = \mathcal{O}(|F_1|)$ resolution steps. Define a variable renaming r on $F(\mathbf{M}(G), \mathbf{0})$ as follows:

$$r(\xi_{j, \kappa}) := \begin{cases} y_{a_{w_j}, a_{w_j}} & \text{if } \kappa = 0 \\ y_{a_{w_j}, b_{w_j}} & \text{if } \kappa = 1 \end{cases}$$

Then we have the derivation $F_1 \vdash_{\mathcal{O}(|F_1|)} r(F(\mathbf{M}(G), \mathbf{0}))$. As a consequence, $|F(\mathbf{M}(G), \mathbf{0})| = \mathcal{O}(|F_1|)$. The clauses of $r(V)$ are simply the Type 1 clauses of F_1 .

To apply Theorem 24, we need to translate the symmetries $\Delta_{\mathbf{d}} \in \text{Sym}(F(\mathbf{M}(G), \mathbf{0}))$ into symmetries of F_1 . Let $\mathbf{d} \in \mathbb{F}_2^n$ such that $\mathbf{M}(G)\mathbf{d} = \mathbf{0}$. Define $D := \{w_i \in W \mid \mathbf{d}_i = 1\}$. Then the following map $\psi_{\mathbf{d}}$ is a symmetry of F_1 : for $w \in W$ set $\psi_{\mathbf{d}}(y_{a_w, a_w})$ to be y_{a_w, b_w} if $w \in D$ and y_{a_w, a_w} otherwise. Similarly $\psi_{\mathbf{d}}(y_{a_w, b_w})$ is y_{a_w, a_w} if $w \in D$ and y_{a_w, b_w} otherwise. We also set $\psi_{\mathbf{d}}(y_{b_w, a_w})$ to be y_{b_w, b_w} if $w \in D$ and y_{b_w, a_w} otherwise and we set $\psi_{\mathbf{d}}(y_{b_w, b_w})$ to be y_{b_w, a_w} if $w \in D$ and y_{b_w, b_w} otherwise. Finally for $v \in V$ we define $\psi_{\mathbf{d}}(z_{S,T}^v) := z_{S, T \Delta D}^v$ and have the property $\psi_{\mathbf{d}}(r(c)) = r(\Delta_{\mathbf{d}}(c))$ for all clauses $c \in F(\mathbf{M}(G), \mathbf{0})$.

Now, if the graph $MP(G)$ is asymmetric, the only solution of $\mathbf{M}(G)\mathbf{y} = \mathbf{0}$ is $\mathbf{y} = \mathbf{0}$. Then we can deduce $\mathbf{y}_k = 0$ from the equation system by combining rows. Applying Theorem 24, we get $F(\mathbf{M}(G), \mathbf{0}) \wedge V \vdash_{\mathcal{O}(m2^{L+1})}^{\text{SRC-II}} \xi_{k,0}$. Renaming variables yields $r(F(\mathbf{M}(G), \mathbf{0})) \wedge r(V) \vdash_{\mathcal{O}(m2^{L+1})}^{\text{SRC-II}} y_{a_\omega, a_\omega}$. As we have seen, $r(F(\mathbf{M}(G), \mathbf{0}))$ and $r(V)$ can be derived from F_1 and the symmetries are preserved; hence $F_1 \vdash_{\mathcal{O}(m2^{L+1})}^{\text{SRC-II}} y_{a_\omega, a_\omega}$.

Note that F_0 is obtained from F_1 simply by replacing y_{a_ω, a_ω} and y_{b_ω, b_ω} with 0. Hence, $F_0 \vdash_{\mathcal{O}(m2^{L+1})}^{\text{SRC-II}} \perp$. \blacktriangleleft

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, International Conference, TABLEAUX 2000, St Andrews, Scotland, UK, July 3-7, 2000, Proceedings*, volume 1847 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 2000. doi:10.1007/10722086_3.
- 2 Belaid Benhamou and Lakhdar Sais. Tractability through symmetries in propositional calculus. *J. Autom. Reasoning*, 12(1):89–102, 1994. doi:10.1007/BF00881844.
- 3 Joshua Blinkhorn and Olaf Beyersdorff. Proof complexity of QBF symmetry recomputation. In Mikolás Janota and Inês Lynce, editors, *Theory and Applications of Satisfiability Testing – SAT 2019 – 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings*, volume 11628 of *Lecture Notes in Computer Science*, pages 36–52. Springer, 2019. doi:10.1007/978-3-030-24258-9_3.
- 4 Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992. doi:10.1007/BF01305232.
- 5 Vasek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988. doi:10.1145/48014.48016.
- 6 Thierry Boy de la Tour and Stéphane Demri. On the complexity of extending ground resolution with symmetry rules. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995*, pages 289–297. Morgan Kaufmann, 1995. URL: <http://ijcai.org/Proceedings/95-1/Papers/038.pdf>.
- 7 Heidi E. Dixon, Matthew L. Ginsberg, David K. Hofer, Eugene M. Luks, and Andrew J. Parkes. Implementing a generalized version of resolution. In Deborah L. McGuinness and George Ferguson, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, pages 55–60. AAAI Press / The MIT Press, 2004. URL: <http://www.aaai.org/Library/AAAI/2004/aaai04-009.php>.
- 8 Uwe Egly. A first order resolution calculus with symmetries. In Andrei Voronkov, editor, *Logic Programming and Automated Reasoning, 4th International Conference, LPAR'93, St. Petersburg, Russia, July 13-20, 1993, Proceedings*, volume 698 of *Lecture Notes in Computer Science*, pages 110–121. Springer, 1993. doi:10.1007/3-540-56944-8_46.

- 9 Michael Frank and Michael Codish. Logic programming with graph automorphism: Integrating nauty with prolog (tool description). *TPLP*, 16(5-6):688–702, 2016. doi:10.1017/S1471068416000223.
- 10 Yuri Gurevich and Saharon Shelah. On finite rigid structures. *J. Symb. Log.*, 61(2):549–562, 1996. doi:10.2307/2275675.
- 11 Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985. doi:10.1016/0304-3975(85)90144-6.
- 12 Balakrishnan Krishnamurthy. Short proofs for tricky formulas. *Acta Informatica*, 22(3):253–275, August 1985. doi:10.1007/BF00265682.
- 13 Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014. doi:10.1016/j.jsc.2013.09.003.
- 14 Daniel Neuen and Pascal Schweitzer. An exponential lower bound for individualization-refinement algorithms for graph isomorphism. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 138–150. ACM, 2018. doi:10.1145/3188745.3188900.
- 15 Pascal Schweitzer and Constantin Seebach. Resolution with symmetry rule applied to linear equations. *CoRR*, abs/2101.05142, 2021. (Full version of the paper). arXiv:2101.05142.
- 16 Stefan Szeider. The complexity of resolution with generalized symmetry rules. *Theory Comput. Syst.*, 38(2):171–188, 2005. doi:10.1007/s00224-004-1192-0.
- 17 Jacobo Torán. On the resolution complexity of graph non-isomorphism. In Matti Jarvisalo and Allen Van Gelder, editors, *Theory and Applications of Satisfiability Testing - SAT 2013 - 16th International Conference, Helsinki, Finland, July 8-12, 2013. Proceedings*, volume 7962 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2013. doi:10.1007/978-3-642-39071-5_6.
- 18 G. S. Tseitin. *On the Complexity of Derivation in Propositional Calculus*, pages 466–483. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. doi:10.1007/978-3-642-81955-1_28.
- 19 Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, January 1987. doi:10.1145/7531.8928.
- 20 Alasdair Urquhart. The symmetry rule in propositional logic. *Discrete Applied Mathematics*, 96-97:177–193, 1999. doi:10.1016/S0166-218X(99)00039-6.