Satisfiability Problems for Finite Groups

Paweł M. Idziak
Jagiellonian University, Kraków, Poland

Piotr Kawałek
Jagiellonian University, Kraków, Poland

Jacek Krzaczkowski
Maria Curie-Sklodowska University, Lublin, Poland

Armin Weiß
Universität Stuttgart, FMI, Germany

Abstract

Over twenty years ago, Goldmann and Russell initiated the study of the complexity of the equation satisfiability problem (PolSat) and the NUDFA program satisfiability problem (ProgramSat) in finite groups. They showed that these problems are in P for nilpotent groups while they are NP-complete for non-solvable groups.

In this work we completely characterize finite groups for which the problem ProgramSat can be solved in randomized polynomial time under the assumptions of the Randomized Exponential Time Hypothesis and the Constant Degree Hypothesis. We also determine the complexity of PolSat for a wide class of finite groups. As a by-product, we obtain a classification for ListPolSat, a version of PolSat where each variable can be restricted to an arbitrary subset. Finally, we also prove unconditional algorithms for these problems in certain cases.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Complexity classes

Keywords and phrases Satisifiability, Solvable groups, ProgramSat, PolSat, Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.127

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Funding This research is partially supported by Polish NCN Grant # 2014/14/A/ST6/00138 and German DFG Grant WE 6835/1-2.

1 Introduction

Non-uniform deterministic finite automata (NUDFA) are a well-known concept introduced by Barrington [1], which proves its usefulness in describing important classes of languages defined by Boolean circuits such as NC^1 [2], ACC^0 and AC^0 [6]. Formally, a NUDFA (also called $\mathbf{G}$-program) over a group $\mathbf{G} = (G, \cdot)$ computing an n-ary function $f\colon \{0,1\}^n \longmapsto \{0,1\}$ consists of an l-ary polynomial $\mathbf{p}$ over $\mathbf{G}$ (i.e. the term over $\mathbf{G}$ with some variables replaced with constants from G), a set $S \subseteq G$ (the accepting set) and a sequence of l triples (instructions) of the form $\langle i_j, g_0^j, g_1^j\rangle$ where $1 \leqslant i_j \leqslant n$ and $g_0^j, g_1^j \in G$ for $0 \leqslant j \leqslant l$ (the number of the triple) such that $f(\overline{x}) = 1$ iff $\mathbf{p}(g^1_{x_{i_1}}, g^2_{x_{i_2}}, \ldots, g^l_{x_{i_l}}) \in S$. Note that this definition of NUDFA is not exactly the same as the one introduced in [1] but it is equivalent to the original one. In particular, there are obvious linear time transformations between the two program definitions. Moreover, this new definition does not change the class of languages recognized by NUDFA's over a fixed group.


The natural problem to decide whether the language recognized by a given NUDFA over a fixed group is non-empty (ProgramSat) was introduced in [14] and considered together with the problem of the existence of a solution to a given equation of polynomials over some fixed finite group (PolSat – here the input consists of polynomials $\mathbf{p}$ and $\mathbf{q}$ with variables $x_1, \ldots, x_n$, the question is whether there is some $\overline{a} \in G^n$ with $\mathbf{p}(\overline{a}) = \mathbf{q}(\overline{a})$). In the case of (finite) groups PolSat can be reduced to ProgramSat. In [14] Goldmann and Russell established a polynomial-time algorithm for ProgramSat in nilpotent groups and concluded that also PolSat for nilpotent groups is in P. On the other hand, their proof that PolSat for non-solvable groups is NP-complete implies NP-completeness of ProgramSat for such groups. This last fact was proved directly in [4]. These results reflect the ability of expressing all the functions AND_n by short polynomials of a particular group or by short NUDFA programs. It was known that over any fixed non-solvable group AND_n can be computed by NUDFAs (of size polynomial in n) [2], while over nilpotent groups NUDFAs are able to compute AND_n only of bounded arity n [5]. Note also that over solvable but non-nilpotent groups NUDFA programs are in fact able to compute all of the AND_n's [2], but sometimes these programs are of exponential length, namely $2^{\Omega(n)}$ [5].

It turned out that for a fixed group $\mathbf{G}$ the length of the shortest NUDFA program (over $\mathbf{G}$) computing AND_n plays a crucial role in classifying the computational complexity of ProgramSat($\mathbf{G}$). On the one hand, it was proved in [4] that, if the size of the shortest NUDFA program over a group $\mathbf{G}$ computing AND_n is $2^{\Omega(n)}$ (following [4] we say that $\mathbf{G}$ is AND-weak), then there exists a quasi-polynomial time algorithm solving ProgramSat($\mathbf{G}$). On the other hand, the same paper shows that, if $\mathbf{G}$ is AND-strong (i.e. for every n there exists an efficiently computable NUDFA over $\mathbf{G}$ of polynomial size computing AND_n), then ProgramSat for a wreath product $\mathbf{G} \wr \mathbb{Z}_k$ is NP-complete if $k \geqslant 4$. The problem is that determining the length of a shortest NUDFA program over a fixed group $\mathbf{G}$ computing AND_n is often highly non-trivial. Because of this difficulty with estimating the size of AND_n (occurring, in fact, in many models of computations), [5] introduced the so-called Constant Degree Hypothesis (CDH). In a circuit language it can be stated as follows: Consider a circuit of depth three where the input gates are connected to bounded fan-in AND gates followed by a layer of MOD_p gates and a MOD_q gate as output gate. Under CDH such a circuit needs size $2^{\Omega(n)}$ for computing AND_n (see [15, 16]). In this paper we will only use that CDH implies that groups of the form $\mathbf{G} = \mathbf{G}_p \rtimes \mathbf{N}$, where $\mathbf{G}_p$ is a p-group and $\mathbf{N}$ is a nilpotent group, are AND-weak (see Theorem 10 in [5]).

The difficulty of determining which groups are AND-weak is probably the reason why for almost 20 years after publishing [14] and [4] not too much progress has been made in characterizing the complexity of ProgramSat and PolSat for solvable but non-nilpotent groups. A number of results [19, 18, 11] were proved but all of them were restricted to showing polynomial time algorithms solving PolSat for some subclasses of groups with so-called Fitting length at most 2 (groups $\mathbf{G}$ having a nilpotent normal subgroup $\mathbf{N}$ such that $\mathbf{G}/\mathbf{N}$ is nilpotent), see [12] for a most comprehensive result. An important step towards the full classification of the computational complexity of ProgramSat for finite groups was made in 2020 when we proved in [20, 30, 23] that for every finite group of Fitting length at least 3 the problem PolSat (and in a consequence also ProgramSat) is not in P, unless the Exponential Time Hypothesis (ETH) fails. Shortly thereafter in [21] the first three authors gave examples of groups with Fitting length 2 and non-tractable PolSat (and ProgramSat), again, under the assumption of ETH.

This paper (among other things) gives a full characterization of finite groups for which ProgramSat is tractable in randomized polynomial time. Our classification also works for a related problem we call ListPolSat($\mathbf{G}$): given polynomials $\mathbf{p}$ and $\mathbf{q}$ with variables $x_1, \ldots, x_n$

and a list of subsets $A_1, \ldots, A_n \subseteq G$, decide whether there is $\overline{a} = (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n$ with $\mathbf{p}(\overline{a}) = \mathbf{q}(\overline{a})$. While we are not aware of any previous results on ListPolSat for finite algebras, it has been studied in the form of equations with rational constraints in [29] for word equations and in [10] for groups. Our classification of the complexity of these problems relies on both above-mentioned complexity assumptions (hypotheses): rETH and CDH.

▶ **Theorem 1.** *Under the assumption of rETH and CDH, the problems* ProgramSat($\mathbf{G}$) *and* ListPolSat($\mathbf{G}$) *for a finite group* $\mathbf{G}$ *are in* RP *if and only if there is a prime* p *and a normal* p*-subgroup* $\mathbf{G}_p$ *of* $\mathbf{G}$ *with* $\mathbf{G}/\mathbf{G}_p$ *being nilpotent.*

Note that our results partially confirm the intuition behind CDH, which was stated over 30 years ago: we show that in all cases in which there is a chance for AND-weakness, it is implied by CDH.

The proof of Theorem 1 is based on two main ideas. The first one is to show that, if some group $\mathbf{G}$ is AND-weak, then ProgramSat($\mathbf{G}$) satisfies what we call the *none-or-many property*: if a language recognized by a given program over $\mathbf{G}$ is not empty, then this language contains at least a polynomial fraction of all words (see Lemma 10). This gives us a randomized polynomial time algorithm solving ProgramSat($\mathbf{G}$) for groups which are AND-weak. Now, assuming CDH we immediately obtain the upper bound from Theorem 1. The second idea is to use polynomials witnessing the non-nilpotency of $\mathbf{G}$ to produce relatively short (of subexponential size) programs (and polynomials) expressing CNF-formulas. This, together with rETH ensures us that there is no polynomial time randomized algorithm solving ProgramSat for a group $\mathbf{G}$ which does not have a normal p-subgroup $\mathbf{G}_p$ with a nilpotent quotient $\mathbf{G}/\mathbf{G}_p$. In fact, we prove the main lemmas of this part of the proof (Lemma 14 and Theorem 16) in the much more general setting of solvable algebras (in a sense universal algebraic sense) from a so-called congruence permutable variety. We use two powerful universal algebraic tools: Tame Congruence Theory [17] and Commutator Theory [13] to prove Lemma 14, which tells us how to use non-nilpotency to produce polynomials over non-nilpotent algebras which imitate polynomials over finite fields. We conclude with a subexponential reduction from 3-CNF-SAT to ProgramSat which is based on the polynomials of small degree from [3] that describe symmetric periodic functions and were used to construct relatively small modular circuits for AND.

The situation for PolSat is more involved: indeed, we are far from getting full classification. The main reason is that in this case we cannot restrict the arguments of a polynomial to certain values. Hence, we need much more control to be able to use the polynomials from [3] in the proof of Theorem 16. In order to state Theorem 16 in the group case, we write $C_{\mathbf{G}}(\mathbf{A}) = \{\, g \in G \mid ga = ag \text{ for all } a \in A \,\}$ for the centralizer of a normal subgroup $\mathbf{A}$ of $\mathbf{G}$:

▶ **Theorem 2.** *Let* $\mathbf{G}$ *be a finite solvable group with two minimal non-trivial normal subgroups* $\mathbf{A}$ *and* $\mathbf{B}$ *such that* $|A|$ *and* $|B|$ *are coprime and* $C_{\mathbf{G}}(\mathbf{A}) \cdot C_{\mathbf{G}}(\mathbf{B}) \neq G$*. Then the problem* PolSat($\mathbf{G}$) *is not in* RP *under rETH.*

▶ **Corollary 3.** *If a finite group* $\widetilde{\mathbf{G}}$ *has a quotient as in Theorem 2, then* PolSat($\widetilde{\mathbf{G}}$) *is not in* RP *under rETH.*

Notice that the conditions for hardness in Theorem 1 and Theorem 2 are quite similar. Indeed, if we are not in the RP-case of Theorem 1, then there are two different primes in $|[\mathbf{G}, \mathbf{G}]|$ witnessed by two normal subgroups $\mathbf{A}$ and $\mathbf{B}$ of $\mathbf{G}$ (but contained in $[\mathbf{G}, \mathbf{G}]$) of coprime order with $C_{\mathbf{G}}(\mathbf{A}) \neq G \neq C_{\mathbf{G}}(\mathbf{B})$ as opposed to $C_{\mathbf{G}}(\mathbf{A}) \cdot C_{\mathbf{G}}(\mathbf{B}) \neq G$ in Theorem 2. This subtle difference prevents us from giving a classification as Theorem 1 for the case of PolSat (for details, we refer to the proof of Theorem 16). Moreover, Theorem 4 shows that this is not due to our ignorance but there are indeed groups for which PolSat is in RP and ProgramSat is not (under rETH).

▶ **Theorem 4.** *Let* $\mathbf{G}$ *be the semidirect product* $\mathbf{N} \rtimes \mathbf{H}$ *of nilpotent groups* $\mathbf{N}$ *and* $\mathbf{H}$ *and let* $|G|$ *have at most two prime factors. Then* PolSat($\mathbf{G}$) *is in* RP *under CDH. Moreover, if* $\mathbf{H}$ *is abelian,* PolSat($\mathbf{G}$) *is in* RP *unconditionally.*

Note here that the last sentence of Theorem 4 gives an unconditional upper bound for the complexity of PolSat. This together with Theorem 1 enables us to classify the computational complexity of PolSat for dihedral groups (i.e the groups of symmetries of regular polygons, where $\mathbf{D}_m = \mathbb{Z}_m \rtimes \mathbb{Z}_2$ denotes the symmetry group of the m-gon).

▶ **Corollary 5** (Classification of dihedral groups)**.**

(i) *If* $m = 2^\alpha p^\beta$ *for* $\alpha, \beta \in \mathbb{N}$ *and an odd prime* p*, then* PolSat($\mathbf{D}_m$) *is in* RP*.*

(ii) *Otherwise* PolSat($\mathbf{D}_m$) *cannot be solved in* RP *under rETH (resp.* P *under ETH).*

Furthermore, based on Theorem 2 we obtain (under the assumption of CDH and rETH) a classification of PolSat for wreath products of nilpotent groups.

▶ **Corollary 6.** *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be nilpotent groups. Under CDH and rETH,* PolSat($\mathbf{G} \wr \mathbf{H}$) *is in* RP *if an only if* $\mathbf{G}$ *is a* p*-group or* $|G| \cdot |H|$ *has at most two prime divisors.*

In [12] a question was asked about the computational complexity of PolSat for several examples of groups. Among them were four groups of order 24 (complexity of PolSat for groups of smaller order was known previously). In this paper we determine the computational complexity of PolSat for three of these groups: $\mathbf{D}_{12}$, $(\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_3) \rtimes \mathbb{Z}_2$ and $\mathbb{Z}_3 \rtimes \mathbf{Q}$, where $\mathbf{Q}$ is the quaternion group. It turns out that PolSat for these groups is in RP (see Corollary 5 and Example 21). On the other hand, $\mathbf{S}_4$, the fourth of the groups mentioned above, was shown in [20, 23] to have non-tractable PolSat problem (assuming ETH).

2 Preliminaries

We use $[m \,..\, n]$ to denote the interval of integers $\{m, \ldots, n\}$, the difference of sets is denoted by $A - B$. We use standard notation from complexity theory, which can be found in any textbook on complexity, e.g. [27]. In particular, we write RP for randomized polynomial time (with one-sided error).

ETH and rETH. The Exponential Time Hypothesis (ETH) and its randomized version rETH (see e.g. [9]) is the conjecture that there is some $\delta > 0$ such that every (randomized) algorithm for 3-CNF-SAT needs time $\Omega(2^{\delta n})$ in the worst case where n is the number of variables of the given 3-CNF-SAT instance. By the Sparsification Lemma [25, Thm. 1] this is equivalent to the existence of some $\epsilon > 0$ such that every algorithm for 3-CNF-SAT needs (randomized) time $\Omega(2^{\epsilon(m+n)})$ in the worst case where m is the number of clauses of the given 3-CNF-SAT instance (see also [8, Thm. 14.4]). In particular, under ETH/rETH there is no (randomized) algorithm for 3-CNF-SAT running in time $2^{o(n+m)}$. Here we only consider one-sided errors; clearly our results also remain true when understanding rETH with two-sided error.

Group Theory. Throughout, we only consider finite groups $\mathbf{G}$ with underlying set (universe) G. We follow the notation of [28]. For groups $\mathbf{G}$ and $\mathbf{H}$ we write $\mathbf{H} \leqslant \mathbf{G}$ if $\mathbf{H}$ is a subgroup of $\mathbf{G}$ and $\mathbf{H} < \mathbf{G}$ if $\mathbf{H}$ is a proper subgroup of $\mathbf{G}$. For a subset $X \subseteq G$ we write $\langle X \rangle$ for the subgroup generated by X. We write $[x, y] = x^{-1}y^{-1}xy$ for the *commutator*. The commutator of subgroups $\mathbf{X}, \mathbf{Y} \leqslant \mathbf{G}$ is defined by $[\mathbf{X}, \mathbf{Y}] = \langle\, [x, y] \mid x \in X, y \in Y \,\rangle$.

Let $\mathbf{N} \leqslant \mathbf{H}$ be a normal subgroup of $\mathbf{G}$. We define the *centralizer* of $\mathbf{H}$ modulo $\mathbf{N}$ as $C_{\mathbf{G}}(\mathbf{H}/\mathbf{N}) = \{ g \in G \mid [g,h] \in N \text{ for all } h \in H \}$. The center of a group is $Z(\mathbf{G}) = C_{\mathbf{G}}(\mathbf{G})$. Nilpotent groups can be defined inductively: $\mathbf{G}$ is nilpotent of class 1 if it is abelian; $\mathbf{G}$ is *nilpotent* of class c if $\mathbf{G}/Z(\mathbf{G})$ is nilpotent of class $c-1$. A group is called a *p-group* (for p prime) if its order is p^k for some $k \in \mathbb{N}$. It is well-known that a finite group is nilpotent iff it is a direct product of p-groups (see e.g. 5.1.3 and 5.2.4 in [28]). We will use this fact without further reference. If $\mathbf{G} = \mathbf{NH}$ where $\mathbf{N}$ is a normal subgroup of $\mathbf{G}$ and $\mathbf{H}$ a subgroup with $N \cap H = \{1\}$, $\mathbf{G}$ is called a *semidirect product* and we write $\mathbf{G} = \mathbf{N} \rtimes \mathbf{H}$. Notice that $\mathbf{G}/\mathbf{N} = \mathbf{H}$ in this case. Elements of $\mathbf{N} \rtimes \mathbf{H}$ can be viewed as pairs (n,h) with $n \in N$ and $h \in H$ with the multiplication rule $(n_1,h_1)(n_2,h_2) = (n_1\, {}^{h_1}n_2, h_1h_2)$ where ${}^{h_1}n_2 = h_1n_2h_1^{-1}$ is an action (via automorphisms) of $\mathbf{H}$ on $\mathbf{N}$. We will use the following classical result (see e.g. [28, 9.1.2]):

▶ **Fact 7** (Schur-Zassenhaus Theorem). *If* $\mathbf{G}$ *is a group with a normal subgroup* $\mathbf{N}$ *and* $|N|$ *and* $|\mathbf{G}/\mathbf{N}|$ *are coprime, then* $\mathbf{G} = \mathbf{N} \rtimes \mathbf{G}/\mathbf{N}$.

To define the wreath product $\mathbf{G} \wr \mathbf{H}$ of groups $\mathbf{G}$ and $\mathbf{H}$, we start with the direct power $\mathbf{G}^H = \{ f : H \to G \}$. Now $\mathbf{H}$ has a natural left action on $\mathbf{G}^H$ given by $({}^hf)(y) = f(h^{-1}y)$ for $f \in \mathbf{G}^H$, $h \in H$ and $y \in H$. Now the *wreath product* $\mathbf{G} \wr \mathbf{H}$ is defined to be the corresponding semidirect product $\mathbf{G}^H \rtimes \mathbf{H}$.

Algebra. A finite algebra $\mathbf{A}$ is a finite universe A together with a finite number of k_i-ary fundamental operations $f_i : A^{k_i} \to A$ for $i \in I$. A *term* is a composition of fundamental operations and a *polynomial* is a term with some variables replaced by constants from A; we write $\mathrm{Pol}(\mathbf{A})$ for the set of polynomials over $\mathbf{A}$. An equivalence relation which preserves all operations of $\mathbf{A}$ is called a *congruence* (more precisely, an equivalence relation $\alpha \subseteq A \times A$ is a congruence iff for every fundamental operation f, say r-ary, of $\mathbf{A}$ and $(a_1,b_1), \ldots, (a_r,b_r) \in \alpha$ we have $(f(\overline{a}), f(\overline{b})) \in \alpha$). We usually write $a \overset{\alpha}{\equiv} b$ to express that $(a,b) \in \alpha$. For a congruence relation α on $\mathbf{A}$ the congruence class containing an element a is denoted by a/α and A/α is the set of all congruence classes of α. The congruences of an algebra $\mathbf{A}$, when ordered by inclusion (denoted by $\alpha \leqslant \beta$), form the complete lattice with the top element $1_{\mathbf{A}} = A \times A$ and the bottom element $0_{\mathbf{A}}$ consisting of all pairs (a,a) for $a \in A$. We frequently omit the subscripts in $0_{\mathbf{A}}$ and $1_{\mathbf{A}}$. Given two congruences β and γ, we write $\beta \vee \gamma$ for the *join*, which is the smallest congruence containing both β and γ. A congruence α is called *join-irreducible* iff, whenever $\alpha = \beta \vee \gamma$ for congruences β and γ, already $\alpha \in \{\beta, \gamma\}$. Note that a join-irreducible α has a unique congruence α_- such that $\alpha_- < \alpha$ and there is no β with $\alpha_- < \beta < \alpha$.

A normal subgroup $\mathbf{N}$ of a group $\mathbf{G}$ defines a congruence $\alpha_{\mathbf{N}} = \{ (a,b) \in G^2 \mid a^{-1}b \in N \}$. As $\mathbf{G}/\mathbf{N} = \mathbf{G}/\alpha_{\mathbf{N}}$, we do not distinguish between congruences and normal subgroups.

The commutator $[\mathbf{H}, \mathbf{K}]$ of normal subgroups of a group has been generalized as an operation on congruences for arbitrary algebras (see [13]) and then used to extend notions of solvability and nilpotency onto such algebras. For a precise definition of the commutator $[\alpha, \beta]$ of congruences α and β we refer to [13] and simply note that for groups it is the usual commutator. We say that a congruence α *centralizes* β modulo γ iff $[\alpha, \beta] \leqslant \gamma$. The biggest α which centralizes β modulo γ is denoted by $(\gamma : \beta)$ and called the *centralizer* of β modulo γ. In the group case this agrees exactly with the usual definition of the centralizer.

Another important concept of universal algebra derived from group theory is a *Malcev term*: it is a term $\mathbf{d}$ satisfying $\mathbf{d}(y,x,x) = \mathbf{d}(x,x,y) = y$. The existence of such a term implies many nice properties of an algebra (e.g. connected with the behaviour of commutators and the congruence lattice). In the group case, the term $\mathbf{d}(x,y,z) = xy^{-1}z$ is an example of a Malcev term. An algebra with a Malcev term is called a *Malcev algebra*.

Some of our proofs use advanced tools of universal algebra: the Commutator Theory and the Tame Congruence Theory as presented in the books [13] and [17]. The reader may find a not too long introduction to the needed notions and facts from these theories in [24].

Satisfiability problems. For the definition of a $\mathbf{G}$-program (aka. NUDFA over $\mathbf{G}$) and ProgramSat we refer to the introduction. Usually, we denote both polynomials and $\mathbf{G}$-programs by $\mathbf{p}$ and, for a simpler notation, also use $\mathbf{p}$ to denote the function $f : \{0,1\}^n \to \{0,1\}$ computed by the $\mathbf{G}$-program. For a unified treatment of ListPolSat and ProgramSat, we further define 2-ListPolSat which is like ListPolSat but the A_i all may contain only two elements. Notice that in groups, as we can multiply by inverses, we can assume that the input for these problems consists of only one polynomial (plus the lists restricting variables). Thus, in the group setting we call $\overline{a} \in G^n$ a solution to an instance $\mathbf{p}(\overline{x})$ for PolSat($\mathbf{G}$) if $\mathbf{p}(\overline{a}) = 1$. Likewise, we call some $\overline{a} \in \{0,1\}^n$ a solution for a $\mathbf{G}$-program $\mathbf{p}$ if $\mathbf{p}(\overline{a}) = 1$ (recall that $\mathbf{p}$ computes a function $\{0,1\}^n \to \{0,1\}$). It is obvious that we can treat 2-ListPolSat as a "subproblem" of ProgramSat in which NUDFA's programs have the following property: for every two instructions (i_j, g_0^j, g_1^j), (i_k, g_0^k, g_1^k) of the program, if $i_j = i_k$, then $g_0^j = g_0^k$ and $g_1^j = g_1^k$. Also note that even for general algebras PolSat($\mathbf{A}$) is the special case of ListPolSat($\mathbf{A}$) where all the A_i are A. These observations and some other well-known results are summarized as follows (see also [14, 4]):

▶ **Lemma 8.** *Let* $\mathbf{G}$ *be a group. Then*

- *If* $\mathbf{H} = \mathbf{G}/\mathbf{N}$, *then* PolSat($\mathbf{H}$) $\leqslant_{dtt}$ PolSat($\mathbf{G}$).
- *If* $\mathbf{H} \leqslant \mathbf{G}$ *or* $\mathbf{H} = \mathbf{G}/\mathbf{N}$, *then* ProgramSat($\mathbf{H}$) $\leqslant_m$ ProgramSat($\mathbf{G}$) *and* ListPolSat($\mathbf{H}$) $\leqslant_m$ ListPolSat($\mathbf{G}$).
- PolSat($\mathbf{G}$) $\leqslant_m$ 2-ListPolSat($\mathbf{G}$)
- 2-ListPolSat($\mathbf{G}$) $\leqslant_m$ ProgramSat($\mathbf{G}$) *and* 2-ListPolSat($\mathbf{G}$) $\leqslant_m$ ListPolSat($\mathbf{G}$).

Here $A \leqslant_m B$ denotes a polynomial-time many-one reduction and $A \leqslant_{dtt} B$ denotes a polynomial-time disjunctive truth-table reduction: one instance x for A is reduced to several instances $y_1, \ldots, y_k$ of B such that $x \in A$ if and only if there is some i with $y_i \in B$.

3 CDH and the Many-Solutions Property

We wish to treat NUDFA programs and polynomials (with variables restricted to lists) in a unified setting. For this we have to face the problem that programs are defined on Boolean domains whereas the domain of a variable of a polynomial is (a subset of) a group. So with an instance of ListPolSat / ProgramSat we associate an indicator function $f : \prod_{i=1}^n A_i \to \{0,1\}$ with $|A_i| \geqslant 2$ ($A_i \subseteq G$ resp. $A_i = \{0,1\}$) such that $f(\overline{x}) = 1$ iff $\overline{x}$ is a solution to the corresponding NUDFA program/polynomial equation with lists. In these cases we measure the size of f (denoted by size(f)) by the size of the smallest program/polynomial representing it (in the second case together with the sizes of lists). More precisely, in the definition of the size we take the smallest possible representation for f.

Note that independently of the model the function f was created in, if we replace some variables by constants, or restrict some of the A_i's from the domain to smaller sets $A_i' \subseteq A_i$ with $|A_i'| \geqslant 2$, we still obtain an indicator function (of possibly smaller arity) for another ListPolSat / ProgramSat instance. Moreover, the size of a function after such operations does not increase. This is the crucial property we use in Proposition 9 below. We will use the following notation:

- $f[x_J/\overline{a}]$ for a function obtained from f by substituting x_j by a_j for all $j \in J$,
- $f_{|\mathcal{B}}$ for a function obtained from f by restricting the domain of f, that is $\prod_{i=1}^n A_i$, to $\mathcal{B} = \prod_{i=1}^n B_j$ (for $B_j \subseteq A_j$),
- $f[x_i/a]$ for the substitution of one variable.

Also we combine these notations, e.g. by writing $f[x_I/\overline{b}, x_i/a]_{|\mathcal{B}}$.

Our aim is to bound the size of a general function using previous knowledge about the size for describing the AND function. In order to do so, we call $f : \prod_{i=1}^n A_i \to \{0,1\}$ with $|A_i| \geqslant 2$ for all i an n-ary *spike* if there is some $\overline{a} \in \prod_{i=1}^n A_i$ with $f(\overline{a}) = 1$ and for all $\overline{x} \neq \overline{a}$ we have $f(\overline{x}) = 0$. Let $\gamma : \mathbb{N} \to \mathbb{N}$ be the function $\gamma(n) = \min\{\,\mathrm{size}(f) \mid f \text{ is an } n\text{-ary spike}\,\}$. Notice that γ is monotone. Thus, its inverse $\gamma^{-1}(m) = \max\{\,n \in \mathbb{N} \mid \gamma(n) \leqslant m\,\}$ as a function $\mathbb{N} \to \mathbb{N}$ is well-defined and also monotone.

Sometimes we will also write $\gamma_{\mathrm{Prog},\mathbf{G}}$ or $\gamma_{\mathrm{Pol},\mathbf{G}}$ when we want to specify the model.

▶ **Proposition 9.** *Let f be an n-ary indicator function with domain $\mathcal{A} = \prod_{i=1}^n A_i$. Then either f is constant zero or $\left|f^{-1}(1)\right| / |\mathcal{A}| \geqslant 1/|\mathcal{A}|^{\gamma^{-1}(\mathrm{size}(f))}$. In particular, if $\gamma(n) \in 2^{\Omega(n)}$, then $\left|f^{-1}(1)\right| / |\mathcal{A}| \geqslant 1/\mathrm{size}(f)^{\mathcal{O}(1)}$.*

Proof. Our proof relies on the same idea as [21, Theorem 6.1]; however, we have to overcome the difficulty to deal with non-boolean domains. If f is constantly 1 the statement holds; otherwise f is non-nonstant. The idea is to successively substitute constants for variables while not increasing the density of $f^{-1}(1)$ in the full domain: If there is some $i \in [1\,..\,n]$ such that for all $a \in A_i$ the function $f[x_i/a]$ is not constant 0, we choose $b \in A_i$ such that $\left|f[x_i/b]^{-1}(1)\right|$ is minimal among all $\left|f[x_i/a]^{-1}(1)\right|$ for $a \in A_i$. Observe that $f[x_i/b]$ is not a constant function and $\left|f[x_i/b]^{-1}(1)\right| \leqslant \left|f^{-1}(1)\right| / |A_i|$. Now, we proceed by induction eventually obtaining some $I \subseteq [1\,..\,n]$ and $\overline{b} \in \prod_{i\in I} A_i$ such that $f[x_I/\overline{b}]$ is not constant, $1 \leqslant \left|f[x_I/\overline{b}]^{-1}(1)\right| \cdot \prod_{i\in I} |A_i| \leqslant \left|f^{-1}(1)\right|$, and for every $i \in [1\,..\,n] - I$ there is some $a \in A_i$ such that $f[x_I/\overline{b}, x_i/a]$ is constant zero. Now, we can restrict all remaining variables to two-element sets and we obtain a $(n - |I|)$-ary spike. As during this process the size of f does not increase, we obtain:

$$\frac{\left|f^{-1}(1)\right|}{|\mathcal{A}|} \geqslant \frac{\prod_{i\in I} |A_i|}{\prod_{i=1}^n |A_i|} = \frac{1}{\prod_{[1..n]-I} |A_i|} \geqslant \frac{1}{|\mathcal{A}|^{n-|I|}} \geqslant \frac{1}{|\mathcal{A}|^{\gamma^{-1}(\mathrm{size}(f))}}.$$ ◀

We say that ProgramSat($\mathbf{G}$), ListPolSat($\mathbf{G}$) or PolSat($\mathbf{G}$) has the *none-or-many property* if for any instance $\mathbf{p}$ of length m either $\mathbf{p}$ has no solution or a random assignment will be a solution with $1/m^{\mathcal{O}(1)}$ probability (recall that we call $\overline{a} \in G^n$ a solution to an instance $\mathbf{p}(\overline{x})$ for PolSat($\mathbf{G}$) if $\mathbf{p}(\overline{a}) = 1$ and similarly for ListPolSat($\mathbf{G}$) and ProgramSat).

▶ **Lemma 10.** *Let G be a group with $\gamma_{\mathrm{Prog},G}(n) \in 2^{\Omega(n)}$. Then, the problems* ProgramSat($\mathbf{G}$)*,* ListPolSat($\mathbf{G}$) *and* PolSat($\mathbf{G}$) *have the none-or-many property.*

In particular, ProgramSat($\mathbf{G}$)*,* ListPolSat($\mathbf{G}$) *and* PolSat($\mathbf{G}$) *are in* RP*.*

For the proof of Lemma 10 notice that a G-program for a spike is never longer than a polynomial for a spike. Therefore, the requirement $\gamma_{\mathrm{Prog},G}(n) \in 2^{\Omega(n)}$ also can be used for ListPolSat. The result for PolSat then follows by Lemma 8. For some polynomial or $\mathbf{G}$-program $\mathbf{p}$ we use Proposition 9 to get a bound $\left|\mathbf{p}^{-1}(1)\right| / |G^n| \geqslant 1/|G|^{\gamma^{-1}(|\mathbf{p}|)} \in 1/|G|^{\mathcal{O}(\log|\mathbf{p}|)} \subseteq 1/|\mathbf{p}|^{\mathcal{O}(1)}$.

▶ Remark 11. Notice that, if $\mathbf{G}$ is a group with $\gamma_{\mathrm{Prog},\mathbf{G}}(n) \in 2^{\Omega(n)}$, then by [4, Theorem 2] ProgramSat($\mathbf{G}$), ListPolSat($\mathbf{G}$) and PolSat($\mathbf{G}$) also can be solved in deterministic quasi-polynomial time (notice that [4, Theorem 2] technically is not for ListPolSat($\mathbf{G}$) but the proof can easily be adapted).

CDH-based conditional algorithms. Let us recall the consequence of the constant degree hypothesis (CDH) which has been proved in Theorem 10 of [5]:

(A) Let $\mathbf{G}_p$ be a p-group, $\mathbf{N}$ a nilpotent group and $\mathbf{G} = \mathbf{G}_p \rtimes \mathbf{N}$. If CDH is true, then every $\mathbf{G}$-program computing the n-ary AND function has length $2^{\Omega(n)}$.

Note that [5] gives a proof of CDH in the case that $\mathbf{N}$ is abelian, see (C) in Section 6 below. As an immediate consequence of Lemma 10 and (A), we obtain our next result.

▶ **Corollary 12.** *Let $\mathbf{G}_p$ be a p-group, $\mathbf{N}$ a nilpotent group and $\mathbf{G} = \mathbf{G}_p \rtimes \mathbf{N}$. If CDH is true, then* ProgramSat($\mathbf{G}$), ListPolSat($\mathbf{G}$) *or* PolSat($\mathbf{G}$) *have the none-or-many property.*

In particular, if CDH is true, then ProgramSat($\mathbf{G}$), ListPolSat($\mathbf{G}$) *and* PolSat($\mathbf{G}$) *are in* RP.

4 Lower Bounds

In this section we will prove our hardness conditions. Instead of ProgramSat we will prove intractability of 2-ListPolSat. For that we need go deeply into the local structure of finite algebras (called sometimes as Tame Congruence Theory) as described in [17]. In our setting, i.e. solvable Malcev algebras, this local structure, (relative to a pair of congruences $\alpha \prec \beta$, i.e. $\alpha < \beta$ with no congruence γ in between: $\alpha < \gamma < \beta$) reduces to a vector space over a finite field. The prime that is the characteristic of this field, is also called the *characteristic* of the pair (α, β). For a join-irreducible congruence γ we define its *characteristic* to be the characteristic of the pair (γ_-, γ).

Different characteristics of two join-irreducible congruences is one of two important ingredients we use in the proof of Theorem 16. The second one is what we call congruence collaboration: Two congruences α and β of an algebra $\mathbf{A}$ are called *collaborating* if there are 3 different elements $a, c, b \in A$ satisfying $a \overset{\alpha-\beta}{\frown} c \overset{\beta}{\frown} b$ or $a \overset{\alpha}{\frown} c \overset{\beta-\alpha}{\frown} b$, where $x \overset{R}{\frown} y$ stands for $(x, y) \in R$. Notice that in a group case any two distinct non-trivial normal subgroups are collaborating. We will also use following easy observation.

▶ **Lemma 13.** *For two congruences $\varphi \neq 1_{\mathbf{A}} \neq \psi$ of an algebra $\mathbf{A}$ we have $\varphi \cup \psi \neq 1_{\mathbf{A}}$.*

Proof. Suppose $\varphi \cup \psi = 1_{\mathbf{A}} \neq \varphi$. To see that then $(a, b) \in \psi$ for all $a, b \in A$, note first that $(a, b) \in \psi$ whenever $(a, b) \notin \varphi$. Now let $(a, b) \in \varphi$. Obviously $\varphi \neq 1_{\mathbf{A}}$ gives an element $c \in A - a/\varphi = A - b/\varphi$. But the previous case puts both (c, a) and (c, b) into ψ, so that $(a, b) \in \psi$ as claimed. ◀

Note that Lemma 13 cannot be extended to more than 2 congruences: in a finitely dimensional vector space the union of all 1-dimensional subspaces covers the entire space.

In the next Lemma we use the polynomial which witnesses the lack of a centralization to produce a polynomial of the algebra which imitates a given polynomial over the field $GF(p)$. A similar argument was used in [22].

▶ **Lemma 14.** *Let $\mathbf{A}$ be a finite solvable Malcev algebra. Moreover, let γ be a join-irreducible congruence of characteristic p such that $(\gamma_- : \gamma) \neq 1$. Then for:*

- *every pair $(e, a) \in \gamma$ of different elements,*
- *every subset $\top \subseteq A$ that is a union of $(\gamma_- : \gamma)$-cosets,*
- *and every n-ary polynomial $w(\overline{x})$ of degree s over the field $GF(p)$ that sends the set $\{0, 1\}^n$ to $\{0, 1\}$*

there is an n-ary polynomial $[w]_{\gamma, \top, a, e}$ of the algebra $\mathbf{A}$ such that

- *both the size of* $[w]_{\gamma,\top,a,e}$ *and the time needed to compute it are bounded by* $2^{O(s\cdot\log n)}$,
- *for any tuple* $\overline{x}\in A^n$ *we have*

$$[w]_{\gamma,\top,a,e}(\overline{x}) = \begin{cases} a, & \text{if } w(b_\top(x_1),\dots,b_\top(x_n)) = 1,\\ e, & \text{if } w(b_\top(x_1),\dots,b_\top(x_n)) = 0,\end{cases}$$

where $b_\top : A \longrightarrow \{0,1\}$ *is defined by* $b_\top(x) = 1$ *iff* $x\in\top$.

Proof. First put $\gamma^\star = (\gamma_- : \gamma)$ and let $\{d_0, d_1, \dots, d_r\}$ be a transversal of the quotient $\mathbf{A}/\gamma^\star$. Moreover, pick a (γ_-,γ)-minimal set V and two elements $(e',a')\in\gamma|_V-\gamma_-$. By N denote the trace of V containing both e' and a'. Note that $(\mathbf{A}|_N)/\gamma_-$ is polynomially equivalent to a one-dimensional vector space over a field of characteristic p. In fact we will use only its additive structure $((\mathbf{A}|_N)/\gamma_-,+)$ which is isomorphic to a certain power of $(\mathbb{Z}_p,+)$. We can assume that e'/γ_- is the neutral element of this group and that addition in this group is realized (modulo γ_-) by the polynomial $x+y=\mathbf{d}(x,e',y)$, where $\mathbf{d}$ is a Malcev term for $\mathbf{A}$. Note that, despite the fact that the addition defined above is guaranteed to behave nicely only modulo γ_-, the properties of the Malcev term give us that $e'+e' = \mathbf{d}(e',e',e') = e'$. Since we will be summing the large amount of summands, to keep the sum relatively short (i.e. of the length which is polynomial in the sum of the summands' lengths) we will compose the addition in a balanced binary way, i.e. the tree of a polynomial realizing the sum of m summands is supposed to be the complete binary tree with m leaves – we will point out when this is necessary. Now we use the argument for the second half of Lemma 3.1 in [22] to show that for any pair $(c,d)\notin\gamma^\star$ there is a binary polynomial $\mathbf{s}_{cd}(x,y)$ of $\mathbf{A}$, satisfying

$$\begin{aligned} \mathbf{s}_{cd}(e',y) &= e', && \text{for all } y\in A,\\ \mathbf{s}_{cd}(a',c) &\overset{\gamma_-}{\equiv} e', &&\\ \mathbf{s}_{cd}(a',d) &= a'. && \end{aligned} \tag{1}$$

Because $(e',a')\notin\gamma_-$ and γ is join-irreducible, we know that $\gamma=\Theta(e',a')$. Now if $(c,d)\notin\gamma^\star$ then $[\Theta(c,d),\Theta(e',a')]\not\leqslant\gamma_-$ so that Exercise 6.6 in [13] supplies us with a binary polynomial $\mathbf{s}(x,y)$ of $\mathbf{A}$ such that $\mathbf{s}(e',c)\overset{\gamma_-}{\equiv}\mathbf{s}(a',c)$ but $\mathbf{s}(e',d)\overset{\gamma_-}{\not\equiv}\mathbf{s}(a',d)$. The second property gives $\Theta(\mathbf{s}(e',d),\mathbf{s}(a',d))=\gamma\ni(e',a')$ and therefore there is a unary polynomial $\mathbf{p}$ of $\mathbf{A}$ that takes the pair $(\mathbf{s}(e',d),\mathbf{s}(a',d))$ to (e',a') (see e.g. Lemma 3.2 in [24]). Now it should be easy to check that for the polynomial $\mathbf{s}_{cd}(x,y)=\mathbf{d}(\mathbf{ps}(x,y),\mathbf{ps}(e',y),e')$ we have

$$\begin{aligned} \mathbf{s}_{cd}(e',y) &= \mathbf{d}(\mathbf{ps}(e',y),\mathbf{ps}(e',y),e') &&= e',\\ \mathbf{s}_{cd}(a',c) &= \mathbf{d}(\mathbf{ps}(a',c),\mathbf{ps}(e',c),e') &&\overset{\gamma_-}{\equiv} \mathbf{d}(\mathbf{ps}(e',c),\mathbf{ps}(e',c),e') = e',\\ \mathbf{s}_{cd}(a',d) &= \mathbf{d}(\mathbf{ps}(a',d),\mathbf{ps}(e',d),e') &&= \mathbf{d}(a',e',e') = a', \end{aligned}$$

as claimed in (1).

Using the fact that $[\gamma,\gamma^\star]\leqslant\gamma_-$ we can keep conditions (1) modulo γ_- by varying the second variable modulo $\gamma^\star$:

$$\begin{aligned} \mathbf{s}_{cd}(e',y) &= e', && \text{for each } y\in A,\\ \mathbf{s}_{cd}(a',y) &\overset{\gamma_-}{\equiv} e', && \text{for each } y\in c/\gamma^\star,\\ \mathbf{s}_{cd}(a',y) &\overset{\gamma_-}{\equiv} a', && \text{for each } y\in d/\gamma^\star. \end{aligned} \tag{2}$$

Now we want c and d to range over our transversal of $\mathbf{A}/\gamma^\star$ so that for $i\neq j$ we put $\mathbf{s}_{ij}(x,y)=\mathbf{e}_V\mathbf{s}_{d_id_j}(\mathbf{e}_V(x),y)$, where $\mathbf{e}_V$ is the unary idempotent polynomial of $\mathbf{A}$ with range V. Obviously $\mathbf{s}_{ij}$ satisfies all the properties of $\mathbf{s}_{d_id_j}$ listed in (2), but the polynomial $\mathbf{s}_{ij}$ has its range contained in V and for any fixed $y\in A$ the mapping $V\ni v\longmapsto\mathbf{s}_{ij}(v,y)\in V$ is either a permutation of the (γ_-,γ)-minimal set V or collapses $\gamma|_V$ to γ_-, i.e. it is constant modulo γ_- on $\gamma|_V$-classes.

Hence, as for $v \in N$ and $y \in d_i/\gamma_-$ the map $v \longmapsto \mathbf{s}_{ij}(v,y)$ is not a permutation, we have $\mathbf{s}_{i,j}(v,y) \overset{\gamma_-}{\equiv} \mathbf{s}_{i,j}(a',y) \overset{\gamma_-}{\equiv} e' = \mathbf{s}_{i,j}(e',y)$, which allows us to replace the second line in (2) by:

$$\mathbf{s}_{ij}(v,y) \overset{\gamma_-}{\equiv} e', \quad \text{for each } v \in N \text{ and } y \in d_i/\gamma^\star. \tag{3}$$

Now for each $j = 0,\dots,r$ put $\mathbf{s}_j(v,y) = \mathbf{s}_{i_1 j}(\dots \mathbf{s}_{i_{r-1} j}(\mathbf{s}_{i_r j}(v,y),y)\dots,y)$, with $\{i_1,\dots,i_r\} = \{0,1,\dots,r\} - \{j\}$. Obviously $\mathbf{s}_j$ has its range contained in V. We will show that

$$\begin{array}{lcll} \mathbf{s}_j(e',y) & = & e', & \text{for each } \ y \in A, \\ \mathbf{s}_j(v,y) & \overset{\gamma_-}{\equiv} & e', & \text{for each } \ v \in N \text{ and } \ y \in A - d_j/\gamma^\star, \\ \mathbf{s}_j(a',y) & \overset{\gamma_-}{\equiv} & a', & \text{for each } \ y \in d_j/\gamma^\star. \end{array} \tag{4}$$

Indeed, the first and the last item follow directly from the definition of $\mathbf{s}_j$. For the middle one, note that for $v \in N$, $y \in d_{i_\ell}/\gamma^\star$, and $v' = \mathbf{s}_{i_{\ell+1} j}(\dots \mathbf{s}_{i_{r-1} j}(\mathbf{s}_{i_r j}(v,y),y)\dots,y)$ we have $v' \overset{\gamma}{\equiv} \mathbf{s}_{i_{\ell+1}}(\dots \mathbf{s}_{i_{r-1} j}(\mathbf{s}_{i_r j}(e',y),y)\dots,y) = e'$, i.e. $v' \in N$ so that (2) yields $\mathbf{s}_{i_\ell j}(v',y) \overset{\gamma_-}{\equiv} e'$, and consequently $\mathbf{s}_j(v,y) = \mathbf{s}_{i_1 j}(\dots \mathbf{s}_{i_{\ell-1} j}(\mathbf{s}_{i_\ell j}(v',y),y)\dots,y) \overset{\gamma_-}{\equiv} \mathbf{s}_{i_1 j}(\dots \mathbf{s}_{i_{\ell-1} j}(e',y)\dots,y) = e'$. This establishes (4).

Now for a positive integer m and a $\gamma^\star$-block, i.e. the product $Q_{i_1,\dots,i_n} = d_{i_1}/\gamma^\star \times \dots \times d_{i_m}/\gamma^\star \subseteq A^m$ we define $(1+m)$-ary polynomial $\mathbf{q}_{i_1,\dots,i_m}(v,y_1,\dots,y_m)$ by putting

$$\mathbf{q}_{i_1,\dots,i_m}(v,y_1,\dots,y_m) = \mathbf{s}_{i_m}(\mathbf{s}_{i_{m-1}}(\dots \mathbf{s}_{i_2}(\mathbf{s}_{i_1}(v,y_1),y_2)\dots,y_{m-1}),y_m).$$

Then we observe that due to (4) we have

$$\begin{array}{lcll} \mathbf{q}_{i_1,\dots,i_m}(e',\overline{y}) & = & e', & \text{for all } \overline{y} \in A^m, \\ \mathbf{q}_{i_1,\dots,i_m}(v,\overline{y}) & \overset{\gamma_-}{\equiv} & e', & \text{for } v \in N \text{ and } \overline{y} \notin Q_{i_1,\dots,i_m}, \\ \mathbf{q}_{i_1,\dots,i_m}(a',\overline{y}) & \overset{\gamma_-}{\equiv} & a', & \text{for } \overline{y} \in Q_{i_1,\dots,i_m}. \end{array} \tag{5}$$

Note that the length of the polynomial $\mathbf{q}_{i_1,\dots,i_m}$ is bounded by $2^{O(m)}$, as it is a composition of m polynomials of the form $\mathbf{s}_j$ each of which having the size bounded by the same constant that depends only on $\mathbf{A}$.

Now, if $Q = Q_1 \cup \dots \cup Q_l$, with each Q_i being a single n-dimensional $\gamma^\star$-block, we sum up (in a balanced binary way) the polynomials $\mathbf{q}_i(\overline{x})$ produced, as above, separately for each block Q_i to get $\mathbf{q}_Q(\overline{x})$ satisfying

$$\begin{array}{lcll} \mathbf{q}_Q(e',\overline{x}) & = & e', & \text{for each } \overline{x} \in A^m, \\ \mathbf{q}_Q(v,\overline{x}) & \overset{\gamma_-}{\equiv} & e', & \text{if } v \in N \text{ and } \overline{x} \notin Q, \\ \mathbf{q}_Q(a',\overline{x}) & \overset{\gamma_-}{\equiv} & a', & \text{if } \overline{x} \in Q. \end{array} \tag{6}$$

The balanced way in which the $\mathbf{q}_i$'s are summed up guaranties that the resulting polynomial $\mathbf{q}_Q$ has its length bounded by $2^{O(m)}$, even if there are exponentially many summands determined by the blocks contained in A^m.

Now, let w be an n-ary polynomial over the field $GF(p)$ of degree s. To produce the associated polynomial $[w]_{\gamma,\top,a,e}$ of $\mathbf{A}$ we first produce $[w]_{\gamma,\top,a',e'}$ and then compose it with a unary polynomial $\mathbf{p}(x)$ that maps a' to a and e' to e (the algebra $\mathbf{A}$ has such a polynomial $\mathbf{p}$ as $(e,a) \in \gamma = \Theta(e',a')$). To produce the required n-ary polynomial $[w]_{\gamma,\top,a',e'}$ we first construct a $(1+n)$-ary polynomial $(w)_{\gamma,\top,a',e'}$ satisfying

$$\begin{array}{lcll} (w)_{\gamma,\top,a',e'}(e',\overline{x}) & = & e', & \text{for each } \overline{x} \in A^n, \\ (w)_{\gamma,\top,a',e'}(v,\overline{x}) & = & e', & \text{if } v \in N \text{ and } w(\mathsf{b}_\top(x_1),\dots,\mathsf{b}_\top(x_n)) = 0, \\ (w)_{\gamma,\top,a',e'}(v,\overline{x}) & = & v, & \text{if } v \in V \text{ and } w(\mathsf{b}_\top(x_1),\dots,\mathsf{b}_\top(x_n)) = 1, \end{array} \tag{7}$$

to put $[w]_{\gamma,\top,a',e'}(\overline{x}) = (w)_{\gamma,\top,a',e'}(a',\overline{x})$, which, by the last two lines in (7), will do the job.

To transform the polynomial $w(\overline{x})$ over the field $GF(p)$ into $(w)_{\gamma,\top,a',e'}(a',\overline{x})$, we first assume that $w(\overline{x})$ is given as sum of monomials and that the monomials of $w(\overline{x})$ are of the form $x_{i_1}\cdot\ldots\cdot x_{i_m}$ (with $m\leqslant s$) or are constant 1 (i.e., the monomials carry no leading constant – we can achieve this by replacing $2x$ by $x+x$ etc.). When passing from w to $(w)_{\gamma,\top,a',e'}(a',\overline{x})$ the constant 1 will be represented by the unary polynomial $\mathbf{e}_V(v)$, while the monomial $x_{i_1}\cdot\ldots\cdot x_{i_m}$ turns into $\mathbf{q}_{T^m}(v,x_{i_1},\ldots x_{i_m})$. Since $\top$ has been assumed to be a join of $\gamma^\star$-cosets, $\top^m$ is obviously a sum of $\gamma^\star$-blocks. Note also that for $\overline{y}\in A^m$ we have $\mathbf{q}_{\top^m(\overline{y})}(a',\overline{y})\subseteq e'/\gamma_-\cup a'/\gamma_-$ and, due to (6), we have

$$\mathbf{q}_{\top^m(\overline{y})}(a',\overline{y})\overset{\gamma_-}{\equiv}a'\quad\text{iff}\quad\overline{y}\in\top^m\quad\text{iff}\quad\mathsf{b}_\top(y_1)\cdot\mathsf{b}_\top(y_2)\cdot\ldots\cdot\mathsf{b}_\top(y_m)=1. \tag{8}$$

Now we sum up (appropriate amount of) the polynomials $\mathbf{e}_V(v)$ and appropriate polynomials $\mathbf{q}_Q(v,x_{i_1},\ldots x_{i_m})$ to get $(w)_{\gamma,\top,a',e'}(a',\overline{x})$ (again we have to be careful to use a balanced summation tree). Next we are using the fact that there is an isomorphism of the group $\mathbb{Z}_p$ with the subgroup of $((\mathbf{A}|_N)/\gamma_-,+)$ generated by a'/γ_- that sends 1 to a'/γ_-. Applying this isomorphism to (8) we get

$$\begin{array}{lcll}(w)_{\gamma,\top,a',e'}(e',\overline{x}) & = & e', & \text{for each } \overline{x}\in A^m,\\ (w)_{\gamma,\top,a',e'}(a',\overline{x}) & \overset{\gamma_-}{\equiv} & e', & \text{if } w(\mathsf{b}_\top(x_1),\ldots,\mathsf{b}_\top(x_n))=0,\\ (w)_{\gamma,\top,a',e'}(a',\overline{x}) & \overset{\gamma_-}{\equiv} & a', & \text{if } w(\mathsf{b}_\top(x_1),\ldots,\mathsf{b}_\top(x_n))=1.\end{array} \tag{9}$$

Since there are at most $O(n^s)=O(2^{s\cdot\log n})$ monomials of degree at most s, while the polynomials $\mathbf{q}_{\top^m}$ representing them has sizes bounded by $2^{O(s)}$ the length of $(w)_{\gamma,\top,a',e'}(v,\overline{x})$ is at most $2^{O(s\cdot\log n)}$.

Finally to pass from (9) to (7) we start with reminding that for a fixed $\overline{x}\in A^n$ the mapping $V\ni v\longmapsto(w)_{\gamma,\top,a',e'}(v,\overline{x})\in V$ either permutes the set V or collapses $\gamma|_V$ to γ_-, i.e. it is constant modulo γ_- on $\gamma|_V$-classes. Thus, iterating $(w)_{\gamma,\top,a',e'}(v,\overline{x})$ in the first variable a sufficient number of times, we may additionally assume that $(w)_{\gamma,\top,a',e'}(v,\overline{x})$ is either the identity map on V or it is constant, modulo γ_-, on $\gamma|_V$-classes, depending on whether $w(\mathsf{b}(\overline{x}))$ is 1 or 0. Thus, $w(\mathsf{b}(\overline{x}))=1$ gives us the third equality in (7).

In the other case, $(w)_{\gamma,\top,a',e'}(v,\overline{x})$ collapses the entire trace N to e'/γ_- so that we have $(w)_{\gamma,\top,a',e'}(v,\overline{x})\overset{\gamma_-}{\equiv}e'$ for $v\in N$. Now we go down along the chain $0=\theta_0\prec\theta_1\prec\ldots\prec\theta_l=\gamma_-$ to show that $(w)_{\gamma,\top,a',e'}(v,\overline{x})\overset{\theta_i}{\equiv}e'$ yields $(w)_{\gamma,\top,a',e'}(v,\overline{x})\overset{\theta_{i-1}}{\equiv}e'$. Since the unary polynomial $\mathbf{f}:V\ni v\longmapsto(w)_{\gamma,\top,a',e'}(v,\overline{x})\in V$ does not permute V, $\mathbf{f}(A)=\mathbf{f}(V)\subsetneq V$. This gives that for each $i=l,\ldots,1$ the polynomial $\mathbf{f}$ collapses θ_i to θ_{i-1}, as otherwise V would properly contain a (θ_{i-1},θ_i)-minimal set. But this is impossible in view of Lemma 4.30 in [17]. Therefore, composing l times the polynomial $\mathbf{f}$ we get that this composition satisfies $(w)_{\gamma,\top,a',e'}(v,\overline{x})=e'$ so that (7) is shown. This iteration inflates the size of $(w)_{\gamma,\top,a',e'}(v,\overline{x})$ by raising it to the l-th power so that it is still bounded by $2^{O(s\cdot\log n)}$. ◀

We are going to use the following fact that is borrowed from [3] (see [21, Fact 3.4] for a recent proof).

▶ **Fact 15.** *Let p be a prime and $\nu\geqslant 1$ be an integer. Then there is a polynomial $w(\overline{x})\in GF(p)[\overline{x}]$ of degree at most $p^\nu-1$, such that for $\overline{x}\in\{0,1\}^n\subseteq\mathbb{Z}_p^n$ we have*

$$w(\overline{x})=\begin{cases}0, & \textit{if } \left|\overline{x}^{-1}(0)\right|\equiv 0 \textit{ modulo } p^\nu,\\ 1, & \textit{else.}\end{cases}$$

Now we are in a position to prove the following.

▶ **Theorem 16.** *Let* $\mathbf{A}$ *be a finite solvable Malcev algebra and* α, β *be two collaborating join-irreducible congruences of different characteristics. Then, assuming the (randomized) Exponential Time Hypothesis, the following hold:*

- *if* $(\alpha_- : \alpha) \vee (\beta_- : \beta) < 1$*, then* PolSat($\mathbf{A}$) *is not in* P *(resp.* RP*),*
- *if* $(\alpha_- : \alpha) < 1$ *and* $(\beta_- : \beta) < 1$*, then* 2-ListPolSat($\mathbf{A}$) *is not in* P *(resp.* RP*).*

More precisely, we show that under rETH there is no randomized algorithm of running time $2^{o((\log n/\log\log n)^2)}$ for these problems. Before we prove Theorem 16, let us point out how it implies Theorem 2 from the introduction:

Proof of Theorem 2. As $\mathbf{A}$ and $\mathbf{B}$ are minimal normal subgroups, they are join-irreducible. As $\mathbf{A} \neq \mathbf{B}$, they are collaborating and by assumption they are of different characteristic. The condition $C_{\mathbf{G}}(\mathbf{A}) \cdot C_{\mathbf{G}}(\mathbf{B}) \neq \mathbf{G}$ is just the same as $(\alpha_- : \alpha) \vee (\beta_- : \beta) < 1$ written in a group language (here $\mathbf{A} = \alpha$, $\mathbf{B} = \beta$ and $\alpha_- = \beta_- = \{1\}$). Now we apply Theorem 16. ◀

Proof of Theorem 16. Since α, β are collaborating, without loss of generality we may assume that there are 3 different elements $a, e, b \in A$ such that $(a, e) \in \alpha$ and $(e, b) \in \beta - \alpha$. Let $\mathbf{d}$ denote a Malcev term for $\mathbf{A}$. Observe here that

(*) for $u \in \{a, e\}$ and $v \in \{b, e\}$ we have $\mathbf{d}(u, e, v) = e$ iff $u = e = v$.

Indeed, $\mathbf{d}(a, e, e) = a$ and $\mathbf{d}(e, e, b) = b$, while $\mathbf{d}(a, e, b) = e$ would give $b = \mathbf{d}(e, e, b) \overset{\alpha}{\equiv} \mathbf{d}(a, e, b) = e$, contrary to our choice of $(b, e) \in \beta - \alpha$.

To unify our arguments for both PolSat($\mathbf{A}$) and 2-ListPolSat($\mathbf{A}$), observe that $\alpha^* = (\alpha_- : \alpha) < 1$ and $\beta^* = (\beta_- : \beta) < 1$ gives $\alpha^* \cup \beta^* \neq 1$, by Lemma 13. This allows us to pick a pair $(c, d) \in A^2$ satisfying

[P] $(c, d) \notin \alpha^* \vee \beta^*$,

[LP] $(c, d) \notin \alpha^* \cup \beta^*$,

depending on which of the two problems PolSat($\mathbf{A}$), 2-ListPolSat($\mathbf{A}$) we are considering.

With the help of Lemma 14 and Fact 15, to each 3-CNF-SAT formula $\Phi(\overline{x})$ having n variables $\overline{x} = (x_1, \ldots, x_n)$ and ℓ clauses we are going to associate two n-ary polynomials $\mathbf{t}^\Phi(\overline{x})$ and $\mathbf{s}^\Phi(\overline{x})$ of length $2^{O(\sqrt{\ell} \cdot \log \ell)}$ such that

[P] $\Phi(\overline{x})$ is satisfiable iff the equation $\mathbf{t}^\Phi(\overline{x}) = e$ has a solution $\overline{x}$ in A^n,

[LP] $\Phi(\overline{x})$ is satisfiable iff the equation $\mathbf{s}^\Phi(\overline{x}) = e$ has a solution $\overline{x}$ in the set $\{c, d\}^n$.

To do that, let p and q be the characteristics of α and β, respectively, and pick $\mu, \nu \in \mathbb{N}$ with $p^{\mu-1} \leqslant \sqrt{\ell} < p^\mu$ and $q^{\nu-1} \leqslant \sqrt{\ell} < q^\nu$. Now Fact 15 supplies us with ℓ-ary polynomials (with their variables c_i's later to be substituted by the clauses C_i's of the formula $\Phi(\overline{x})$):

- $w_p(c_1, \ldots, c_\ell) \in GF(p)[\overline{c}]$, with degree bounded by $p^\mu - 1$,
- $w_q(c_1, \ldots, c_\ell) \in GF(q)[\overline{c}]$, with degree bounded by $q^\nu - 1$,

which on $\overline{c} \in \{0, 1\}^\ell$ take values from $\{0, 1\}$, such that $w_p(\overline{c}) = 0$ iff $\left|\overline{c}^{-1}(0)\right| \equiv 0 \mod p^\mu$ (resp. $w_q(\overline{c}) = 0$ iff $\left|\overline{c}^{-1}(0)\right| \equiv 0 \mod q^\mu$).

With a 3-ary clause $C(z^1, z^2, z^3)$ we associate a polynomial $C'(z^1, z^2, z^3)$ of degree 3 over $GF(p)$ (resp. $GF(q)$) in an obvious way, so that for example the clause $z^1 \vee z^2 \vee \neg z^3$ goes to $1 - ((1 - z^1) \cdot (1 - z^2) \cdot z^3)$. Now for $\Phi(\overline{x}) = \bigwedge_{i=1}^{\ell} C_i$ we feed up the polynomials w_p and w_q by substituting C_i' for the variable c_i to produce (at most 3ℓ-ary) polynomials $w_p^\Phi(\overline{z})$ and $w_q^\Phi(\overline{z})$ of degrees bounded by $3(p^\mu - 1)$ and $3(q^\nu - 1)$, respectively. Note that again the new polynomials w_p^Φ (or w_q^Φ) on arguments from $\{0, 1\}$ return values from the same set $\{0, 1\}$, but this time 0 is taken exactly on valuations of variables in Φ under which the number of unsatisfied clauses is divisible by p^μ (or by q^ν, respectively). The important feature is that

(**) the polynomials w_p^Φ and w_q^Φ simultaneously return 0 on a valuation from the set $\{0, 1\}$ iff this valuation satisfies Φ.

Indeed, $w_p^\Phi(\overline{z}) = 0 = w_q^\Phi(\overline{z})$ tells us that the number of unsatisfied clauses in Φ is divisible both by p^μ and q^ν. Since $p \neq q$, this number is divisible by $p^\mu \cdot q^\nu > \ell$. However, there are only ℓ clauses, so that none of them is unsatisfied by $\overline{z}$.

Now with the help of Lemma 14 we are able to define

$$\mathbf{t}^\Phi(\overline{x}) = \mathbf{d}\left(\left[w_p^\Phi\right]_{\alpha,\top_\alpha,a,e}(\overline{x}), e, \left[w_q^\Phi\right]_{\beta,\top_\beta,b,e}(\overline{x})\right),$$

where a and b are as above and the sets $\top_\alpha, \top_\beta \neq \emptyset, A$ are chosen to be sums of α^*- or β^*-cosets, respectively. Since our assumption for the problem PolSat says that $\alpha^* \vee \beta^* < 1$, one way to ensure this is by putting $\top_\alpha = \top_\beta$ to be a single $\alpha^* \vee \beta^*$-coset, e.g. $d/(\alpha^* \vee \beta^*)$. Now, combining Lemma 14 with the properties (*) and (**), we know that for $\overline{x} = (x_1, \ldots, x_n) \in A^n$ we have $\mathbf{t}^\Phi(\overline{x}) = e$ iff $\Phi(\mathsf{b}(x_1), \ldots, \mathsf{b}(x_n)) = 1$, where $\mathsf{b} = \mathsf{b}_{\top_\alpha} = \mathsf{b}_{\top_\beta}$.

In case of 2-ListPolSat the polynomial $\mathbf{s}^\Phi$ is defined in a similar way, i.e. we put

$$\mathbf{s}^\Phi(\overline{x}) = \mathbf{d}\left(\left[w_p^\Phi\right]_{\alpha,\top_\alpha,a,e}(\overline{x}), e, \left[w_q^\Phi\right]_{\beta,\top_\beta,b,e}(\overline{x})\right),$$

but this time we cannot ensure $\alpha^* \vee \beta^* < 1$. Instead we can bound the range for the x_i's in A to be smaller. In fact, we restrict these ranges to $\{c, d\}$ so that, by the very same argument as before, the sets $\top_\alpha = d/\alpha^*$ and $\top_\beta = d/\beta^*$ will do the job.

Finally, Fact 15 together with Lemma 14 ensure us that the sizes of $\left[w_p^\Phi\right]_{\alpha,\top_\alpha,a,e}$, and $\left[w_q^\Phi\right]_{\beta,\top_\beta,b,e}$, and therefore of both $\mathbf{t}^\Phi(\overline{x})$ and $\mathbf{s}^\Phi(\overline{x})$, are bounded by $2^{O(\sqrt{\ell}\cdot\log\ell)}$. Thus, ETH (resp. rETH) puts both the problems PolSat($\mathbf{A}$) and 2-ListPolSat($\mathbf{A}$) outside P (resp. RP). Indeed, assume that PolSat($\mathbf{A}$) or 2-ListPolSat($\mathbf{A}$) could be decided in (randomized) time $2^{o((\log m/\log\log m)^2)}$ where m is the length of the input polynomial. Then we could decide a 3-CNF-SAT instance of length ℓ in (randomized) time

$$2^{o((\log 2^{O(\sqrt{\ell}\cdot\log\ell)}/\log\log 2^{O(\sqrt{\ell}\cdot\log\ell)})^2)} = 2^{o(\sqrt{\ell}^2\cdot\log^2\ell/\log^2(\sqrt{\ell}\cdot\log\ell))} = 2^{o(\ell)}.$$ ◀

5 A Dichotomy for ProgramSat

Proof of Theorem 1. First, consider the case that $\mathbf{G}$ has a normal p-subgroup $\mathbf{G}_p$ such that $\mathbf{G}/\mathbf{G}_p$ is nilpotent (possibly $\mathbf{G}_p$ is trivial). Then $\mathbf{G}/\mathbf{G}_p = \mathbf{H}_p \times \mathbf{H}$ for some maximal p-group $\mathbf{H}_p$ (also $\mathbf{H}_p$ might be trivial). We denote the preimage of $\mathbf{H}_p$ in $\mathbf{G}$ by $\widetilde{\mathbf{G}}_p = \mathbf{H}_p\mathbf{G}_p$. Since $|H|$ and $\left|\widetilde{G}_p\right|$ are coprime, by the Schur-Zassenhaus theorem (Fact 7), we conclude that $\mathbf{G} = \widetilde{\mathbf{G}}_p \rtimes \mathbf{H}$ and so, by Corollary 12, ListPolSat($\mathbf{G}$) and ProgramSat($\mathbf{G}$) are in RP under CDH.

On the other hand, assume that $\mathbf{G}$ is not of the above form. If $\mathbf{G}$ is non-solvable, ListPolSat($\mathbf{G}$) and ProgramSat($\mathbf{G}$) are NP-complete by [14], hence, not in RP under rETH. If $\mathbf{G}$ is solvable but does not have a nilpotent normal subgroup $\mathbf{N}$ with nilpotent quotient $\mathbf{G}/\mathbf{N}$, then [23] (for certain cases also [20, 30]) shows that PolSat($\mathbf{G}$) (hence, also ListPolSat($\mathbf{G}$) and ProgramSat($\mathbf{G}$)) is not in P under ETH. The same proof shows that these problems are not in RP under rETH (as a randomized algorithm for ListPolSat($\mathbf{G}$) or ProgramSat($\mathbf{G}$) would lead to a randomized algorithm for 3-CNF-SAT).

Finally, let $\mathbf{N}$ denote the smallest normal subgroup such that $\mathbf{G}/\mathbf{N}$ is nilpotent. Such an $\mathbf{N}$ exists and it is nilpotent as we excluded already all the other cases. Moreover, we know that $|N|$ has at least two distinct prime divisors p and q since otherwise, we would be in the RP case. Notice that $\mathbf{N}$ is the direct product of its Sylow subgroups. Thus, after taking

a quotient (recall that ListPolSat($\mathbf{G}/\mathbf{H}$) $\leqslant$ ListPolSat($\mathbf{G}$) and ProgramSat($\mathbf{G}/\mathbf{H}$) $\leqslant$ ProgramSat($\mathbf{G}$), see Lemma 8), we may assume that there are precisely two non-trivial normal subgroups $\mathbf{A}$ and $\mathbf{B}$ of $\mathbf{G}$ below $\mathbf{N}$ and $|A| = p^\alpha$ and $|B| = q^\beta$ for some $\alpha, \beta \geqslant 1$ and $\mathbf{N} = \mathbf{A} \times \mathbf{B}$. Clearly $\mathbf{A}$ and $\mathbf{B}$ are join-irreducible.

Now assume for a contradiction that $C_{\mathbf{G}}(\mathbf{A}) = G$. Observe that $C_{\mathbf{G}}(\mathbf{A}) = C_G(\mathbf{N}/\mathbf{B})$ because $[g, a] = 1$ if and only if $[g, ab] \in B$ for $b \in B$. This means that $\mathbf{N}/\mathbf{B}$ is in the center of $\mathbf{G}/\mathbf{B}$. Thus, as $\mathbf{G}/\mathbf{N} = (\mathbf{G}/\mathbf{B})/(\mathbf{N}/\mathbf{B})$ is nilpotent, this implies that $\mathbf{G}/\mathbf{B}$ is nilpotent contradicting that $\mathbf{N}$ is the smallest normal subgroup such that $\mathbf{G}/\mathbf{N}$ is nilpotent.

By symmetry we also have $C_{\mathbf{G}}(\mathbf{B}) \neq G$. Since $\mathbf{A}$ and $\mathbf{B}$ are collaborating, we have verified the requirements of Theorem 16 showing that 2-ListPolSat($\mathbf{G}$) is not in RP under rETH. By Lemma 8 also ListPolSat($\mathbf{G}$) and ProgramSat($\mathbf{G}$) are not in RP under rETH. ◀

6 Unconditional Algorithms

Proving lower bounds for a non-trivial computational model is usually a challenging task. Rare examples of results of such kind are either proven in some very restricted settings, or rely on additional assumptions. The same issue affects programs over groups and their expressiveness of AND, where essentially our knowledge can be summarized as follows:

(B) For a nilpotent group $\mathbf{N}$ there is a constant $d_{\mathbf{N}}$ such that there is no $\mathbf{N}$-program for the n-ary AND function for $n \geqslant d_{\mathbf{N}}$ ([5, Corollary to Theorem 6]).

(C) If $\mathbf{Q} = \mathbf{G}_q \rtimes \mathbf{A}$ for some q-group $\mathbf{G}_q$ and abelian group $\mathbf{A}$, then $\gamma_{\mathrm{Prog},\mathbf{Q}}(n) \in 2^{\Omega(n)}$ ([5, Corollary to Theorem 9], recall the definition of $\gamma_{\mathrm{Prog},\mathbf{Q}}(n)$ in Section 3).

Lemma 10 implies that these two cases lead to RP algorithms for ProgramSat, ListPolSat, and PolSat. In fact, for ProgramSat for nilpotent groups and for PolSat, in both cases (B) and (C), even polynomial time algorithms have been obtained [14, 12]; however, as for now, ProgramSat in case (C) is only known to have quasi-polynomial time algorithms [4].

The main aim of the next theorem is to present a new lower bound result for groups that are direct products of these presented in (B), (C), which then by Lemma 10 also leads to RP algorithms. To the best of our knowledge this is the first result going beyond the cases (B) and (C) in this direction.

▶ **Theorem 17.** *Let $\mathbf{N}$ be a nilpotent group, $\mathbf{A}$ an abelian group, $\mathbf{Q} = \mathbf{G}_q \rtimes \mathbf{A}$ for some q-group $\mathbf{G}_q$ and let $\mathbf{G} \leqslant \mathbf{N} \times \mathbf{Q}$. Then $\gamma_{\mathrm{Prog},\mathbf{G}}(n) \in 2^{\Omega(n)}$.*

The fact that this lower bound applies also to subgroups of the product will allow us to construct a series of natural examples of groups for which we achieve efficient algorithms. Of particular interest is the case that $\mathbf{A}$ is also a subgroup of $\mathbf{N}$, so there is some non-trivial interaction between $\mathbf{N}$ and $\mathbf{Q}$.

The proof of Theorem 17 relies on a similar construction as used in [7] for showing that for every low degree polynomial there is a large affine subspace on which the polynomial is constant. For the proof we need some preparation: For $\overline{a}, \overline{b} \in \prod_{i=1}^n |A_i|$ with $\overline{a} = (a_1, \ldots, a_n)$ and $\overline{b} = (b_1, \ldots, b_n)$ we define $\mathrm{HammingDist}(\overline{a}, \overline{b}) = |\{\, i \in [1\,..\,n] \mid a_i \neq b_i \,\}|$. The following easy combinatorial observation has been used in slightly different forms in [4, Theorem 2] or [21, Theorem 6.1].

▶ **Fact 18.** *Let f be an n-ary indicator function with domain $\mathcal{A} = \prod_{i=1}^n A_i$ with $|A_i| \geqslant 2$ for all i and $\mathrm{size}(f) < \gamma(n)$. Let $\overline{b} \in f^{-1}(1)$. Then there is some $\overline{a} \in f^{-1}(1)$ with $\overline{b} \neq \overline{a}$ and $\mathrm{HammingDist}(\overline{a}, \overline{b}) \leqslant \gamma^{-1}(\mathrm{size}(f)) + 1$.*

Proof. As $\mathrm{size}(f) < \gamma(n)$, there is some $\overline{c} \in f^{-1}(1)$ with $\overline{c} \neq \overline{b}$. Writing $\overline{b} = (b_1, \ldots, b_n)$ and $\overline{c} = (c_1, \ldots, c_n)$ that means $b_i \neq c_i$ for some i. Now, set $\overline{b}' = (b_1, \ldots, b_{i-1}, c_i, b_{i+1}, \ldots, b_n)$, i.e., $\overline{b}'$ agrees with $\overline{b}$ on all but the i-th coordinate. Consider some $\overline{a}$ with $f[x_i/c_i](\overline{a}) = 1$ of minimal Hamming distance k to $\overline{b}'$ (note that possibly $\overline{a} = \overline{b}'$). Then setting all variables on which $\overline{a}$ and $\overline{b}'$ agree to this constant and restricting all other variables x_j (with $a_j \neq b_j$) to $\{a_j, b_j\}$, we obtain a k-ary spike. Thus, $\mathrm{size}(f) \geqslant \gamma(k)$ and Fact 18 follows. ◀

Proof of Theorem 17. It is enough to consider $\mathbf{G} = \mathbf{N} \times \mathbf{Q}$, as lower bounds for subgroups are inferred from their containing group. We can think of a $\mathbf{G}$-program in the direct product as a system of two separate programs sharing the same variables. Thus, let $\mathbf{p}$ be an $\mathbf{N}$-program and $\mathbf{q}$ a $\mathbf{Q}$-program with variables $x_1, \ldots, x_n$ for a spike (i.e., the n-ary AND function). Without loss of generality, we can assume that the all-zero vector $\overline{0}$ is the only satisfying assignment to the conjunction of $\mathbf{p}$ and $\mathbf{q}$ (we can achieve this by simply replacing variables by their negations if necessary – this does not increase the size of the programs). Thus, we can also associate $\mathbf{p}, \mathbf{q}$ with one-element accepting sets $\{s_{\mathbf{p}}\}, \{s_{\mathbf{q}}\}$ respectively. Now we show that $|\mathbf{q}| \in 2^{\Omega(n)}$.

We will identify a vector $\overline{b} \in \{0,1\}^n$ with the subset $\overline{b}^{-1}(1)$ of $[1\,..\,n]$. By (B) there is a constant d such that no $\mathbf{N}$-program can be a d-ary spike. We are going to exploit this fact by proving that, if $\mathbf{q}$ is relatively short, then there must be a relatively large boolean cube on which $\mathbf{p}$ behaves like an AND. Note that for $\overline{b}_1, \ldots, \overline{b}_d \in \{0,1\}^n$ with $\overline{b}_i \cap \overline{b}_j = \varnothing$ for all $i \neq j$ we can simulate the behaviour of $\mathbf{p}$ on $B = \left\{ \sum_{i=1}^d \alpha_i \overline{b}_i \;\middle|\; \alpha_i \in \{0,1\} \right\}$ by creating a new d-ary program $\hat{\mathbf{p}}(y_1, \ldots, y_d) = \mathbf{p}(\sum_{i=1}^d y_i \overline{b}_i)$. Indeed, consider an instruction $\langle j, g, h\rangle$ of $\mathbf{p}$: if j is not in any of the $\overline{b}_i$ just replace it with constant g and if $j \in \overline{b}_i$ for some i replace it with the instruction $\langle i, g, h\rangle$. To find a cube of our interest we start with the following claim.

▷ Claim 19. Let $\tilde{\mathbf{Q}} = \mathbf{Q}^{2^d}$. If $|\mathbf{q}| < \gamma_{\mathrm{Prog},\tilde{\mathbf{Q}}}(n/d - 1)$, then there are $\overline{b}_1, \ldots, \overline{b}_d \in \{0,1\}^n$ with $\overline{b}_i \cap \overline{b}_j = \varnothing$ for all $i \neq j$ such that $\mathbf{q}(\overline{x}) = 1$ for all $\overline{x} \in \left\{ \sum_{i=1}^d \alpha_i \overline{b}_i \;\middle|\; \alpha_i \in \{0,1\} \right\} \subseteq \{0,1\}^n$.

Proof. For $k \in [1\,..\,d]$ consider the group $\mathbf{Q}_k = \mathbf{Q}^{2^k}$. As having more coordinates clearly gives more expressive power to a program, we have $\gamma_{\mathrm{Prog},\tilde{\mathbf{Q}}}(n) \leqslant \gamma_{\mathrm{Prog},\mathbf{Q}_k}(n)$.

Assume we already constructed $\overline{b}_1, \ldots, \overline{b}_k$ for some $k \in [0\,..\,d-1]$ (for $k = 0$ that means we have no $\overline{b}_i$'s). Let $X_k = \bigcup_{i=1}^k \overline{b}_i$. We additionally require by induction that $|X_k| \leqslant kn/d$. First observe that a new vector $\overline{b}_{k+1}$ which could extend the sequence $\overline{b}_1, \ldots, \overline{b}_k$ needs to satisfy a system of 2^k equations $\mathbf{q}(\alpha_1 \overline{b}_1 + \alpha_2 \overline{b}_2 + \ldots + \alpha_k \overline{b}_k + \overline{x}) = 1$, one equation for each $\overline{\alpha} \in \{0,1\}^k$. As we expect $\overline{b}_{k+1}$ to have disjoint support with all preceding $\overline{b}_i$'s, we just put $x_i = 0$ for $i \in X_k$, so that each $\mathbf{q}_\alpha(\overline{x}) = \mathbf{q}(\alpha_1 \overline{b}_1 + \ldots + \alpha_k \overline{b}_k + \overline{x})$ is of arity $n - |X_k|$. Notice that we can encode those 2^k conditions as one program condition in the group $\mathbf{Q}_k = \mathbf{Q}^{2^k}$. To produce a program $\hat{\mathbf{q}}(\overline{x}) = (\mathbf{q}_\alpha(\overline{x}))_{\alpha \in \{0,1\}^k}$ with associated accepting set $\{s_{\mathbf{q}}\}^{2^k}$, we replace each instruction of $\mathbf{q}$ with one in the new domain: whenever $j \notin X_k$ we just replace $\langle j, g, h\rangle$ with $\langle j, (g, \ldots, g), (h, \ldots, h)\rangle$ and whenever $j \in \overline{b}_i$ (for some $i \in [1\,..\,k]$) we replace it by the constant $(c_\alpha)_{\alpha \in \{0,1\}^k}$, where $c_\alpha = g$ when $\alpha_i = 0$ and $c_\alpha = h$ when $\alpha_i = 1$.

Since $|\hat{\mathbf{q}}| \leqslant |\mathbf{q}| \leqslant \gamma_{\mathrm{Prog},\tilde{\mathbf{Q}}}(n/d - 1)$ and $n - |X_k| \geqslant \frac{n}{d}$, we know by Fact 18 that not only $\overline{0}$ is a solution to $\hat{\mathbf{q}}(x) = 1$ but we have another solution with Hamming weight at most $\gamma^{-1}_{\mathrm{Prog},\tilde{\mathbf{Q}}}(|\hat{\mathbf{q}}|) + 1 \leqslant \gamma^{-1}_{\mathrm{Prog},\tilde{\mathbf{Q}}}(\gamma_{\mathrm{Prog},\tilde{\mathbf{Q}}}(n/d - 1)) + 1 = n/d$. We can clearly choose this solution to become $\overline{b}_{k+1}$ and finish the induction by noticing that $|X_{k+1}| = |X_k| + \left|\overline{b}_{k+1}\right| \leqslant kn/d + n/d = (k+1)n/d$. ◁

Finally, by (C), we know that $\gamma_{\mathrm{Prog},\tilde{\mathbf{Q}}}(n) \geqslant 2^{\delta n - C}$ for some suitable constants δ and C. Assume for a contradiction that $|\mathbf{q}| \leqslant 2^{\delta n/d - C - d - 1} \leqslant \gamma_{\mathrm{Prog},\tilde{\mathbf{Q}}}(n/d - 1)$. By Claim 19 we obtain a boolean cube $B = \left\{ \sum_{i=1}^d \alpha_i \overline{b}_i \;\middle|\; (\alpha_1, \ldots, \alpha_d) \in \{0,1\}^d \right\} \subseteq \{0,1\}^n$ with $\mathbf{q}(B) = \{1\}$.

It means that the equation $\mathbf{p}(x) = 1$ has only one solution $\overline{0}$ in the set B, otherwise the system $(\mathbf{p}(\overline{x}), \mathbf{q}(\overline{x}))$ would not define the spike (with accepting set $\{(s_{\mathbf{p}}, s_{\mathbf{q}})\}$). But now to get a contradiction define a program $\hat{\mathbf{p}}(y_1, \ldots, y_d) = \mathbf{p}(y_1\overline{b}_1 + \ldots + y_d\overline{b}_d)$, which must be a d-ary spike – contrary to the choice of d. ◀

As an immediate consequence of Theorem 17 and Lemma 10 we get the following.

▶ **Corollary 20.** *Let* $\mathbf{N}$ *be a nilpotent group,* $\mathbf{A}$ *abelian,* $\mathbf{Q} = \mathbf{G}_q \rtimes \mathbf{A}$ *for some* q*-group* $\mathbf{G}_q$ *and let* $\mathbf{G} \leqslant \mathbf{N} \times \mathbf{Q}$*. Then* ProgramSat($\mathbf{G}$)*,* ListPolSat($\mathbf{G}$) *or* PolSat($\mathbf{G}$) *have the none-or-many property.*

In particular, ProgramSat($\mathbf{G}$)*,* ListPolSat($\mathbf{G}$) *and* PolSat($\mathbf{G}$) *are in* RP*.*

Proof of Corollary 5. We can apply Corollary 20 to dihedral groups of order $2^\alpha p^\beta$. Indeed, each such group is a subgroup of $\mathbf{D}_{2^\alpha} \times \mathbf{D}_{p^\beta}$, where $\mathbf{D}_{2^\alpha}$ is nilpotent and $\mathbf{D}_{p^\beta}$ is isomorphic to the semidirect product $\mathbb{Z}_{p^\beta} \rtimes \mathbb{Z}_2$. So for each such group PolSat is in RP (if $\alpha \in \{0, 1\}$ it is even in P by [18, Corollary 2]). On the other hand, all other dihedral groups $\mathbf{D}_m$ have a quotient $\mathbf{D}_k$ where k is odd and has exactly two different prime divisors. By [21, Theorem 7.1], PolSat($\mathbf{D}_k$) is not in P under ETH. The same argument also shows that it is not in RP under rETH. By Lemma 8 this transfers to $\mathbf{D}_m$. We can see this also as a consequence of Theorem 2: the minimal normal subgroups $\mathbf{A}$ and $\mathbf{B}$ are just cyclic subgroups of $\mathbb{Z}_m$ of coprime odd order. The centralizer of both of them is $\mathbb{Z}_m < \mathbf{D}_m$. Hence, Theorem 2 tells us that PolSat($\mathbf{D}_m$) is not in RP under rETH. ◀

The positive case of the proof of Corollary 5 obviously generalizes to groups $\mathbf{G}$ with a nilpotent normal subgroup $\mathbf{N}$ of order $p^\alpha q^\beta$ such that $\mathbf{G}/\mathbf{N}$ is abelian of order p^γ (just apply the Schur-Zassenhaus Theorem, Fact 7). In particular, this applies as follows:

▶ **Example 21.** Note that the complexity of PolSat for three natural examples of order 24, namely the dihedral group $\mathbf{D}_{12}$, the quaternion group $\mathbf{Q}$ acting over $\mathbb{Z}_3$ (i.e. $\mathbb{Z}_3 \rtimes \mathbf{Q}$), and the group $(\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_3) \rtimes \mathbb{Z}_2$ was left unsolved in [12, Problem 3]. Now our Corollary 20 covers all of these examples.

7 Extending Randomized Algorithms for PolSat

The smallest example of a group of Fitting length two for which we know superpolynomial lower bounds under ETH is $\mathbf{D}_{15}$ [21]. We can embed $\mathbf{D}_{15}$ into the group $\mathbf{D}_3 \times \mathbf{D}_5$ which has polynomial-time decidable PolSat. On the contrary, both ProgramSat and ListPolSat share superpolynomial complexities under ETH in this case. Moreover, for a given group $\mathbf{G}$, under assumption of CDH and ETH the problem ProgramSat($\mathbf{G}$) is in P whenever ListPolSat($\mathbf{G}$) is. So, in a sense, complexities of ProgramSat and ListPolSat seem to coincide, while the complexity of PolSat may differ in certain cases. It is due to the fact that upper bounds for PolSat are not inherited by the subgroups in a very strong way:

▶ **Observation 22.** *Every group of Fitting length two can be embedded into a group with* PolSat *in* RP *under CDH. Moreover, if* $\mathbf{G}$ *is nilpotent-by-abelian, it can be embedded into a group with* PolSat *in* P *(unconditionally).*

Proof. Let $\mathbf{G}$ be a group of Fitting length two. Thus, there is some nilpotent normal subgroup $\mathbf{N}$ such that $\mathbf{G}/\mathbf{N}$ is nilpotent. Since $\mathbf{N}$ is nilpotent, we can write it as a direct product $\mathbf{N} = \mathbf{G}_{p_1} \times \cdots \times \mathbf{G}_{p_k}$ for p_i-groups $\mathbf{G}_{p_i}$. Let $\mathbf{N}_i = \prod_{j \neq i} \mathbf{G}_{p_j}$. Then $\mathbf{G}_{p_i}$ is a

normal nilpotent subgroup of $\mathbf{G}/\mathbf{N}_i$ and $(\mathbf{G}/\mathbf{N}_i)/\mathbf{G}_{p_i}$ is nilpotent. Thus, by Theorem 1 PolSat$(\mathbf{G}/\mathbf{N}_i)$ is in RP under CDH. Finally, notice that $\mathbf{G}$ embeds into the direct product $\prod_{i=1}^{k} \mathbf{G}/\mathbf{N}_i$, which by Lemma 8 has also PolSat in RP under CDH.

For the second part just observe that, if $\mathbf{G}$ is nilpotent-by-abelian, $\mathbf{G}/\mathbf{N}_i$ has a normal p_i-subgroup with abelian quotient and so by [12, Theorem 1] PolSat is in P. ◀

The unusual properties of PolSat allow us to create larger classes of groups with polynomial time algorithms than for the other two problems.

Proof of Theorem 4. We start with some preparation and decompose $\mathbf{G}$ into smaller groups. Since $|G|$ has only two prime factors, say p and q, we can write $\mathbf{N} = \mathbf{N}_q \times \mathbf{N}_p$ and $\mathbf{H} = \mathbf{H}_p \times \mathbf{H}_q$ where $\mathbf{N}_p, \mathbf{H}_p$ are p-groups and $\mathbf{N}_q, \mathbf{H}_q$ are q-groups. Notice that $\mathbf{N}_p$ and $\mathbf{N}_q$ are also normal in $\mathbf{G}$ (this is because they are characteristic subgroups of $\mathbf{N}$). We define subsets L, R, P, Q of G by putting $L = N_q H_p$, $R = N_p H_q$, $P = N_p H_p$ and $Q = N_q H_q$. Notice that we defined indeed subgroups $\mathbf{L}, \mathbf{R}, \mathbf{P}$, and $\mathbf{Q}$ since each of them is a product of a normal subgroup and a subgroup (though they might not be normal subgroups) and $LR = G = PQ$. In particular, we can write each $g \in G$ uniquely as $g = \ell r$ where $\ell \in L$ and $r \in R$. Moreover, they are all semidirect products: $\mathbf{L} = \mathbf{N}_q \rtimes \mathbf{H}_p$, $\mathbf{R} = \mathbf{N}_p \rtimes \mathbf{H}_q$, $\mathbf{P} = \mathbf{N}_p \rtimes \mathbf{H}_p$ and $\mathbf{Q} = \mathbf{N}_q \rtimes \mathbf{H}_q$, where the actions of $\mathbf{H}_p$ and $\mathbf{H}_q$ on $\mathbf{N}_p$ and $\mathbf{N}_q$ are the restrictions of the actions of $\mathbf{H}$ on $\mathbf{N}$.

We will now prove the many-solutions property for $\mathbf{G}$ by restricting the variables to the subgroups $\mathbf{L}$ and $\mathbf{R}$. However, we cannot apply Corollary 12 directly; instead, we will construct another group $\mathbf{G}_1$ with the desired properties for which we can apply Corollary 12 for ListPolSat.

We define $\mathbf{G}_1 = \mathbf{P} \rtimes \mathbf{Q}$ where the action is given by the action of $\mathbf{H}_q \leqslant \mathbf{Q}$ on $\mathbf{N}_p \leqslant \mathbf{P}$ (and $\mathbf{P}$ and $\mathbf{N}_q \leqslant \mathbf{Q}$ commute). Notice that there is a canonical bijection (in general not an isomorphism) between G and G_1 and we have $\mathbf{R} \leqslant \mathbf{G}_1$. Moreover, notice that $\mathbf{G}_1$ meets the requirements of Corollary 12 and $\mathbf{G}/\mathbf{N}_q = (\mathbf{N}_p \rtimes (\mathbf{H}_p \times \mathbf{H}_q) = (\mathbf{N}_p \rtimes \mathbf{H}_p) \rtimes \mathbf{H}_q = \mathbf{G}_1/\mathbf{N}_q$.

Our next step is to transform a polynomial $\mathbf{q} \in \mathrm{Pol}(\mathbf{G})$ to a polynomial $\theta(\mathbf{q}) \in \mathrm{Pol}(\mathbf{G}_1)$ which, when restricting variables to R, has the same solution set (this makes sense as $\mathbf{R}$ is both a subgroup of $\mathbf{G}$ and $\mathbf{G}_1$).

In order to do so, write $\mathbf{q}(\overline{x}) = g_0 h_0 \xi_1 g_1 h_1 \cdots \xi_m g_m h_m$ where $g_i \in N_q$ and $h_i \in H_p$ (i.e., $g_i h_i \in L$) and the ξ_i are constants from $\mathbf{R} = \mathbf{N}_p \rtimes \mathbf{H}_q$ or variables. We define $\theta : \mathrm{Pol}(\mathbf{G}) \to \mathrm{Pol}(\mathbf{G}_1)$ by $\theta(\mathbf{q}) = g_0 h_0 \xi_1 ({}^{h_0}g_1) h_1 \cdots \xi_m ({}^{h_0 \cdots h_{m-1}}g_m) h_m$, where ${}^{h}g = hgh^{-1}$ denotes the action of $h \in H_p$ on $g \in N_q$ (considered in $\mathbf{G}$); we view ${}^{h}g$ as a fixed element of $N_q \leqslant G_1$ and forget that it comes from the action in $\mathbf{G}$. Notice that up to a constant factor $|\theta(\mathbf{q})|$ and $|\mathbf{q}|$ are equal. We say $\overline{x}$ is a solution of $\mathbf{q}$ if $\mathbf{q}(\overline{x}) = 1$.

▷ Claim 23. Let $\overline{x} \in R^n$. Then $\overline{x}$ is a solution of $\mathbf{q}$ if and only if $\overline{x}$ is a solution of $\theta(\mathbf{q})$.

Proof. As $\mathbf{G}/\mathbf{N}_q = \mathbf{G}_1/\mathbf{N}_q$, we may assume that $\mathbf{q}(\overline{x})$ and $\theta(\mathbf{q})(\overline{x})$ are both in N_q. Then, in $\mathbf{G}$ we have

$$\begin{aligned}\mathbf{q}(\overline{x}) &= g_0 h_0 \xi_1 g_1 h_1 \cdots \xi_m g_m h_m \\ &= g_0 \, {}^{h_0\xi_1}g_1 \cdots {}^{h_0\xi_1\cdots h_{m-1}\xi_m}g_m \cdot h_0 \xi_1 \cdots h_{m-1}\xi_m h_m \\ &= g_0 \, {}^{h_0\xi_1}g_1 \cdots {}^{h_0\xi_1\cdots h_{m-1}\xi_m}g_m.\end{aligned}$$

On the other hand, in $\mathbf{G}_1$ we have

$$\begin{aligned}\theta(\mathbf{q})(\overline{x}) &= g_0 h_0 \xi_1 ({}^{h_0}g_1) h_1 \cdots \xi_m ({}^{h_0\cdots h_{m-1}}g_m) h_m \\ &= g_0 \, {}^{\xi_1}({}^{h_0}g_1) \cdots {}^{\xi_1\cdots\xi_m}({}^{h_0\cdots h_{m-1}}g_m) \cdot h_0 \xi_1 \cdots h_{m-1}\xi_m h_m \\ &= g_0 \, {}^{\xi_1}({}^{h_0}g_1) \cdots {}^{\xi_1\cdots\xi_m}({}^{h_0\cdots h_{m-1}}g_m).\end{aligned}$$

Now, we can read the last line as an element of $\mathbf{G}$ interpreting ${}^{h}g = hgh^{-1}$ again as the action of $h \in H_p$ on $g \in N_q$. As in $\mathbf{G}$ the ξ_i commute with h_i modulo $\mathbf{N}_p$, which is contained in the centralizer of $\mathbf{N}_q$ in $\mathbf{G}$, we conclude that as an equality in $\mathbf{G}$ we have

$$\theta(\mathbf{q})(\overline{x}) = g_0 {}^{h_0\xi_1}g_1 \cdots {}^{h_0\xi_1\cdots h_{m-1}\xi_m}g_m.$$

This proves the claim. ◁

If a polynomial $\mathbf{q} \in \mathrm{Pol}(\mathbf{G})$ has a solution with variables restricted to R, by Claim 23, $\theta(\mathbf{q})$ also has a solution with variables restricted to R. Now, we can apply Corollary 12 (in the CDH case) or Corollary 20 (if $\mathbf{H}$ is abelian), which gives us that a polynomial fraction $1/|\theta(\mathbf{q})|^{\mathcal{O}(1)}$ of all assignments $\overline{y} \in R^n$ are satisfying for $\theta(\mathbf{q})$ (i.e., there are at least $|R|^n/|\theta(\mathbf{q})|^{\mathcal{O}(1)}$ satisfying assignments among $|R|^n$ possible assignments). By Claim 23 also at least a polynomial fraction $1/|\mathbf{q}|^{\mathcal{O}(1)}$ of all assignments $\overline{y} \in R^n$ are satisfying for $\mathbf{q}$.

By symmetry the same argument applies to a polynomial with variables restricted to L: if a polynomial $\mathbf{q} \in \mathrm{Pol}(\mathbf{G})$ with variables restricted to L has a solution, at least $1/|\mathbf{q}|^{\mathcal{O}(1)}$ of all assignments $\overline{y} \in L^n$ are satisfying for $\mathbf{q}$.

Since $\mathbf{G} = \mathbf{LR}$, we can show the none-or-many property for $\mathbf{G}$ as follows: assume $\mathbf{p}$ is a polynomial with a solution $\overline{a} = (a_1, \ldots, a_n)$. We can write each $a_i = \ell_i r_i$ with $\ell_i \in L$ and $r_i \in R$. Let $\mathbf{q}$ be the polynomial obtained from $\mathbf{p}$ by substituting every variable x_i by $\ell_i y_i$ where y_i is a new variable. We know that $\mathbf{q}$ has a solution when restricting all variables to R – hence, it has at least $|R|^n/|\mathbf{p}|^{\mathcal{O}(1)}$ solutions in R^n. For each of these solutions $\overline{r}' = (r'_1, \ldots, r'_n) \in R^n$ again we obtain a polynomial $\mathbf{r}$ from $\mathbf{p}$ by replacing each variable x_i by $z_i r'_i$ where z_i is a new variable restricted to L. Now, $\mathbf{r}$ has at least $|L|^n/|\mathbf{p}|^{\mathcal{O}(1)}$ many solutions. Since any of these solutions gives us a solution to $\mathbf{p}$, we obtain at least $|R|^n/|\mathbf{p}|^{\mathcal{O}(1)} \cdot |L|^n/|\mathbf{p}|^{\mathcal{O}(1)}$ solutions for $\mathbf{p}$.

Therefore, picking random assignments leads to an RP algorithm (like in Lemma 10). ◀

A straightforward (though not the smallest) example for Theorem 4 not covered by previous results is the wreath product $\mathbb{Z}_6 \wr \mathbb{Z}_6 = (\mathbb{Z}_6)^6 \rtimes \mathbb{Z}_6$. By Theorem 4 we know that PolSat is in RP for this group, whereas ProgramSat is not in RP under rETH by Theorem 1.

As we show in Corollary 6, we can even classify the complexity of PolSat for arbitrary wreath products. Before we outline the proof, let us remark that, if $\mathbf{G}$ is nilpotent and $\mathbf{H}$ abelian, then PolSat($\mathbf{G} \wr \mathbf{H}$) is in RP as soon as $\mathbf{G}$ is a p-group or $|G|$ and $|H|$ only have the same two prime divisors – without requiring CDH. This is an immediate consequence of Corollary 20 and Theorem 4.

Proof sketch of Corollary 6. The CDH-based RP algorithms are due to Corollary 12 and Theorem 4. Being not in the RP case, $|G|$ has at least two prime divisors $q \neq r$. Moreover, $|H|$ has a third prime divisor $p \neq q, r$. Thus, we will find a wreath product $(\mathbb{Z}_q \times \mathbb{Z}_r) \wr \mathbb{Z}_p$ as a subgroup of a quotient of $\mathbf{G} \wr \mathbf{H}$. By [26, Theorm 4.1.10] neither $\mathbb{Z}_q \wr \mathbb{Z}_p$ nor $\mathbb{Z}_r \wr \mathbb{Z}_p$ is nilpotent. Thus, we can find covering pairs of normal subgroups $\mathbf{B}_q, \mathbf{A}_q$ and $\mathbf{B}_r, \mathbf{A}_r$ such that $C_{\mathbb{Z}_q \wr \mathbb{Z}_p}(\mathbf{B}_q/\mathbf{A}_q) \neq \mathbb{Z}_q \wr \mathbb{Z}_p$. It remains to lift them to $\mathbf{G} \wr \mathbf{H}$ and apply Corollary 3. ◀

8 Conclusion

In this paper, under the assumptions of rETH and CDH, we fully classified in which cases the computational complexity of ProgramSat and ListPolSat for finite groups is in RP. It seems that eliminating the assumptions (especially rETH) can be really hard, but there is still a chance to improve Theorem 1 by showing polynomial time deterministic algorithms instead of the randomized ones:

▶ **Problem 1.** *Is there a polynomial time* deterministic *algorithm solving* ProgramSat($\mathbf{G}$) *and* ListPolSat($\mathbf{G}$) *for* $\mathbf{G}$ *such that there is a prime* p *and a normal* p*-subgroup* $\mathbf{G}_p$ *of* $\mathbf{G}$ *with* $\mathbf{G}/\mathbf{G}_p$ *being nilpotent?*

We took a step towards full classification of the complexity of PolSat for finite groups. Our study reveals that the interactions of normal subgroups of different characteristics play a crucial role. To conclude we present an example of a group of Fitting length 2 for which the complexity of PolSat can not be resolved by our results.

▶ **Problem 2.** *What is the computational complexity of* PolSat($\mathbf{G}$) *for*

$$\mathbf{G} = (\mathbb{Z}_3 \times \mathbb{Z}_5 \times \mathbb{Z}_7) \rtimes (\mathbb{Z}_2 \times \mathbb{Z}_2),$$

where the first $\mathbb{Z}_2$ *acts on* $\mathbb{Z}_3 \times \mathbb{Z}_5$ *by inversion and the second* $\mathbb{Z}_2$ *acts on* $\mathbb{Z}_5 \times \mathbb{Z}_7$ *by inversion?*

Note that the group $\mathbf{G}$ from Problem 2 has $\mathbb{Z}_3$, $\mathbb{Z}_5$, and $\mathbb{Z}_7$ as normal subgroups of different characteristics with $C_{\mathbf{G}}(\mathbb{Z}_p) \neq G$ for $p = 3, 5, 7$ and $C_{\mathbf{G}}(\mathbb{Z}_p) \cdot C_{\mathbf{G}}(\mathbb{Z}_q) = G$ for $p \neq q$. In particular, the last property prevents us from using Theorem 2 or Corollary 3. On the other hand, four different primes dividing the size of $\mathbf{G}$ blocks Theorem 4 from being applied here. Moreover, also Corollary 12 cannot be applied here since the largest nilpotent quotient of $\mathbf{G}$ is $\mathbb{Z}_2 \times \mathbb{Z}_2$ and the kernel of the projection is clearly not a p-group.

References

1. David A. Mix Barrington. Width-3 permutation branching programs. Technical Report TM-293, MIT Laboratory for Computer Science, 1985.
2. David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In *Proceedings of STOC'86*, pages 1–5, 1986. doi:10.1145/12130.12131.
3. David A. Mix Barrington, Richard Beigel, and Steven Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:367–382, 1994. doi:10.1007/BF01263424.
4. David A. Mix Barrington, Pierre McKenzie, Cristopher Moore, Pascal Tesson, and Denis Thérien. Equation satisfiability and program satisfiability for finite monoids. In *Proceedings of MFCS'00*, pages 172–181, 2000. doi:10.1007/3-540-44612-5_13.
5. David A. Mix Barrington, Howard Straubing, and Denis Thérien. Non-uniform automata over groups. *Inf. Comput.*, 89(2):109–132, 1990. doi:10.1016/0890-5401(90)90007-5.
6. David A. Mix Barrington and Denis Thérien. Finite monoids and the fine structure of NC^1. *J. ACM*, 35(4):941–952, 1988. doi:10.1145/48014.63138.
7. Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. In *Proceedings of APPROX/RANDOM'15*, pages 680–709, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.680.
8. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
9. Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Trans. Algorithms*, 10(4):21:1–21:32, 2014. doi:10.1145/2635812.
10. Volker Diekert, Claudio Gutiérrez, and Christian Hagenah. The existential theory of equations with rational constraints in free groups is PSPACE-complete. *Inf. Comput.*, 202(2):105–140, 2005. doi:10.1016/j.ic.2005.04.002.

11 Attila Földvári. The complexity of the equation solvability problem over semipattern groups. *IJAC*, 27(2):259, 2017. doi:10.1142/S0218196717500126.

12 Attila Földvári and Gábor Horváth. The complexity of the equation solvability and equivalence problems over finite groups. *IJAC*, 30(03):607–623, 2020. doi:10.1142/S0218196720500137.

13 Ralph Freese and Ralph McKenzie. *Commutator Theory for Congruence Modular Varieties*. London Mathematical Society Lecture Notes, No. 125. Cambridge University Press, 1987.

14 Mikael Goldmann and Alexander Russell. The complexity of solving equations over finite groups. *Inf. Comput.*, 178(1):253–262, 2002. doi:10.1006/inco.2002.3173.

15 Vince Grolmusz. A degree-decreasing lemma for (MOD_p-MOD_m) circuits. *Discret. Math. Theor. Comput. Sci.*, 4(2):247–254, 2001. doi:10.46298/dmtcs.289.

16 Vince Grolmusz and Gábor Tardos. Lower bounds for (MOD_p-MOD_m) circuits. *SIAM J. Comput.*, 29(4):1209–1222, 2000. doi:10.1137/S0097539798340850.

17 David Hobby and Ralph McKenzie. *Structure of Finite Algebras*. Contemporary Mathematics vol. 76. American Mathematical Society, 1988. doi:10.1090/conm/076.

18 Gábor Horváth. The complexity of the equivalence and equation solvability problems over meta-Abelian groups. *J. Algebra*, 433:208–230, 2015. doi:10.1016/j.jalgebra.2015.03.015.

19 Gábor Horváth and Csaba A. Szabó. The complexity of checking identities over finite groups. *IJAC*, 16(5):931–940, 2006. doi:10.1142/S0218196706003256.

20 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Intermediate problems in modular circuits satisfiability. In *Proceedings of LICS'20*, pages 578–590, 2020. doi:10.1145/3373718.3394780.

21 Pawel M. Idziak, Piotr Kawalek, and Jacek Krzaczkowski. Complexity of modular circuits. In *37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS '22), August 2–5, 2022, Haifa, Israel*. ACM, New York, NY, USA, 2022. doi:10.1145/3531130.3533350.

22 Pawel M. Idziak, Piotr Kawalek, and Jacek Krzaczkowski. Satisfiability of circuits and equations over finite Malcev algebras. In *Proceedings of STACS'22*, pages 37:1–37:14, 2022. doi:10.4230/LIPIcs.STACS.2022.37.

23 Pawel M. Idziak, Piotr Kawalek, Jacek Krzaczkowski, and Armin Weiß. Equation satisfiability in solvable groups. *Theory Comput. Syst.*, to appear, 2022. arXiv:2010.11788.

24 Paweł M. Idziak and Jacek Krzaczkowski. Satisfiability in multi-valued circuits. *SIAM J. Comp.*, 51(3):337–378, 2022. doi:10.1137/18M1220194.

25 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

26 J. D. P. Meldrum. *Wreath products of groups and semigroups*, volume 74 of *Pitman Monographs and Surveys in Pure and Applied Mathematics*. Longman, Harlow, 1995.

27 Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

28 Derek J. S. Robinson. *A course in the theory of groups*, volume 80 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1996. doi:10.1007/978-1-4419-8594-1.

29 Klaus U. Schulz. Makanin's algorithm for word equations – two improvements and a generalization. In *Proceedings of Word Equations and Related Topics, First International Workshop, IWWERT*, pages 85–150, 1990. doi:10.1007/3-540-55124-7_4.

30 Armin Weiß. Hardness of Equations over Finite Solvable Groups Under the Exponential Time Hypothesis. In *Proceedings of ICALP'20*, pages 102:1–102:19, 2020. doi:10.4230/LIPIcs.ICALP.2020.102.