# Constant Depth Circuit Complexity for Generating Quasigroups

NATHANIEL A. COLLINS, Colorado State University, Department of Mathematics, USA
JOSHUA A. GROCHOW, University of Colorado Boulder, Departments of Computer Science and Mathematics, USA
MICHAEL LEVET, College of Charleston, Department of Computer Science, USA
ARMIN WEISS, Universität Stuttgart, FMI, Germany

We investigate the constant-depth circuit complexity of the Isomorphism Problem, Minimum Generating Set Problem (MGS), and Sub(quasi)group Membership Problem (Membership) for groups and quasigroups (=Latin squares), given as input in terms of their multiplication (Cayley) tables. Despite decades of research on these problems, lower bounds for these problems even against depth-2 AC circuits remain unknown. Perhaps surprisingly, Chattopadhyay, Torán, and Wagner (FSTTCS 2010; *ACM Trans. Comput. Theory*, 2013) showed that Quasigroup Isomorphism could be solved by AC circuits of depth $O(\log\log n)$ using $O(\log^2 n)$ nondeterministic bits, a class we denote $\exists^{\log^2 n}\mathsf{FOLL}$. We narrow this gap by improving the upper bound for these problems to $\mathsf{quasiAC}^0$, thus decreasing the depth to constant.

In particular, we show that Membership can be solved in $\mathsf{NTIME}(\mathrm{polylog}(n))$ and use this to prove the following:

- MGS for quasigroups belongs to $\exists^{\log^2 n}\forall^{\log n}\mathsf{NTIME}(\mathrm{polylog}(n)) \subseteq \mathsf{quasiAC}^0$. Papadimitriou and Yannakakis (*J. Comput. Syst. Sci.*, 1996) conjectured that this problem was $\exists^{\log^2 n}$P-complete; our results refute a version of that conjecture for completeness under $\mathsf{quasiAC}^0$ reductions unconditionally, and under polylog-space reductions assuming EXP ≠ PSPACE.
  It furthermore implies that this problem is not hard for any class containing Parity. The analogous results concerning Parity were known for Quasigroup Isomorphism (Chattopadhyay, Torán, & Wagner, *ibid.*) and Membership for groups (Fleischer, *Theory Comput.* 2022), though not for MGS.
- MGS for groups belongs to $\mathsf{AC}^1(\mathsf{L})$. Our $\mathsf{AC}^1(\mathsf{L})$ bound improves on the previous, very recent, upper bound of P (Lucchini & Thakkar, *J. Algebra*, 2024). Our $\mathsf{quasiAC}^0$ upper bound is incomparable to P, but has similar consequences to the above result for quasigroups.
- Quasigroup Isomorphism $\in \exists^{\log^2 n}\mathsf{AC}^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)))$, which is contained in $\mathsf{quasiAC}^0$. As a consequence of this result and previously known $\mathsf{AC}^0$ reductions, this implies the same upper bound for the Isomorphism Problems for: Steiner triple systems, pseudo-STS graphs, Latin square graphs, and Steiner $(t, t+1)$-designs. This improves upon the previous upper bound for these problems, which was $\exists^{\log^2 n}\mathsf{L} \cap \exists^{\log^2 n}\mathsf{FOLL} \subseteq \mathsf{quasiFOLL}$ (Chattopadhyay, Torán, & Wagner, *ibid.*; Levet, *Australas. J. Combin.* 2023).
- As a strong contrast, we show that MGS for arbitrary magmas is NP-complete.

Our results suggest that understanding the constant-depth circuit complexity may be key to resolving the complexity of problems concerning (quasi)groups in the multiplication table model.

CCS Concepts: • **Theory of computation → Circuit complexity**; **Algebraic complexity theory**; • **Mathematics of computing → Discrete mathematics**.

Additional Key Words and Phrases: Group Isomorphism, Quasigroup Isomorphism, Minimum Generating Set, Membership Testing, Constant-Depth Circuits, quasiAC0, Circuit Complexity

**ACM Reference Format:**
Nathaniel A. Collins, Joshua A. Grochow, Michael Levet, and Armin Weiß. 2024. Constant Depth Circuit Complexity for Generating Quasigroups. *ACM Trans. Graph.* 37, 4, Article 111 (August 2024), 9 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

[1]The Group Isomorphism (GpI) problem is a central problem in computational complexity and computer algebra. When the groups are given as input by their multiplication (a.k.a. Cayley) tables, the problem reduces to Graph Isomorphism (GI), and because the best-known runtimes for the two are quite close ($n^{O(\log n)}$ [29][2] vs. $n^{O(\log^2 n)}$ [3][3]), the former stands as a key bottleneck towards further improvements in the latter.

Despite this, GpI seems quite a bit easier than GI. For example, Tarjan's $n^{\log n + O(1)}$ algorithm for groups [29] can now be given as an exercise to undergraduates: every group is generated by at most $\lceil \log_2 |G| \rceil$ elements, so the algorithm is to try all possible $\binom{n}{\log n} \leq n^{\log n}$ generating sets, and for each, check in $n^{O(1)}$ time whether the map of generating sets extends to an isomorphism. In contrast, the quasipolynomial time algorithm for graphs was a tour de force that built on decades of cutting-edge research into algorithms and the structure of permutation groups. Nonetheless, it remains unknown whether the problem for groups is actually easier than that for graphs, or even whether both problems are in P!

Using a finer notion of reduction, Chattopadhyay, Torán, and Wagner [10] proved that there was no $\mathsf{AC}^0$ reduction from GI to GpI. This gave the first (and still only known) unconditional evidence that there is *some formal sense* (namely, the $\mathsf{AC}^0$ sense) in which GpI really is easier than GI. The key to their result was that the generator-enumeration technique described above can be implemented by non-deterministically guessing $\log^2 n$ bits (for the $\log n$ generators,

Authors' addresses: Nathaniel A. Collins, naco3124@colostate.edu, Colorado State University, Department of Mathematics, Fort Collins, Colorado, USA; Joshua A. Grochow, jgrochow@colorado.edu, University of Colorado Boulder, Departments of Computer Science and Mathematics, Boulder, Colorado, USA; Michael Levet, levetm@cofc.edu, College of Charleston, Department of Computer Science, Charleston, SC, USA; Armin Weiß, armin.weiss@fmi.uni-stuttgart.de, Universität Stuttgart, FMI, Germany.


0730-0301/2024/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

[1]Some proofs are shortened, sketched, or omitted in this conference submission to save space; all such proofs are given in full detail in the preprint of the full version [12].
[2]Miller [29] credits Tarjan for $n^{\log n + O(1)}$.
[3]Babai [3] proved quasipolynomial time, and the exponent of the exponent was analyzed and improved by Helfgott [20]

each of $\log n$ bits), and then verifying an isomorphism by a circuit of depth only $O(\log \log n)$, a class we denote $\exists^{\log^2 n}\mathsf{FOLL}$. Note that $\exists^{\log^2 n}\mathsf{FOLL} \subseteq \mathsf{quasiFOLL}$, which cannot compute Parity [10; 32; 37]. As GI is DET-hard [40]—and hence can compute Parity—there can be no $\mathsf{AC}^0$ reduction from GI to GpI.

Such a low-depth circuit was quite surprising, although that surprise is perhaps tempered by the use of non-determinism. Nonetheless, it raises the question:

Is it possible that Group Isomorphism is in $\mathsf{AC}^0$?

The authors would be shocked if the answer were "yes," and yet we do not even have results showing that Group Isomorphism cannot be computed by polynomial-size circuits of (!) depth 2. The upper bound of $\exists^{\log^2 n}\mathsf{FOLL}$ rules out most existing lower bound techniques against $\mathsf{AC}^0$, as most such techniques also yield similar lower bounds against $\exists^{\log^2 n}\mathsf{FOLL}$.

In this paper, we aim to close the gap between $\mathsf{AC}^0$ and $\exists^{\log^2 n}\mathsf{FOLL}$ in the complexity of Group Isomorphism and related problems. Our goal is to obtain constant-depth circuits of quasipolynomial size, a natural benchmark in circuit complexity [6]. Getting such circuits of polynomial size would resolve the long-standing question of putting these problems into P; in contrast, our current target is to reduce their *parallel complexity*. Our first main result along these lines is:

**Theorem A.** *(Quasi)Group Isomorphism is in* $\mathsf{quasiAC}^0$.

(We discuss quasigroups in more detail below.) The prior best depth bounds on the general problem were all super-constant (with quasipolynomial size): $\exists^{\log^2 n}\mathsf{SC}^2$ by Tang [38] yields (by a standard simulation argument) circuits of depth $\log^2 n$, while $\exists^{\log^2 n}\mathsf{L} \cap \exists^{\log^2 n}\mathsf{FOLL}$ [10] has depth $\log \log n$. In particular, prior to our work, Quasigroup Isomorphism was not known to be solvable using quasiAC circuits of depth $o(\log \log n)$. The special case of Abelian GpI was previously known to belong to $\mathsf{quasiAC}^0$ [23].

**Remark 1.1.** We in fact get a more precise bound of

$$\exists^{\log^2 n}\forall^{\log n}\exists^{\log n}\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)).$$

This more precise bound is notable because it is contained in $\mathsf{quasiAC}^0$ and $\exists^{\log^2 n}\mathsf{FOLL} \cap \exists^{\log^2 n}\mathsf{L}$, thus improving on [10]. We get similarly precise bounds with complicated-looking complexity classes for the other problems we study, but we omit the precise bounds in the introduction for readability.

*Minimum generating set.* Another very natural problem in computational algebra is the Min Generating Set (MGS) problem. Given a group, this problem asks to find a generating set of the smallest possible size. Given that many algorithms on groups depend on the size of a generating set, finding a minimum generating set has the potential to be a widely applicable subroutine. The MGS problem for groups was shown to be in P by Lucchini & Thakkar very recently [28]. We improve their complexity bound:

**Theorem B.** *MGS for groups can be solved in* $\mathsf{quasiAC}^0$ *and in* $\mathsf{AC}^1(\mathsf{L})$ *($O(\log n)$-depth, unbounded fan-in circuits with a logspace oracle).*

We note that, although $\mathsf{quasiAC}^0$ is incomparable to P because of the quasipolynomial size (whereas $\mathsf{AC}^1(\mathsf{L}) \subseteq \mathsf{P}$), the key we are focusing on here is reducing the depth. For nilpotent groups (widely believed to be the hardest cases of GpI), if we only wish to compute the minimum *number* of generators, we can further improve this complexity to a subclass of $\mathsf{L} \cap \mathsf{FOLL} \cap \mathsf{quasiAC}^0$ (Prop. 7.3).

While our $\mathsf{AC}^1(\mathsf{L})$ bound above is essentially a careful complexity analysis of the polynomial-time algorithm of Lucchini & Thakkar [28], the $\mathsf{quasiAC}^0$ upper bound is in fact a consequence of our next, more general result for *quasi*groups, which involves some new ingredients.

*Enter quasigroups.* Quasigroups can be defined in (at least) two equivalent ways: (1) an algebra whose multiplication table is a Latin square,[4] or (2) a group-like algebra that need not have an identity nor be associative, but in which left and right division are uniquely defined, that is, for all $a, b$, there are unique $x$ and $y$ such that $ax = b$ and $ya = b$, denoted $x = a\backslash b$ and $y = b/a$.

In the paper in which they introduced $\log^2(n)$-bounded nondeterminism, Papadimitriou and Yannakakis showed that for arbitrary magmas,[5] testing whether the magma has $\log n$ generators was in fact *complete* for $\exists^{\log^2 n}\mathsf{P}$, and conjectured:

**Conjecture 1.2** (Papadimitriou & Yannakakis [31, p. 169])**.** *MGS for quasigroups is* $\exists^{\log^2 n}\mathsf{P}$*-complete.*

They explicitly did *not* conjecture the same for MGS for *groups*, writing:

> "We conjecture that this result [$\exists^{\log^2 n}\mathsf{P}$-completeness] also holds for the more structured MINIMUM GENERATOR SET OF A QUASIGROUP problem. In contrast, QUASIGROUP ISOMORPHISM was recently shown to be in $\mathrm{DSPACE}(\log^2 n)$ [43]. Notice that the corresponding problems for groups were known to be in $\mathrm{DSPACE}(\log^2 n)$ [26]."—Papadimitriou & Yannakakis [31, p. 169]

We thus turn our attention to the analogous problems for quasigroups: MGS, Isomorphism, and the key subroutine, Membership.

**Theorem C.** *MGS for quasigroups is in* $\mathsf{quasiAC}^0 \cap \mathrm{DSPACE}(\log^2 n)$.

To the best of our knowledge, MGS for Quasigroups has not been studied from the complexity-theoretic viewpoint previously. While a $\mathrm{DSPACE}(\log^2 n)$ upper bound for MGS for *groups* follows from [1; 38], as far as we know it remained open for quasigroups prior to our work.

As with prior results on (Quasi)group Isomorphism [10] and other isomorphism problems (e.g., [25]), Thm. C shows that Parity does not reduce to MGS for Quasigroups, thus ruling out most known lower bound methods that might be used to prove that MGS for Quasigroups is not in $\mathsf{AC}^0$. We observe a similar bound for MGS for Groups using Fleischer's technique [17].

Papadimitriou & Yannakakis did not specify the type of reduction used in their conjecture, though their $\exists^{\log^2 n}\mathsf{P}$-completeness result for Log Generating Set for magmas works in both logspace and

[4] A Latin square is an $n \times n$ matrix where for each row and each column, the elements of $[n]$ appear exactly once

[5] A magma is a set $M$ together with a function $M \times M \to M$ that need not satisfy any additional axioms.

$AC^0$ (under a suitable input encoding). Our two upper bounds rule out such reductions for MGS for Quasigroups:

**Corollary C.** *Conj. 1.2 is false under* quasi$AC^0$ *reductions. It is also false under polylog-space reductions assuming* EXP ≠ PSPACE.

In strong contrast, we show that MGS for Magmas is NP-complete (Thm. 7.10).

A key ingredient in our proof of Thm. C is an improvement in the complexity of another central problem in computational algebra: the Sub-quasigroup Membership problem (Membership,[6] for short):

**Theorem D.** *Membership for quasigroups is in* $\mathsf{NTIME}(\mathrm{polylog}(n)) \subseteq$ quasi$AC^0$.

Membership for *groups* is well-known to belong to L, by reducing to the connectivity problem on the Cayley graph (cf. [8; 33]), but as L sits in between $AC^0$ and $AC^1$, this is not low enough depth for us.

*Additional results.* We also obtain a number of additional new results on related problems, some of which we highlight here:

- By known $AC^0$ reductions (see, e.g., [25]), our quasi$AC^0$ analysis of Chattopadhyay, Torán, and Wagner's algorithm for Quasigroup Isomorphism yields the same upper bound for the isomorphism problems for Steiner triple systems, pseudo-STS graphs, Latin square graphs, and Steiner $(t, t+1)$-designs, as well as Latin Square Isotopy.
- GpI for groups from a dense set of orders can be solved in a subclass of quasi$AC^0 \cap$ FOLL $\cap$ L. This improves the parallel complexity compared to the original result [15].
- Abelian Group Isomorphism is in
$$\forall^{\log\log n}\mathsf{MAC}^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n))).$$
The key novelties here are (1) a new observation that allows us to reduce the number of co-nondeterministic bits from $\log n$ (as in [18]) down to $\log\log n$, and (2) using an $AC^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)))$ circuit for order finding, rather than FOLL as in [10].
- Membership for nilpotent groups is in $\mathsf{NTISP}(\mathrm{polylog}(n), \log(n))$, which is contained in FOLL $\cap$ quasi$AC^0$.

## 1.1 Methods

Several of our results involve careful analysis of the low-level circuit complexity of extant algorithms, showing that they in fact lie in smaller complexity classes than previously known. We crucially use simultaneous time- and space-restricted computations. This not only facilitates several proofs and gives better complexity bounds, but also gives rise to new algorithms such as for Membership for nilpotent groups, which previously was not known to be in FOLL.

One such instance is in our improved bound for order-finding and exponentiation in a semigroup (Lem. 3.1). The previous proof [7] (still state of the art 23 years later) used a then-novel and clever "double-barrelled" recursive approach to compute these in FOLL. In contrast, our proof uses standard repeated doubling, noting that it can be done in $\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)) \subseteq$ FOLL $\cap$ quasi$AC^0$, recovering their result with standard tools and reducing the depth.

We use this improved bound on order-finding to improve the complexity of isomorphism testing of Abelian groups (Thm. 4.1), simple groups (Cor. 3.3), and groups of almost all orders (Thm. 5.1).

For several results we additionally need to develop new tools to work with quasigroups. Notably, for the quasi$AC^0$ upper bound on MGS for quasigroups, we cannot directly adapt the technique of [10], as the existence alone of cube generating sets seems insufficient for Membership and MGS. The first key is Thm. D, putting Membership for quasigroups into $\mathsf{NTIME}(\mathrm{polylog}(n))$. Here, we avoid their use of cube generating sets and instead utilize straight-line programs (SLPs) directly. To show short SLPs exist we extend the Babai–Szemerédi Reachability Lemma [4, Thm. 3.1] from groups (its original setting) to quasigroups. As division in quasigroups is nuanced due to lack of associativity, our proof is a careful adaptation of the technique of [4, Thm. 3.1], with a few quasigroup twists.

## 1.2 Prior work

*Isomorphism testing.* The best known runtime bound for GpI is $n^{(1/4)\log_p(n)+O(1)}$ [35] (see [24, Sec. 2.2]), though this tells us little about parallel complexity. Lipton, Snyder, & Zalcstein [26] independently observed the generator-enumeration procedure and used it to give a bound of $\mathsf{DSPACE}(\log^2 n)$. Miller [29] extended Tarjan's result to quasigroups. There has been subsequent work on improving the parallel complexity of generator enumeration for quasigroups, resulting in bounds of $\exists^{\log^2 n}AC^1$ [43][7], $\exists^{\log^2 n}SAC^1$ [42], and $\exists^{\log^2 n}\mathsf{L} \cap \exists^{\log^2 n}$FOLL [10]. For groups, generator enumeration is also known to belong to $\exists^{\log^2 n}SC^2$ [38]. There has been considerable work on polynomial-time isomorphism tests for several families of groups, and more recent work on NC isomorphism tests—see [15; 18; 19] for a survey. We are not aware of work on isomorphism testing for specific families of quasigroups that are not groups.

*Min Generating Set.* As every (quasi)group has a generating set of size $\leq \lceil \log n \rceil$, MGS admits an $n^{\log(n)+O(1)}$-time solution for (quasi)groups. Arvind & Torán [1] improved the complexity to $\mathsf{DSPACE}(\log^2 n)$ for groups. They also gave a polynomial-time algorithm in the special case of nilpotent groups. Tang further improved the general bound for MGS for groups to $\exists^{\log^2 n}SC^2$ [38]. We observe that Wolf's technique for placing Quasigroup Isomorphism into $\mathsf{DSPACE}(\log^2 n)$ also suffices to get MGS for quasigroups into the same class. Recently, MGS for groups was placed into P [28]. Prior to [28], MGS for groups was considered comparable to Group Isomorphism in terms of difficulty [13]. Our $AC^1(\mathsf{L})$ bound (Thm. C) further closes the gap between Membership and MGS for groups, and in particular suggests that MGS is of comparable difficulty to Membership rather than GpI. Note that Membership is known to belong to L [8; 33].

[6] In the literature, the analogous problem for groups is sometimes called Cayley Group Membership or CGM, to highlight that it is in the Cayley table model.

[7] Wolf actually claims a bound of $\exists^{\log^2 n}NC^2$; however, he uses $NC^1$ circuits to multiply two elements of a quasigroup rather than $AC^0$ circuits.

## 2 ADDITIONAL PRELIMINARIES

### 2.1 Algebra

A *magma* $M$ is an algebraic structure together with a binary operation $\cdot : M \times M \to M$. We will frequently consider subclasses of finite magmas, such as groups, quasigroups, and semigroups.

**Quasigroups**. As quasigroups are non-associative, the parenthesization of a given expression may impact the resulting value. For a sequence $S := (s_0, s_1, \dots, s_k)$ and parenthesization $P$ from a quasigroup $G$, define: $\text{Cube}(S) = \{P(s_0 s_1^{e_1} \cdots s_k^{e_k}) : e_1, \dots, e_k \in \{0,1\}\}$. We say that $S$ is a *cube generating sequence* if each element $g \in G$ can be written as $g = P(s_0 s_1^{e_1} \cdots s_k^{e_k})$, for $e_1, \dots, e_k \in \{0,1\}$. Here, $s_i^0$ indicates that $s_i$ is not being considered in the product. For every parenthesization, every quasigroup admits a cube generating sequence of size $O(\log n)$ [10].

**Group Theory**. For a standard reference, see [34]. A *chief series* of $G$ is an ascending chain $(N_i)_{i=0}^k$ of normal subgroups of $G$, where $N_0 = 1$, $N_k = G$, and each $N_{i+1}/N_i$ $(i = 0, \dots, k-1)$ is minimal normal in $G/N_i$.

**Algorithmic Problems.** We will consider the following algorithmic problems, where the magmas are given by their multiplication tables. The Quasigroup Isomorphism problem takes as input two quasigroups $Q_1, Q_2$ and asks if there is an isomorphism $\varphi : Q_1 \cong Q_2$. The Membership problem for groups takes as input a group $G$, a set $S \subseteq G$, and an element $x \in G$, and asks if $x \in \langle S \rangle$ (the subgroup generated by $S$). We define the Membership problem analogously when the input is a semigroup or quasigroup, and $\langle S \rangle$ is considered as the sub-semigroup or sub-quasigroup, respectively. The Minimum Generating Set (MGS) problem takes as input a magma $M$ and asks for a generating set $S \subseteq M$ where $|S|$ is minimum. The *decision variant* of MGS additionally takes an integer $k$ in the input and ask whether there exists a generating set of size at most $k$.

### 2.2 Computational Complexity

We assume familiarity with standard complexity classes such as L, NL, NP, and EXP. For a standard reference on circuit complexity, see [41]. We consider Boolean circuits using AND, OR, NOT, and Majority, where $\text{Majority}(x_1, \dots, x_n) = 1$ iff $\geq n/2$ of the inputs are 1. All our polynomial-size circuit families are DLOGTIME-uniform, see [12, §2.2] for details.

A language $L$ belongs to $\mathsf{NC}^k$ if there is a family of circuits $C_n$ with NOT gates, and 2-input AND and OR gates, of depth $O(\log^k n)$ and size $n^{O(1)}$, such that $x \in L \Leftrightarrow C_{|x|}(x) = 1$. The class $\mathsf{SAC}^k$ is defined analogously, but the OR gates may have unbounded fan-in, while the AND gates still must have fan-in 2. For $\mathsf{AC}^k$ also the AND gates are permitted to have unbounded fan-in. The complexity class $\mathsf{TC}^k$ is defined analogously as $\mathsf{AC}^k$, except that our circuits are now also permitted Majority gates of unbounded fan-in.

$$\mathsf{AC}^0 \subsetneq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{SAC}^1 \subseteq \mathsf{AC}^1 \subseteq \dots.$$

For a language $L$ the class $\mathsf{AC}^k(L)$, apart from Boolean gates, also allows oracle gates for $L$.[8] If $K \in \mathsf{AC}^k(L)$, then $K$ is said to be $\mathsf{AC}^k$ Turing reducible to $L$. For a complexity class $C$ denote $\mathsf{AC}^k(C)$ to be the set of decision problems that are $\mathsf{AC}^k$-Turing reducible to problems in $C$.[9]

**Further circuit classes.** The complexity class $\mathsf{MAC}^0$ is the set of languages decidable by $\mathsf{TC}^0$ circuits with only a single Majority gate that must be the output gate. $\mathsf{MAC}^0$ was introduced (but not so named) in [2], where it was shown that $\mathsf{MAC}^0 \subsetneq \mathsf{TC}^0$, and later given the name $\mathsf{MAC}^0$ in [21].

The complexity class FOLL is the set of languages decidable by uniform AC circuit families of depth $O(\log \log n)$ and polynomial size. It is known that $\mathsf{AC}^0 \subsetneq \mathsf{FOLL} \subsetneq \mathsf{AC}^1$, and it is open as to whether FOLL is contained in NL [7].

We will be particularly interested in AC circuits of quasipolynomial size. For a circuit class $C \subseteq \mathsf{AC}$, the analogous class permitting a quasipolynomial number of gates is denoted quasi$C$. We will focus specifically on $\mathsf{quasiAC}^0$. Note that Parity $\notin \mathsf{quasiFOLL}$ [32; 37].

**Bounded nondeterminism.** For a complexity class $C$, define $\exists^{f(n)} C$ (resp, $\forall^{f(n)} C$) to be the set of languages $L$ such that there exists an $L' \in C$ such that $x \in L$ iff there exists (resp., for all) $y$ of length at most $O(f(|x|))$ such that $(x, y) \in L'$. For any $i \geq 0$, $\exists^{\log^i n}\mathsf{FOLL} \cup \forall^{\log^i n}\mathsf{FOLL} \subseteq \mathsf{quasiFOLL}$, and so cannot compute Parity [10; 37]. Note that $\forall^{\log n} C \cup \exists^{\log n} C \subseteq \mathsf{AC}^0(C)$.

**Time and space-restricted Turing machines.** For complexity classes defined by Turing machines with a time bound $t(n) \in o(n)$, we use Turing machine with random access and a separate address (or *index*) tape. After writing an address, the machine can go to a query state reading the symbol from the input at the location specified by the address tape.

For functions $t(n), s(n) \in \Omega(\log n)$, the classes $\mathsf{DTISP}(t(n), s(n))$ and $\mathsf{NTISP}(t(n), s(n))$ are defined by deterministic (resp. nondeterministic) $t(n)$ time and $s(n)$ space bounded Turing machines. There must be one Turing machine that simultaneously satisfies the time and space bound. See [41, §2.6] for details. For connections to $\mathsf{quasiAC}^0$, see [6; 16].

**Fact 2.1.** $\mathsf{NTISP}(\text{polylog}(n), \log(n)) \subseteq \mathsf{NTIME}(\text{polylog}(n)) \subseteq \mathsf{quasiAC}^0$.

Proof sketch. Take the OR over all $2^{\text{polylog}(n)}$ possible computation histories, of the $\mathsf{AC}^0$ circuit that verifies a computation history (the latter as in the proof of the Cook–Levin Theorem). □

**Lemma 2.2.** $\mathsf{NTISP}(\text{polylog}(n), \log(n)) \subseteq \mathsf{FOLL}$.

Proof sketch. Follows the proof of Savitch's Theorem. The $\log n$ space bound implies the configuration space has only $n^{O(1)}$ vertices, while the polylog$(n)$ time bound means one need only find a path of length polylog$(n)$, which can be done by Savitch's "repeated doubling" technique in recursion depth $O(\log \log n)$. □

By the very definition we have $\mathsf{DTISP}(\text{polylog}(n), \log(n)) \subseteq \mathsf{L}$ and $\mathsf{NTISP}(\text{polylog}(n), \log(n)) \subseteq \mathsf{NL}$. Thus, we obtain

- $\mathsf{AC}^0(\mathsf{DTISP}(\text{polylog}(n), \log(n))) \subseteq \mathsf{L} \cap \mathsf{FOLL} \cap \mathsf{quasiAC}^0$ and
- $\mathsf{AC}^0(\mathsf{NTISP}(\text{polylog}(n), \log(n))) \subseteq \mathsf{NL} \cap \mathsf{FOLL} \cap \mathsf{quasiAC}^0$.

[8] An oracle gate outputs 1 if and only if its input is in $L$.

[9] Be aware that here we follow the notation of [41], which is different from [18; 42] (where $\mathsf{AC}^k(C)$ is used to denote composition of functions).

## 3 ORDER FINDING AND APPLICATIONS

In this section, we improve the parallel complexity of order finding with an easier and more general proof than [7]. We use this in several applications in this section, §4, and §5. The key lemma is:

**Lemma 3.1.** *The following problem is in* $\mathsf{DTISP}(\mathrm{polylog}(n), \log(n))$*: On input of a multiplication table of a semi-group $S$, an element $s \in S$, and a unary or binary number $k \in \mathbb{N}$ with $k \leq |S|$, compute $s^k$.*

Proof. If $k$ is given in unary, we first compute its binary representation using a binary search (we can write it on the work tape as is uses at most $\lceil \log |S| \rceil$ bits). We identify the semigroup elements with the natural numbers $0, \ldots, |S| - 1$. Now, compute $s^k$ using the standard fast exponentiation algorithm. Multiplying two semigroup elements can be done in $\mathsf{DTIME}(\log n)$ as computing the address of their product only involves multiplying two $\log n$-bit addresses. Finally, it is well-known that the fast exponentiation algorithm needs only $O(\log k)$ algebra multiplications and $O(\log k + \log n)$ space. □

**Corollary 3.2.** *On input of a group $G$, an element $g \in G$, and $k \in \mathbb{N}$, we may decide whether* $\mathrm{ord}(g) = k$ *in* $\forall^{\log n}\mathsf{DTISP}(\mathrm{polylog}(n), \log(n))$.

**Application to isomorphism testing.** Using Cor. 3.2, we can improve the upper bound for isomorphism testing of finite simple groups. Previously, this problem was known to be in L [38] and FOLL [18].

**Corollary 3.3.** *Let $G$ be a finite simple group and $H$ be arbitrary. We can decide whether $G \cong H$ in* $\mathsf{AC}^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)))$.

Cor. 3.2 also lets us improve on the $\mathsf{L} \cap \mathsf{FOLL}$ bound [7, Thm. 3.2] for testing whether a group is nilpotent to:

**Corollary 3.4.** *Deciding whether a group $G$ (given by its multiplication table) is nilpotent is in* $\mathsf{AC}^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)))$.

**Application to membership testing.** A group $G$ has the $\log n$ *power basis property* [7; 17] if for every $X \subseteq G$ every $g \in \langle X \rangle$ can be written as $g = g_1^{e_1} \cdots g_m^{e_m}$ with $m \leq \log n$ and suitable $g_i \in X$ and $e_i \in \mathbb{Z}$. Lem. 3.1 provides that Membership for semigroups with the $\log n$ power basis property is in $\mathsf{NTISP}(\mathrm{polylog}(n), \log(n))$. This observation allows us to improve some FOLL bounds from [7; 17] to $\mathsf{NTISP}(\mathrm{polylog}(n), \log(n))$.

**Corollary 3.5.** *Membership is in* $\mathsf{NTISP}(\mathrm{polylog}(n), \log(n))$ *for:*

*(a) commutative semigroups;*
*(b) nilpotent groups;*
*(c) solvable groups of class $O(1)$.*

The above result for nilpotent groups involves no restriction on the nilpotency class, improving considerably over [7, Cor. 3.2].

Proof sketch for nilpotent groups. Given $g \in G, X \subseteq G$. The $m$-th term of the lower central series $\Gamma_0 = \langle X \rangle, \Gamma_{m+1} = [\Gamma_m, \langle X \rangle]$ is generated by $C_m = \{[[[x_1, x_2], x_3], \cdots, x_k] : x_i \in X, k \geq m\}$ (e. g., [11, Lem. 2.6]). Define $C'$ by choosing a minimal generating set for $\Gamma_0/\Gamma_1$ from among $C_0$, then a minimal generating set of $\Gamma_1/\Gamma_2$ among $C_1$, and so on. $C'$ will be a so-called polycyclic generating set of $\langle X \rangle$, so every $g \in \langle X \rangle$ can be written $g = c_1^{e_1} \cdots c_m^{e_m}$ with $e_i \in \mathbb{Z}$ and $c_i \in C'$. We nondeterministically guess the latter expression for $g$. As $m \leq \log |G|$ for any nilpotent group, this gives an $\mathsf{NTISP}(\mathrm{polylog}(n), \log(n))$ algorithm. □

## 4 ABELIAN GROUP ISOMORPHISM

Our next application of our order-finding Lem. 3.1 is:

**Theorem 4.1.** *Let $G$ be an Abelian group, and let $H$ be arbitrary. We can decide isomorphism between $G$ and $H$ in*

$$\forall^{\log\log n}\mathsf{MAC}^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n))).$$

Chattopadhyay, Torán, and Wagner [10] established a $\mathsf{TC}^0(\mathsf{FOLL})$ upper bound on this problem. Grochow & Levet [18, Thm. 5] gave a tighter analysis of their algorithm, placing it in the sub-class $\forall^{\log n}\mathsf{MAC}^0(\mathsf{FOLL})$.[10] Chattopadhyay, Torán, & Wagner also established an upper bound of L for this problem, which is incomparable to the result of Grochow & Levet (*ibid.*). We improve upon both these bounds by (i) showing that $O(\log\log n)$ non-deterministic bits suffice instead of $O(\log n)$ bits, and (ii) using our improved bound on order-finding (Lem. 3.1).

While $\forall^{\log n}\mathsf{MAC}^0(\mathsf{FOLL})$ is contained in $\mathsf{TC}^0(\mathsf{FOLL})$, it is open whether this containment is strict. In contrast, it follows from [9, Thm. 7] that our new bound is a class that is in fact *strictly* contained in $\mathsf{L} \cap \mathsf{TC}^0(\mathsf{FOLL})$ (see [12, Cor. 5.3] for details).

Proof of Thm. 4.1. Following the strategy of [18, Theorem 7.15], we show that *non*-isomorphism can be decided in the same class but with existentially quantified non-deterministic bits.

We may check in $\mathsf{AC}^0$ whether a group is Abelian. So now assume that $H$ is Abelian. If $G \ncong H$, then there exists a prime power $p^e$ such that there are more elements of order $p^e$ in $G$ than in $H$. We first identify the order of each element, which is $\mathsf{AC}^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)))$-computable by Cor. 3.2.

Let $n = p_1^{e_1} \cdots p_\ell^{e_\ell}$ be the prime factorization of $n$. Using $O(\log\log n)$ non-deterministic bits, we can guess a pair $(p, e)$ ($p$ represented by its index in the prime decomposition), where $G$ has more elements of $p^e$ than $H$. We may then, in $\mathsf{AC}^0$, compute $p^e$ from $p$. We use Cor. 3.2 to identify the elements of order $p^e$, and then an $\mathsf{MAC}^0$ circuit to compare the number of such elements in $G$ vs. $H$. See [12, Thm. 5.1] for full details. □

## 5 GROUP ISOMORPHISM FOR ALMOST ALL ORDERS

We use our improved order-finding Lem. 3.1 to improve the parallel complexity of GpI for almost all orders. Dietrich & Wilson [15] proved that there is a dense set $\Upsilon \subseteq \mathbb{N}$ such that if $n \in \Upsilon$ and $G_1, G_2$ are magmas of order $n$ given by their multiplication tables, we can (i) decide if $G_1, G_2$ are groups, and (ii) if so, decide whether $G_1 \cong G_2$ in time $O(n^2 \log^2 n)$, which is quasi-linear time relative to the input size. We improve the parallel complexity of their result:

**Theorem 5.1.** *Let $n \in \Upsilon$, and let $G_1, G_2$ be magmas of order $n$. We can decide whether the $G_i$ are groups and whether $G_1 \cong G_2$ in* $\mathsf{AC}^0(\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)))$.

Proof. Deciding whether a magma is a group is readily seen to be in $\mathsf{AC}^0$. Dietrich & Wilson showed [15, Thm. 2.5] that if $G$ is a group of order $n \in \Upsilon$, then $G = H \ltimes B$, where:

[10] Grochow & Levet consider $\forall^{\log n}\mathsf{MAC}^0 \circ \mathsf{FOLL}$, where $\circ$ denotes composition (see [18] for a precise formulation). Note that as $\mathsf{AC}^0 \circ \mathsf{FOLL} = \mathsf{FOLL} = \mathsf{AC}^0(\mathsf{FOLL})$, we have $\forall^{\log n}\mathsf{MAC}^0 \circ \mathsf{FOLL} = \forall^{\log n}\mathsf{MAC}^0(\mathsf{FOLL})$. Thus, Thm. 4.1 improves upon the previous bound of $\forall^{\log n}\mathsf{MAC}^0(\mathsf{FOLL})$ obtained by Grochow & Levet.

- $B$ is a cyclic group of order $p_1 \cdots p_\ell$, where for each $i \in [\ell]$, $p_i > \log\log n$ and $p_i$ is the maximum power of $p_i$ dividing $n$.
- $|H| = (\log n)^{\text{poly}\log\log n}$; and in particular, if a prime divisor $p$ of $n$ satisfies $p \leq \log\log n$, then $p$ divides $|H|$.

Fix groups $G_1, G_2$ of order $n$. We may, in $\mathsf{AC}^0(\mathsf{DTISP}(\text{polylog}(n), \log(n)))$, do the following: (i) decompose $G_i = H_i \ltimes B_i$, as prescribed by [15, Thm. 2.5], (ii) select a generator $b_i$ for $B_i$, and (iii) select an isomorphism for $H_1 \cong H_2$ by specifying cube generating sequences (or decide if $H_1 \ncong H_2$). Step (iii) utilizes the proof of [12, Thm. 4.1] in the full version. As $|H_1| = |H_2| \leq (\log n)^{\text{poly}\log\log n}$, there are only poly$(n)$ such generating sequences to consider.

Suppose $H_1 \cong H_2$, $B_1 \cong B_2$, and $\gcd(|B_j|, |H_j|) = 1$ for $j = 1, 2$. We have by the Schur–Zassenhaus Theorem that $G_j = H_j \ltimes_{\theta_j} B_j$ ($j = 1, 2$). By Taunt's Lemma [39], it remains to test whether the actions $\theta_1$ and $\theta_2$ are equivalent. This step is computable in $\mathsf{AC}^0(\mathsf{DTISP}(\text{polylog}(n), \log(n)))$; see [12, Thm. 6.1] for details. □

## 6 QUASIGROUP ISOMORPHISM

The main results in the remainder of the paper do not rely on our order-finding Lem. 3.1, but rather on other techniques. The following proof is inspired by [14] where a similar problem is shown to be in the third level of the polynomial hierarchy using the same approach.

**Theorem 6.1.** *Quasigroup Isomorphism belongs to*

$$\exists^{\log^2 n}\forall^{\log n}\exists^{\log n}\mathsf{DTISP}(\text{polylog}(n), \log(n)).$$

Note that we have $\exists^{\log^2 n}\forall^{\log n}\exists^{\log n}\mathsf{DTISP}(\text{polylog}(n), \log(n)) \subseteq \exists^{\log^2 n}\mathsf{AC}^0(\mathsf{DTISP}(\text{polylog}(n), \log(n))) \subseteq \mathsf{quasiAC}^0 \cap \exists^{\log^2 n}\mathsf{L} \cap \exists^{\log^2 n}\mathsf{FOLL}$.

Proof sketch. Careful analysis of the algorithm of [10, Thm. 3.4]. The key idea is to guess cube generating sequences for both input groups (in the $\exists^{\log^2 n}$ part) and then in the remaining computation verify whether these are in fact cube generating sequences that, indeed, induce an isomorphism. □

We obtain several corollaries, that all essentially follow by known $\mathsf{AC}^0$-reductions; [25] is a convenient reference for definitions of the problems below and the reductions.

**Corollary 6.2.** *The following isomorphism problems are in*

$$\exists^{\log^2 n}\mathsf{AC}^0(\mathsf{DTISP}(\text{polylog}(n), \log(n))):$$

*(1) Isomorphism of Steiner triple systems*
*(2) Isomorphism of pseudo-STS graphs*
*(3) Isomorphism of Steiner $(t, t+1)$-designs*
*(4) Isomorphism of Latin square graphs*
*(5) Latin Square Isotopy.*

Latin square graphs are one of the four families[11] of strongly regular graphs under Neumaier's classification [30]. Levet [25] established an upper bound of $\exists^{\log^2 n}\mathsf{AC}^0$ for isomorphism testing of conference graphs, which is a stronger upper bound than we obtain for Latin square graphs. In contrast, the best known algorithmic runtime for identifying conference graphs is $n^{2\log(n)+O(1)}$ [5], whereas Latin square graphs admit an $n^{\log(n)+O(1)}$-time solution [29].

[11] the other families being line graphs of Steiner 2-designs, conference graphs, and graphs whose eigenvalues satisfy the claw bound

## 7 MINIMUM GENERATING SET

In this section, we consider the Minimum Generating Set (MGS) problem for groups, quasigroups, and arbitrary magmas.

### 7.1 MGS for Groups in $\mathsf{AC}^1(\mathsf{L})$

**Theorem 7.1.** *MGS for groups belongs to* $\mathsf{AC}^1(\mathsf{L})$.

We begin with the following lemma from the full version [12].

**Lemma 7.2.** *Let $G$ be a group. We can compute a chief series for $G$ in* $\mathsf{AC}^1(\mathsf{L})$.

Proof of Thm. 7.1. By Lem. 7.2, we can compute a chief series $N_1 \triangleleft N_2 \triangleleft \cdots \triangleleft N_k = G$ in $\mathsf{AC}^1(\mathsf{L})$. We proceed inductively down the $N_i$. As $G/N_{k-1}$ is a finite simple group, and hence at most 2-generated, we can find a generating set in $\mathsf{L}$ [38].

Fix $i < k$. Given a minimum generating sequence $g_1, \ldots, g_d \in G$ for $G/N_i$ we construct a minimum generating sequence for $G/N_{i-1}$ as follows. If $N = N_i/N_{i-1}$ is Abelian (case 1):

- **Case 1a:** We have $G/N_{i-1} = \langle g_1, \cdots, g_i, g_j n, g_{j+1}, \cdots, g_d\rangle$ for some $j \in [d]$ and some $n \in N$ (possibly $n = 1$). There are at most $d \cdot |N|$ generating sets to consider in this case and we can test each of them in $\mathsf{L}$ with Membership [8; 33].
- **Case 1b:** If Case 1a does not hold, then by [27, Thm. 4] we have that $G/N_{i-1} = \langle g_1, \cdots, g_d, x\rangle$ for any non-identity element $x \in N$.

Otherwise, $N$ is non-Abelian (case 2), and we have by [28, Cor. 13] the following holds. Let $\eta_G(N)$ denote the the number of factors in a chief series with order $|N|$. Let $u = \max\{d, 2\}$ and $t = \min\{u, \lceil \frac{8}{5} + \log_{|N|}\eta_G(N)\rceil\}$. Then there exist $n_1, \ldots, n_t \in N_{i-1}$ (possibly $n_1 = \cdots = n_t = 1$) such that $G/N_{i-1} = \langle g_1 n_1, \cdots, g_t n_t, g_{t+1}, \cdots, g_d\rangle$.

By [28, Cor. 13], there are at most $|N|^{\lceil \frac{8}{5} + \log_{|N|}\eta_G(N)\rceil}$ generating sets of this form. As $\log_{|N|}\eta_G(N) \in O(1)$, we may write down these generating sets in parallel with a single $\mathsf{AC}^0$ circuit and test whether each generates $G/N_{i-1}$ in $\mathsf{L}$ using Membership.

Since a chief series has $O(\log n)$ terms, this algorithm requires $O(\log n)$ iterations and each iteration is computable in $\mathsf{L}$, resulting in an algorithm for MGS in $\mathsf{AC}^1(\mathsf{L})$. □

Improving upon the $\mathsf{AC}^1(\mathsf{L})$ bound on MGS for groups appears daunting. We thus inquire as to families of groups where MGS is solvable in complexity classes contained within $\mathsf{AC}^1(\mathsf{L})$. To this end, we examine the class of nilpotent groups. Arvind & Torán previously established a polynomial-time algorithm for nilpotent groups [1, Thm. 7]. We improve their bound as follows. Here, $d(G)$ denotes the minimum size of a generating set for $G$.

**Proposition 7.3.** *For a nilpotent group $G$, we can compute $d(G)$ in* $\mathsf{L} \cap \mathsf{AC}^0(\mathsf{NTISP}(\text{polylog}(n), \log(n)))$.

Proof sketch. Use order-finding (Cor. 3.2) to compute the Sylow $p$-subgroups in $\mathsf{AC}^0(\mathsf{DTISP}(\text{polylog}(n), \log(n)))$; we have $d(G) = \max_{\text{Sylow}P} d(P)$. For each Sylow subgroup $P$, compute the Frattini

subgroup $P^p[P,P]$ based on the Burnside Basis Theorem (see [34, Thm. 5.3.2]) and Cor. 3.2. Then $d(P) = |P/P^p[P,P]|$. A key fact we use along the way is that every element of $[P,P]$ is the product of at most $\log|P|$ commutators [36, I.§4 Ex. 5]. □

**Remark 7.4.** While Prop. 7.3 allows us to compute $d(G)$ for a nilpotent group $G$, the algorithm is non-constructive. It is not clear how to find such a generating set in L. We can, however, provide such a generating set in $\mathsf{AC}^1(\mathsf{NTISP}(\mathrm{polylog}(n), \log(n)))$. This bound is incomparable to $\mathsf{AC}^1(\mathsf{L})$. See the full version [12, Rmk. 7.4] for details.

## 7.2 MGS for Quasigroups

In this section, we consider the Minimum Generating Set problem for quasigroups. Our goal is to establish the following.

**Theorem 7.5.** *For MGS for quasigroups,*

*(a) The decision version belongs to* $\exists^{\log^2 n}\forall^{\log n}\mathsf{NTIME}(\mathrm{polylog}(n))$*;*
*(b) The search version belongs to* $\exists^{\log^2 n}\mathsf{SAC}^1 \subseteq \mathsf{DSPACE}(\log^2 n)$*;*[12] *and*
*(c) The search version belongs to* $\mathsf{quasiAC}^0$*.*

In the paper in which they introduced (polylog-)limited nondeterminism, Papadimitriou and Yannakakis conjectured that MGS for quasigroups was $\exists^{\log^2 n}\mathsf{P}$-complete [31, after Thm. 7]. While they did not specify the type of reductions used, it may be natural to consider polynomial-time many-one reductions. Thm. 7.5 refutes two versions of their conjecture under other kinds of reductions, that are incomparable to polynomial-time many-one reductions: $\mathsf{quasiAC}^0$ reductions unconditionally and polylog-space reductions conditionally. We note that their other $\exists^{\log^2 n}\mathsf{P}$-completeness result in the same section produces a reduction that in fact can be done in logspace and (with a suitable, but natural, encoding of the gates in a circuit) also in $\mathsf{AC}^0$, so our result rules out any such reduction for MGS.

**Corollary 7.6.** *MGS for quasigroups and Quasigroup Isomorphism are not* $\exists^{\log^2 n}\mathsf{P}$*-complete...*

*(a) ...under* $\mathsf{quasiAC}^0$ *Turing reductions;*
*(b) ...under polylog-space Turing reductions unless* $\mathsf{EXP} = \mathsf{PSPACE}$*.*

Proof. (a) As Parity is in P but not in $\mathsf{quasiAC}^0$, problems in $\mathsf{quasiAC}^0$ can't be $\exists^{\log^2 n}\mathsf{P}$-complete under $\mathsf{quasiAC}^0$ reductions.

(b) Both MGS for quasigroups and Quasigroup Isomorphism are in $\mathsf{DSPACE}(\log^2 n)$ by Thm. 7.5 (b), resp. [10]. The closure of $\mathsf{DSPACE}(\log^2 n)$ under poly-log space reductions is contained in $\mathsf{polyL} = \bigcup_{k\geq 0}\mathsf{DSPACE}(\log^k n)$. If either of these two quasigroup problems were complete for $\exists^{\log^2 n}\mathsf{P}$ under polylog-space Turing reductions, we would get $\exists^{\log^2 n}\mathsf{P} \subseteq \mathsf{polyL}$. Under the latter assumption, by a standard padding argument, we can show that $\mathsf{EXP} = \mathsf{PSPACE}$. See [12, Cor. 7.6] for full details. □

Now we return to establishing the main result of this section, Thm. 7.5. To establish Thm. 7.5 (a) and (c), we will crucially leverage the Membership problem for quasigroups. To this end, we will first establish the following.

**Theorem 7.7.** *Membership for quasigroups belongs to* $\mathsf{NTIME}(\mathrm{polylog}(n))$.

Thm. 7.7 immediately yields the following corollary.

**Corollary 7.8.** *For quasigroups, Membership and MGS are not hard under* $\mathsf{AC}^0$*-reductions for any complexity class containing Parity.*

The proofs of Thms. 7.7 and 7.5 rely crucially on the following adaption of the Babai–Szemerédi Reachability Lemma [4, Thm. 3.1] to quasigroups. We first generalize the notion of a straight-line program for groups [4] to SLPs for quasigroups. Let $X$ be a set of generators for a quasigroup $G$. We call a sequence of elements $g_1, \ldots, g_\ell \in G$ a *straight-line program* (SLP for short) if each $g_i$ ($i \in [\ell]$) either belongs to $X$, or is of the form or $g_j g_k$, $g_j \backslash g_k$, or $g_j / g_k$ for some $j, k < i$ (recall from §1 that $a\backslash b$ resp. $b/a$ denotes the unique $x$ such that $ax = b$ resp. $xa = b$). An SLP is said to *compute* or *generate* a set $S$ (or an element $g$) if $S \subseteq \{g_1, \ldots, g_\ell\}$ (resp. $g \in \{g_1, \ldots, g_\ell\}$).

**Lemma 7.9** (Reachability Lemma for quasigroups)**.** *Let $G$ be a finite quasigroup and let $X$ be a set of generators for $G$. For each $g \in G$, there exists a straight-line program over $X$ generating $g$ which has length $O(\log^2|G|)$.*

We follow the same strategy as in the proof of [4, Thm. 3.1], but there are some subtle, yet crucial, modifications due to the fact that quasigroups are non-associative and need not posses an identity element.

Proof. For any sequence of elements $z_1, \ldots, z_k$, let $P(z_1 z_2 \cdots z_k)$ denote the left-to-right parenthesization, e.g., $P(z_1 z_2 z_3) = (z_1 z_2) z_3$. For some initial segment $z_0, z_1, \ldots, z_i$ define the cube

$$K(i) = \{P(z_0 z_1^{e_1} \cdots z_i^{e_i}) : e_1, \ldots, e_i \in \{0,1\}\},$$

where $e_j = 0$ denotes omitting $z_j$ from the product (since there need not be an identity element).

Define $L(i) = K(i)\backslash K(i) = \{g\backslash h : g, h \in K(i)\}$. We will construct a sequence $z_0, z_1, \ldots, z_t$ such that $t \leq \lceil \log_2(n) \rceil$ and $L(t) = G$. Moreover, we derive a bound on the straight-line cost $c(i)$ for $\{z_0, z_1, \ldots, z_i\}$ ($1 \leq i \leq t$), which is defined as the length of the shortest SLP generating $\{z_0, z_1, \ldots, z_i\}$.

We take $z_0$ as an arbitrary element from $X$. Hence, $K(0) = \{z_0\}$, and so $c(0) = 1$. Next, let us construct $K(i+1)$ from $K(i)$. If $L(i) \neq G$, we set $z_{i+1}$ to be an element $z' \notin L(i)$ that minimizes $c(i+1) - c(i)$. We first claim that $|K(i+1)| = 2 \cdot |K(i)|$. Note that $K(i+1) = K(i) \cup K(i) z_{i+1}$ by definition. As right-multiplication by a fixed element is a bijection in a quasi-group, it suffices to show that $K(i) \cap K(i) z_{i+1} = \emptyset$. So, suppose that there exists some $a \in K(i) \cap K(i) z_{i+1}$. Then $a = g z_{i+1}$ for some $g \in K(i)$. Hence, $z_{i+1} = g\backslash a$, contradicting $z_{i+1} \notin L(i)$, since both $a$ and $g$ are in $K(i)$.

It now follows that $|K(i)| = 2^i$ and, hence, $t \leq \lceil \log_2(|G|) \rceil$ and $L(t) = G$. Moreover, for every $g \in G$ we obtain an SLP of length at most $c(t) + 2t + 1$: write $g = a\backslash b$ for $a, b \in K(t)$ and start with the SLP computing $\{z_0, z_1, \ldots, z_t\}$. We obtain an SLP for $a = z_0 z_1^{e_1} \cdots z_t^{e_t}$ by adding a new element for each $z_0 z_1^{e_1} \cdots z_i^{e_i}$ with $e_i = 1$ ($1 \leq i \leq t$,

[12] Wolf [43] showed the containment $\exists^{\log^2 n}\mathsf{SAC}^1 \subseteq \mathsf{DSPACE}(\log^2 n)$.

thus, at most $t$ new elements). Likewise, we get an SLP for $b$. Adding a last element $g = a\backslash b$, we obtain an SLP computing $g$.

It remains to bound the straight-line cost $c(i)$ of $\{z_0, z_1, \ldots, z_i\}$. Here, we claim that that $c(i+1) - c(i) \leq 4i + 3$. It follows from this claim that $c(i) \in O(i^2)$. We will now turn to proving our claim:

If $G \neq L(i)$, then either $X \not\subseteq L(i)$ or $L(i)$ is not a sub-quasigroup. Hence, we have one of the following cases:

**Case 1:** Suppose that there is some $g \in X - L(i)$. In this case, there is an SLP of length one for $g$ and we obtain $c(i+1) - c(i) \leq 1$.

**Case 2:** Suppose there exist $g, h \in L(i)$ with one of $gh$, $g/h$, or $g\backslash h \notin L(i)$. For simplicity, suppose $gh \notin L(i)$. The argument is identical for $g/h$ and $g\backslash h$. As above, given an SLP to compute $\{z_0, z_1, \ldots, z_i\}$, we may construct SLPs for $g$ and $h$ each of additional length $2i + 1$. This yields an SLP for $gh$ of total length at most $c(i) + 4i + 3$, and shows that $c(i+1) - c(i) \leq 4i + 3$. The result now follows. □

For proving Thm. 7.7 we follow essentially the ideas of [17] (though we avoid introducing the notion of Cayley circuits). Fleischer obtained a quasi$\mathsf{AC}^0$ bound for Membership for group by then showing that the Cayley circuits for this problem can be simulated by a quasi$\mathsf{AC}^0$ circuit. We will instead directly analyze the straight-line programs using an $\mathsf{NTIME}(\mathrm{polylog}(n))$ algorithm.

Proof of Thm. 7.7. To decide whether $g \in \langle X \rangle$, guess the sequence of operations for an SLP of length $\ell$, for $\ell \in O(\log^2 |G|)$, which exists iff $g \in \langle X \rangle$ by Lem. 7.9. Computing the values in the SLP and verifying that $g_\ell = g$ can be done in time $\mathrm{poly}(\ell, \log n) \leq \mathrm{polylog}(n)$. □

The proof of Thm. 7.5 (a) and (c) below is by describing a reduction from MGS to Membership. Thm. 7.5 (b) uses a result of Wagner.

Proof of Thm. 7.5. (a) Let $G$ denote the input quasigroup (of order $n$). First, observe that every quasigroup has a generating set of size $\leq \lceil \log n \rceil$ [29]. Therefore, we start by guessing a subset $X \subseteq G$ of size at most $\leq \lceil \log n \rceil$ (resp. the size bound given in the input). For the decision version, we use $O(\log^2 n)$ existentially quantified non-deterministic bits ($\exists^{\log^2 n}$) to guess a generating sequence. To find a minimum-sized generating sequence, we enumerate all possible generating sequences in quasi$\mathsf{AC}^0$. In the next step, we verify whether $X$ actually generates $G$. This is done by checking for all $g \in G$ (universally verifying $O(\log n)$ bits, $\forall^{\log n}$) whether $g \in \langle X \rangle$, which can be done in $\mathsf{NTIME}(\mathrm{polylog}(n)) \subseteq$ quasi$\mathsf{AC}^0$ by Thm. 7.7.

(c) Proceed as in (a), using the same technique to check that all $Y \subseteq G$ with $|Y| < |X|$ do not generate $G$.

(b) Existentially guess a generating set using $O(\log^2 n)$ bits, then use [42, Thm. 10.2.1] to compute $\langle X \rangle$ in $\mathsf{SAC}^1$. Then we check, in $\mathsf{AC}^0$, whether each element of the quasigroup belongs to $\langle X \rangle$. □

We may similarly reduce Quasigroup Isomorphism to Membership for quasigroups. This formalizes the intuition that Membership is an essential subroutine for isomorphism testing and MGS. In particular, in the setting of quasi$\mathsf{AC}^0$, we have that isomorphism testing and MGS reduce to Membership. This latter consequence might seem surprising, as in the setting of groups, Membership belongs to $\mathsf{L}$, while MGS belongs to $\mathsf{AC}^1(\mathsf{L})$ (Thm. 7.1), yet it is a longstanding open problem whether Group Isomorphism is even in $\mathsf{P}$.

### 7.3 MGS for Magmas

**Theorem 7.10.** *The decision variant of Minimum Generating Set for magmas is* NP*-complete.*

This NP-completeness result helps explain the use of Integer Linear Programming in practical heuristic algorithms for the search version of this problem, e.g., [22].

For the closely related problem Log Generators—given the multiplication table of a magma of order $n$, decide whether it has a generating set of size $\leq \lceil \log_2 n \rceil$—Log Generators of Magmas is $\exists^{\log^2 n}\mathsf{P}$-complete under polynomial-time reductions [31, Thm. 7].

Proof sketch. The problem is in NP by guessing a suitable generating set. To show NP-hardness we reduce from 3SAT. Let $F = \bigwedge_{j=1}^{m} C_j$ with variables $X_1, \ldots, X_n$ be an instance of 3SAT. Our magma $M$ consists of the following elements: an element $C_j$ for $j \in [m]$, elements $X_i, \overline{X}_i$ for $i \in [n]$, elements $T_{j,k}$ for $1 \leq j \leq k \leq m$, and a zero element 0. We define the multiplication as follows:

$$C_j X = T_{j,j} \text{ if the literal } X \text{ appears in } C_j$$
$$T_{1,m} X_i = \overline{X}_i, \qquad T_{1,m} \overline{X}_i = X_i,$$
$$T_{j,k} T_{k+1,\ell} = T_{j,\ell}.$$

all other products are defined as 0.

Any generating set for $M$ must include all the $C_j$, since without them, there is no way to generate them from other elements. Similarly, any generating set must include, for each $i \in [n]$, at least one of $X_i$ or $\overline{X}_i$, since they also can't be generated from other elements.

We show that $F$ is satisfiable if and only if $M$ can be generated by $n+m$ elements. When $F$ is satisfiable, let $\varphi$ be a satisfying assignment, then $M$ is generated by $\{C_j : j \in [m]\} \cup \{X_i : \varphi(x_i) = 1\} \cup \{\overline{X}_i : \varphi(x_i) = 0\}$. Conversely, any generating set of size $n + m$ must consist of all $C_j$ and exactly one of each $\{X_i, \overline{X}_i\}$. As the only way to generate $T_{j,j}$ is for one of the literals to satisfy $C_j$, the choice of $X_i, \overline{X}_i$ in the generating set must be a satisfying assignment. □

## 8 OPEN QUESTIONS

The biggest open question about constant-depth complexity on algebras given by multiplication tables is, in our opinion, still whether or not Group Isomorphism is in $\mathsf{AC}^0$ in the Cayley table model. Our results make salient some more specific, and perhaps more approachable, open questions that we now highlight.

**Question 8.1.** Does MGS for groups belong to $\mathsf{L}$?

**Question 8.2.** Does Membership for quasigroups belong to $\mathsf{L}$?

The analogous result is known for groups, by reducing to the connectivity problem on Cayley graphs. The best known bound for quasigroups is $\mathsf{SAC}^1$ [42]. Improvements in this direction would yield improvements in MGS for quasigroups. A *constructive* membership test would also yield improvements for isomorphism testing of $O(1)$-generated quasigroups. Note that isomorphism testing of $O(1)$-generated groups is known to belong to $\mathsf{L}$ [38].

## REFERENCES

[1] Vikraman Arvind and Jacobo Torán. The complexity of quasigroup isomorphism and the minimum generating set problem. In Tetsuo Asano, editor, *Algorithms and Computation, 17th International Symposium, ISAAC 2006, Kolkata, India, December*

*18-20, 2006, Proceedings*, volume 4288 of *Lecture Notes in Computer Science*, pages 233–242. Springer, 2006. doi:10.1007/11940128_25.

[2] James Aspnes, Richard Beigel, Merrick Furst, and Steven Rudich. The expressive power of voting polynomials. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, page 402–409, New York, NY, USA, 1991. Association for Computing Machinery. doi:10.1145/103418.103461.

[3] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 684–697. ACM, New York, 2016. Preprint of full version at arXiv:1512.03547v2 [cs.DS]. doi:10.1145/2897518.2897542.

[4] László Babai and Endre Szemerédi. On the complexity of matrix group problems I. In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 229–240. IEEE Computer Society, 1984. doi:10.1109/SFCS.1984.715919.

[5] László Babai. On the complexity of canonical labeling of strongly regular graphs. *SIAM Journal on Computing*, 9(1):212–216, 1980. doi:10.1137/0209018.

[6] D.A.M. Barrington. Quasipolynomial size circuit classes. In *[1992] Proceedings of the Seventh Annual Structure in Complexity Theory Conference*, pages 86–93, 1992. doi:10.1109/SCT.1992.215383.

[7] David A. Mix Barrington, Peter Kadau, Klaus-Jörn Lange, and Pierre McKenzie. On the complexity of some problems on groups input as multiplication tables. *J. Comput. Syst. Sci.*, 63(2):186–200, 2001. doi:10.1006/jcss.2001.1764.

[8] David A. Mix Barrington and Pierre McKenzie. Oracle branching programs and Logspace versus P. *Inf. Comput.*, 95(1):96–115, 1991. doi:10.1016/0890-5401(91)90017-V.

[9] David A. Mix Barrington and Howard Straubing. Complex polynomials and circuit lower bounds for modular counting. *Comput. Complex.*, 4:325–338, 1994. doi:10.1007/BF01263421.

[10] Arkadev Chattopadhyay, Jacobo Torán, and Fabian Wagner. Graph isomorphism is not $AC^0$-reducible to group isomorphism. *ACM Trans. Comput. Theory*, 5(4):Art. 13, 13, 2013. Preliminary version appeared in FSTTCS '10; ECCC Tech. Report TR10-117. doi:10.1145/2540088.

[11] Anthony E. Clement, Stephen Majewicz, and Marcos Zyman. *The theory of nilpotent groups*. Birkhäuser/Springer, Cham, 2017.

[12] Nathaniel A. Collins, Joshua A. Grochow, Michael Levet, and Armin Weiß. On the constant-depth circuit complexity of generating quasigroups. arXiv:2402.00133 [cs.CC], 2024.

[13] Bireswar Das and Dhara Thakkar. Algorithms for the minimum generating set problem, 2023. arXiv:2305.08405.

[14] Heiko Dietrich, Murray Elder, Adam Piggott, Youming Qiao, and Armin Weiß. The isomorphism problem for plain groups is in $\Sigma_3^p$. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 26:1–26:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.26.

[15] Heiko Dietrich and James B. Wilson. Group isomorphism is nearly-linear time for most orders. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 457–467, 2022. doi:10.1109/FOCS52979.2021.00053.

[16] Flavio Ferrarotti, Senén González, Klaus-Dieter Schewe, and José María Turull-Torres. Proper hierarchies in polylogarithmic time and absence of complete problems. In Andreas Herzig and Juha Kontinen, editors, *Foundations of Information and Knowledge Systems*, pages 90–105, Cham, 2020. Springer International Publishing.

[17] Lukas Fleischer. The Cayley semigroup membership problem. *Theory of Computing*, 18(8):1–18, 2022. doi:10.4086/toc.2022.v018a008.

[18] Joshua A. Grochow and Michael Levet. On the Parallel Complexity of Group Isomorphism via Weisfeiler-Leman. In Henning Fernau and Klaus Jansen, editors, *Fundamentals of Computation Theory - 24th International Symposium, FCT 2023, Trier, Germany, September 18-21, 2023, Proceedings*, volume 14292 of *Lecture Notes in Computer Science*, pages 234–247. Springer, 2023. Preprint of full version at arXiv:2112.11487 [cs.DS]. doi:10.1007/978-3-031-43587-4_17.

[19] Joshua A. Grochow and Youming Qiao. Algorithms for group isomorphism via group extensions and cohomology. *SIAM J. Comput.*, 46(4):1153–1216, 2017. Preliminary version in IEEE Conference on Computational Complexity (CCC) 2014 (DOI:10.1109/CCC.2014.19). Also available as arXiv:1309.1776 [cs.DS] and ECCC Technical Report TR13-123. doi:10.1137/15M1009767.

[20] Harald Andrés Helfgott, Jitendra Bajpai, and Daniele Dona. Graph isomorphisms in quasi-polynomial time, 2017. doi:10.48550/ARXIV.1710.04574.

[21] Jeffrey C. Jackson, Adam R. Klivans, and Rocco A. Servedio. Learnability beyond $AC^0$. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 776–784, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/509907.510018.

[22] Mikolás Janota, António Morgado, and Petr Vojtechovský. Computing generating sets of minimal size in finite algebras. *J. Symb. Comput.*, 119:50–63, 2023. doi:10.1016/J.JSC.2023.02.002.

[23] Emil Jerábek. Answer to "is abelian group isomorphism in $AC^0$?". Theoretical Computer Science Stack Exchange, 06 2020. https://cstheory.stackexchange.com/q/40118 (version: 2020-06-17).

[24] François Le Gall and David J. Rosenbaum. On the group and color isomorphism problems. arXiv:1609.08253 [cs.CC], 2016.

[25] Michael Levet. On the complexity of identifying strongly regular graphs. *Australasian Journal of Combinatorics*, 87:41–67, 2023. URL: https://ajc.maths.uq.edu.au/pdf/87/ajc_v87_p041.pdf.

[26] R. J. Lipton, L. Snyder, and Y. Zalcstein. The complexity of word and isomorphism problems for finite groups. Yale University Dept. of Computer Science Research Report # 91, 1977. URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/a053246.pdf.

[27] Andrea Lucchini and Federico Menegazzo. Computing a set of generators of minimal cardinality in a solvable group. *Journal of Symbolic Computation*, 17(5):409–420, 1994. doi:10.1006/jsco.1994.1027.

[28] Andrea Lucchini and Dhara Thakkar. The minimum generating set problem. *Journal of Algebra*, 640:117–128, 2024. doi:10.1016/j.jalgebra.2023.11.012.

[29] Gary L. Miller. On the $n^{\log n}$ isomorphism technique (a preliminary report). In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 51–58, New York, NY, USA, 1978. Association for Computing Machinery. doi:10.1145/800133.804331.

[30] A. Neumaier. Strongly regular graphs with smallest eigenvalue $-m$. *Archiv der Mathematik*, 33:392–400, 1979. doi:10.1007/BF01222774.

[31] Christos H. Papadimitriou and Mihalis Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *J. Comput. Syst. Sci.*, 53(2):161–170, 1996. doi:10.1006/JCSS.1996.0058.

[32] Alexander A Razborov. An Equivalence between Second Order Bounded Domain Bounded Arithmetic and First Order Bounded Arithmetic. In *Arithmetic, proof theory, and computational complexity*. Oxford University Press, 05 1993. doi:10.1093/oso/9780198536901.003.0012.

[33] Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4), sep 2008. doi:10.1145/1391289.1391291.

[34] D. Robinson. *A Course in the Theory of Groups*. Springer, 1982.

[35] David J. Rosenbaum. Bidirectional collision detection and faster deterministic isomorphism testing. arXiv:1304.3935 [cs.DS], 2013.

[36] Jean-Pierre Serre. *Galois Cohomology*. Springer Berlin, Heidelberg, 1 edition, 1997. doi:10.1007/978-3-642-59141-9.

[37] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82. ACM, 1987. doi:10.1145/28395.28404.

[38] Bangsheng Tang. *Towards Understanding Satisfiability, Group Isomorphism and Their Connections*. PhD thesis, Tsinghua University, 2013. URL: http://papakonstantinou.org/periklis/pdfs/bangsheng_thesis.pdf.

[39] D. R. Taunt. Remarks on the isomorphism problem in theories of construction of finite groups. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(1):16–24, 1955. doi:10.1017/S030500410002987X.

[40] Jacobo Torán. On the hardness of graph isomorphism. *SIAM J. Comput.*, 33(5):1093–1108, 2004. doi:10.1137/S009753970241096X.

[41] Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999. doi:10.1007/978-3-662-03927-4.

[42] F. Wagner. *On the complexity of isomorphism testing for restricted classes of graphs*. PhD thesis, Universität Ulm, 2010. URL: https://oparu.uni-ulm.de/xmlui/bitstream/handle/123456789/3923/vts_7264_10267.pdf.

[43] Marty J. Wolf. Nondeterministic circuits, space complexity and quasigroups. *Theoretical Computer Science*, 125(2):295–313, 1994. doi:10.1016/0304-3975(92)00014-I.

Received 6 February 2024; revised 12 March 2009; accepted 5 June 2009