On the Weisfeiler-Leman Dimension of Finite Groups

Jendrik Brachter
TU Kaiserslautern

Pascal Schweitzer
TU Kaiserslautern

Abstract

In comparison to graphs, combinatorial methods for the isomorphism problem of finite groups are less developed than algebraic ones. To be able to investigate the descriptive complexity of finite groups and the group isomorphism problem, we define the Weisfeiler-Leman algorithm for groups. In fact we define three versions of the algorithm. In contrast to graphs, where the three analogous versions readily agree, for groups the situation is more intricate. For groups, we show that their expressive power is linearly related. We also give descriptions in terms of counting logics and bijective pebble games for each of the versions.

In order to construct examples of groups, we devise an isomorphism and non-isomorphism preserving transformation from graphs to groups. Using graphs of high Weisfeiler-Leman dimension, we construct similar but non-isomorphic groups with equal $\Theta(\log n)$-subgroup-profiles, which nevertheless have Weisfeiler-Leman dimension 3. These groups are nilpotent groups of class 2 and exponent p, they agree in many combinatorial properties such as the combinatorics of their conjugacy classes and have highly similar commuting graphs.

The results indicate that the Weisfeiler-Leman algorithm can be more effective in distinguishing groups than in distinguishing graphs based on similar combinatorial constructions.

CCS Concepts: **• Theory of computation → Graph algorithms analysis**; **Finite Model Theory**; **• Mathematics of computing → Graph algorithms**.

Keywords: group isomorphism, Weisfeiler-Leman, logics with counting, Cai-Fürer-Immermann graphs

ACM Reference Format:
Jendrik Brachter and Pascal Schweitzer. 2020. On the Weisfeiler-Leman Dimension of Finite Groups. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20), July 8–11, 2020, Saarbrücken, Germany.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3373718.3394786

Acknowledgments

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

1 Introduction

The notion of isomorphisms between finite groups remains one of the most basic concepts of group theory for which we do not have efficient algorithmic tools. The algorithmic Group Isomorphism Problem formalizes the task of deciding whether two given (finite) groups are isomorphic, but in fact, we do not understand its complexity. We have neither a polynomial time algoritheorem for testing isomorphism, nor complexity theoretic evidence indicating to us that the problem is not polynomial time solvable. Considering groups of order n, a simple approach, attributed to Tarjan in [29], is to pick a small generating set in one of the groups and to check for all possible images of the generators in the other group, whether the partial map extends to an isomorphism. This approach gives us a worst-case runtime of $n^{\log t+O(1)}$ where t is the size of the generating set. Since every group of order n has a generating set of size at most $\log n$, this yields $n^{\log n+O(1)}$ in the worst case. Despite decades of active research this bound has seen only slight improvements for the general case. In fact, Rosenbaum [34] was able to improve it to $n^{1/2\log n+O(1)}$. (See [27] and [35] for related discussions on isomorphism of p-groups and solvable groups). For various classes of groups, better bounds are known (see further related work). However, even very limited classes of groups provide hard cases for isomorphism testing. One of the most prominent classes in this context is formed by the groups of prime exponent p and nilpotency class 2. Such groups possess a lot of extra structure, but despite this and despite a large body of research into this structure, even for this limited class, no better general bound has been proven. In fact, this class seems to be at the core of the problem. However, a formal reduction to this or a similar class is not known.

While there exists a vast collection of algebraic methods and heuristics for tackling the group isomorphism problem (see further related work), complexity theoretic and combinatorial aspects seem to be less developed. For example, in 2011, Timothy Gowers asked on Lipton's blog [17] whether there is an integer m such that the isomorphism

LICS '20, July 8–11, 2020, Saarbrücken, Germany

ACM ISBN 978-1-4503-7104-9/20/07...$15.00
https://doi.org/10.1145/3373718.3394786

class of each finite group is determined by their m-subgroup-profile. Here the m-subgroup-profile (or m-profile) is the multiset of (isomorphism types of) m-generated subgroups. Glauberman and Grabowski gave a negative answer by constructing pairs of non-isomorphic groups with the same $\Theta(\sqrt{\log n})$-profiles [15]. Subsequently, Wilson constructed many examples of exponent p and nilpotency class 2 groups which agree in various invariants. In particular they have the same $\Theta(\log n)$-profiles [40], which is best possible.

The observation that combinatorial aspects of the group isomorphism problem are less developed is surprising since, for the related graph isomorphism problem, historically, it has been the other way around. Indeed, for graph isomorphism testing, combinatorial approaches are well-developed and often successful, yet their limits have been firmly established. One of the most important tools in this scope is the Weisfeiler-Leman algorithm. The k-dimensional Weisfeiler-Leman algorithm (k-WL) iteratively classifies k-tuples of vertices of a graph in terms of how they are related to other vertices in the graph. It provides an effective invariant for graph-non-isomorphism (see e.g. [8, 37]). Moreover, k-WL can be implemented to run in time $O(n^{k+1} \log n)$ where n is the number of vertices (see [22, 23]). For fixed k, the algorithm is only a partial isomorphism test, in that it can distinguish certain pairs of non-isomorphic graphs, but not all of them. A graph is said to have WL-dimension at most k, if k-WL distinguishes the graph from every non-isomorphic graph. For many important classes of graphs the WL-dimension has been shown to be bounded; examples include planar graphs [19, 25] for which even 3 suffices and more generally classes defined by forbidden minors [20]. On the other hand, Cai, Fürer and Immerman constructed an infinite family of graphs, for which the WL-dimension is linear in the number of vertices, and thus unbounded [8]. Higher dimensional versions of k-WL also appear in Babai's breakthrough result putting graph-isomorphism in quasi-polynomial time [1].

There is a deep and well-understood connection between k-WL and the expressiveness in the logic $\mathbf{C}^{k+1}$, the extension of the $(k+1)$-variable fragment of first order logic on graphs with counting quantifiers [8]. For example, two graphs can be distinguished by k-WL exactly if there is a formula in $\mathbf{C}^{k+1}$ that distinguishes the graphs. Therefore, in some well-defined sense, the k-WL algorithm is universal in that it simultaneously checks all combinatorial properties in an input graph expressible in the aforementioned logic.

Contribution. The first aim of this paper is to introduce Weisfeiler-Leman-type algorithms and the notion of a WL-dimension for groups analogous to the graph case. While at first sight it seems straightforward to do so, it turns out that various concepts that coincide when applied to graphs (potentially) disagree when applied to groups. Specifically, we define three natural but different versions of a Weisfeiler-Leman dimension. One of them is based on a natural logic for groups while another is natural when taking an algorithmic viewpoint. The third version comes from natural translation of groups into graphs in an isomorphism and non-isomorphism preserving manner. Still, we give descriptions in terms of counting logics and bijective pebble games for each of the versions. A core reason why the different versions arise is that the correspondence between the various concepts arising in this context (specifically logics, algorithms and pebble games) is not as clean as for graphs.

However, we argue that the definition is robust after all: we prove that the Weisfeiler-Leman dimensions of the different versions are linearly related. Overall, we obtain a family of algorithms that is similarly universal in checking combinatorial properties as in the graph case. For example, it is easy to see that the k-WL algorithm implicitly computes the k-profile of groups. In particular, abelian groups are completely identified already by the least powerful of the algorithms.

The second aim of this paper is to understand when and how groups are characterized by their combinatorial properties. On the one hand this addresses the question whether combinatorial methods can solve the Group Isomorphism Problem. On the other hand it provides a way of quantifying similarity of non-isomorphic groups. Specifically, we construct pairs of arbitrarily large non-isomorphic groups that agree with respect to many isomorphism invariants but can still be distinguished with 3-dimensional WL-refinement. More precisely these groups are of nilpotency class 2 and prime exponent p. They are non-isomorphic but have the same $\Theta(\log n)$-profile. They also have highly similar commuting graphs.

Theorem 1.1. *For infinitely many n there exist pairs of non-isomorphic groups order n with bounded Weisfeiler-Leman dimension which*

- *have equal $\Theta(\log n)$-profiles,*
- *have commuting graphs that are indistinguishable for the $O(\log(n))$-dimensional Weisfeiler-Leman algorithm (for graphs),*
- *are of exponent p and nilpotency class 2, and*
- *have equal sizes of conjugacy classes.*

The theorem shows that the Weisfeiler-Leman algorithm can be more effective in distinguishing groups than in distinguishing graphs even when they are based on similar combinatorial constructions. The proof that the WL-dimension is low intuitively indicates that the ability to fix products of elements appears to be related to the ability to fix sets of elements and how to exploit this.

In comparison to the previous constructions mentioned above, our construction has the advantage that it is of a purely combinatorial nature. It is therefore easy to analyze the groups, and many combinatorial properties of the resulting groups can be tuned. In fact, we can start with an arbitrary graph and encode it into a group while preserving isomorphisms. We should stress that even though we start with graphs of unbounded Weisfeiler-Leman dimension, the resulting groups have only dimension 3. This highlights the power of Weisfeiler-Leman-type algorithms to distinguish groups beyond the scope of traditional invariants.

1.1 Further related work

Our work can be understood as studying the descriptive complexity of finite groups. We refer to Grohe's monograph [20] for extensive information on the descriptive complexity of graphs (rather than groups). A central result in [20] shows graph classes with a forbidden minor have bounded WL-dimension. A recent paper relating first order logics and groups is [30]. The descriptive complexity of finite abelian groups has been studied in [16]. However, descriptive complexity of groups has been investigated considerably less than that of graphs. In contrast to this, the research body on the algorithmic Group Isomorphism Problem is extensive. The results can generally be divided into research with a more practical and research with a more theoretical focus.

On the practical side the best algorithms for isomorphism testing are typically implemented in computer algebra systems such as SAGE, MAGMA, GAP. Classical algorithms include the one by Smith [36] (for solvable groups), the one by Eick, Leedham-Green and O'Brien (for p-groups) [13, 31], as well as a general algorithm by Cannon and Holt [10]. Newer algorithms have been developed by Wilson [39] with numerous improvements over time together with Brooksbank and Maglione [7]. More recent work introduces ever stronger invariants to distinguish groups quickly. We refer to [6] for an overview and the most recent techniques and an algorithm incorporating many of them. Dietrich and Wilson report that current isomorphism tests are already infeasible in practice on some groups with orders in the thousands [12].

In any case, in our work we focus on the *theoretical side*. As mentioned before, the best bound for the general problem is by Rosenbaum [34]. Polynomial time algorithms have been developed for various classes of groups [2–4, 11, 14, 18, 24, 33]. There is an algorithm running in polynomial time for most orders [12]. For the currently fastest isomorphism algorithm for permutation groups see [38].

Recent efforts incorporate the Weisfeiler-Leman algorithm into the group isomorphism context [6, 26]. However, there is a crucial difference to our work. Indeed, in these papers the authors use a combinatorial construction within the groups acting on vector spaces on which the (graph) WL-algorithm is executed. This is different to the general algorithm for all groups defined here. Thus, a priori the two algorithmic approaches are unrelated, warranting further study.

2 Preliminaries and notation

Groups. Groups will be denoted by capital Latin characters. For a group G and elements $g, h \in G$ we write their *commutator* as $[g, h] := ghg^{-1}h^{-1}$ and we use G' to refer to the subgroup of G generated by all commutators. Then G' is the unique minimal normal subgroup of G with abelian quotient. The *centralizer* of $x \in G$ is $C_G(x) := \{g \in G \mid [x, g] = 1\}$ and then $Z(G) := \{g \in G \mid C_G(g) = G\}$ is the *center* of G. For a prime p, a group is called a p-group if $|G| = p^n$ is a power of p (in particular, we assume G to be finite here). The *exponent* of a group is the least common multiple of the orders of its elements. A p-group G is *elementary abelian* if it is abelian and of prime exponent (i.e., $G \cong \mathbb{F}_p^n$ for some n). The *Frattini-subgroup* $\Phi(G)$ of a group G is the intersection of all maximal subgroups. If G is a p-group then $\Phi(G)$ is the unique minimal normal subgroup of G with elementary abelian quotient. The elements of $\Phi(G)$ are *non-generators* in G, that is, if $\{g_1, \dots, g_m\}$ generates G then so does $\{g_1, \dots, g_m\} \setminus \Phi(G)$.

We define a 1-fold commutator to be just a regular commutator and then a c-fold commutator is an element of the form $[x, z]$ with $x \in G$ and z a $(c-1)$-fold commutator in G. A group G is said to be *nilpotent* if there is some integer c such that c-fold commutators are always trivial in G and if this is the case then the *nilpotency class* of G is the smallest such c. For example abelian groups are exactly the groups of nilpotency class 1 and a group has nilpotency class 2 if and only if it is non-abelian and every commutator is central.

A *group isomorphism* is a bijective map $\varphi\colon G \to H$ that preserves group multiplication. We collect all isomorphisms between G and H in a (possibly empty) set $\mathrm{Iso}(G, H)$ and set $\mathrm{Aut}(G) := \mathrm{Iso}(G, G)$. We write $\mathrm{Sub}(G)$ for the set of all subgroups of G.

We assume the term 'group' to mean 'finite group' and whenever we include infinite groups we do so explicitly.

Graphs. All graphs will be finite simple undirected graphs and referred to with greek characters, primarily Γ, subject to suitable subscripts. That is, a graph Γ consists of a finite set of vertices $V(\Gamma)$ and a set of edges $E(\Gamma) \subseteq \binom{V(\Gamma)}{2} := \{M \subseteq V(\Gamma) \mid |M| = 2\}$. The *complement* of Γ will always be the *simple* complement graph, namely $\mathrm{co}(\Gamma) := \left(V(\Gamma), \binom{V}{2} - E(\Gamma)\right)$.

The set of *neighbors* of $v \in V(\Gamma)$ is $N(v) := \{w \in V(\Gamma) \mid \{v, w\} \in E\}$ and $N[v] := N(v) \cup \{v\}$ is the *closed neighborhood* of v. The *degree* of v is $d(v) := |N(v)|$. A graph is *d-regular* if every vertex has degree d. For a set of vertices $M \subseteq V(\Gamma)$, the *induced subgraph* is $\Gamma[M] := \left(M, E(\Gamma) \cap \binom{M}{2}\right)$.

An *isomorphism* of graphs is a bijective map $\varphi\colon V(\Gamma_1) \to V(\Gamma_2)$ that simultaneously preserves edges and non-edges. The set of isomorphisms between Γ_1 and Γ_2 is $\mathrm{Iso}(\Gamma_1, \Gamma_2)$ and define $\mathrm{Aut}(\Gamma_1) := \mathrm{Iso}(\Gamma_1, \Gamma_1)$.

The *commuting graph* of a group is the graph whose vertices are the group elements and two distinct elements g, g' are adjacent if $[g, g'] = 1$.

2.1 The WL-algorithm for graphs

Before we explore how the WL-algorithm can be applied to groups, we briefly recapitulate its classic definition for graphs. Given a graph Γ, the k-dimensional version of the algorithm for positive $k \in \mathbb{N}$ repeatedly colors the k-tuples of vertices with abstract colors that encode how each tuple is situated within the graph. The initial coloring of each tuple $(g_1, \dots, g_k)$ encodes the isomorphism type of the graph induced by $\{g_1, \dots, g_k\}$, taking into account where the vertices occur in the tuple. Specifically, a coloring $\chi_0\colon V(\Gamma)^k \to S$ into some set of colors is defined so that $\chi_0(g_1, \dots, g_k) = \chi_0(g'_1, \dots, g'_k)$ holds exactly if there is an isomorphism from $\Gamma[\{g_1, \dots, g_k\}]$ to $\Gamma[\{g'_1, \dots, g'_k\}]$ which sends g_i to g'_i for all $i \in \{1, \dots, k\}$.

The coloring is now iteratively refined as follows. For a tuple $\bar{g} = (g_1, \dots, g_k) \in V(\Gamma)^k$ and $x \in V(\Gamma)$, define $\bar{g}_{|i \leftarrow x}$ to be the tuple $(g_1, \dots, g_{i-1}, x, g_{i+1}, \dots, g_k)$ obtained by replacing the i-th entry with x. Then we define for $k > 1$ the coloring $\chi_i(\bar{g}) :=$

$$\left(\chi_{i-1}(\bar{g}), \{\!\{(\chi_{i-1}(\bar{g}_{|1\leftarrow x}), \dots, \chi_{i-1}(\bar{g}_{|k\leftarrow x})) \mid x \in V(\Gamma)\}\!\}\right).$$

Here $\{\!\{\cdots\}\!\}$ denotes multisets. Thus the color of the next iteration consists of the color of the previous iteration and the multiset of colors obtained by replacing each entry in the tuple with another vertex from the graph. For $k = 1$ the definition is slightly different, namely that for $\chi_i(\bar{v})$ the multiset is only taken over vertices x in the neighborhood $N(v)$.

Adding the color of the previous iteration as first entry ensures that the partition induced on $V(\Gamma)$ by χ_i is finer than (or as fine as) the partition induced by χ_{i-1}. Let j be the least positive integer for which the partition induced by χ_{j-1} agrees with the partition induced by χ_j, then we define the final coloring χ_∞ to be χ_{j-1}. Since the domain of the χ_i has size $|V(\Gamma)|^k$, we know that $j \leq |V(\Gamma)|^k$. For fixed $k \in \mathbb{N}$, it is possible to compute the partition of χ_∞ in time $O(n^{k+1}\log(n))$ [22].

To distinguish two non-isomorphic graphs the algorithm is applied on the disjoint union. If in the final coloring the multiset of colors appearing in one graph is different than those appearing in the other graph, then the graphs are not isomorphic. The converse does not necessarily hold, as we explain next.

2.2 The CFI-graphs

As mentioned previously, for each k there is a pair of non-isomorphic graphs not distinguished by k-WL.

Theorem 2.1 (Cai, Fürer, Immerman [8])**.** *There is an infinite family of pairs of non-isomorphic 3-regular graphs on $O(k)$ vertices not distinguished by the k-dimensional Weisfeiler-Leman algorithm.*

Since we intend to exploit the construction by transferring it to groups, we describe it next. We start with a connected base graph Γ. In this graph every vertex is replaced by a particular gadget and the gadgets are interconnected according to the edges of Γ as follows. For a vertex v of degree d we use the gadget F_d, which is a graph whose vertex set consists of external vertices $O_d = \{a_1^v, b_1^v, a_2^v, b_2^v, \dots, a_d^v, b_d^v\}$ and internal vertices M_d. The internal vertices form a copy of the set of those 0-1-strings of length d that have an even number of entries equal to 1. For each i, each internal vertex m is adjacent to exactly one vertex of $\{a_i^v, b_i^v\}$, namely it is adjacent to a_i if the i-th bit of the string m is 0 and to b_i otherwise. An example of F_3 is depicted in Figure 1. It remains to explain how the different gadgets are interconnected. For this, for a vertex $v \in \Gamma$ of degree d each edge is associated with one of the pairs a_i^v, b_i^v. For an edge $(u, v) \in E(\Gamma)$ assume u is associated with the pair (a_i^u, b_i^u) in the gadget corresponding to u and v is associated with the pair (a_j^v, b_j^v) in the gadget corresponding to v. Then we insert (parallel) edges $\{a_i^u, a_j^v\}$ and $\{b_i^u, b_j^v\}$. Adding such parallel edges for each edge of the base graph we obtain the graph $\mathrm{CFI}(\Gamma)$. The *twisted CFI-graph* $\widetilde{\mathrm{CFI}(\Gamma)}$ is obtained by replacing one pair of (parallel) edges $\{a_i^u, a_j^v\}$ and $\{b_i^u, b_j^v\}$ with the (twisted) edges $\{a_i^u, b_j^v\}$ and $\{b_i^u, a_j^v\}$. It can be shown that for connected base graphs (up to isomorphism) it is irrelevant which edge is twisted [8]. For a subset of the edges of the base graph $E' \subseteq E(\Gamma)$, we can define the graph obtained by twisting exactly the edges in E'. The resulting graph is isomorphic to $\mathrm{CFI}(\Gamma)$ if $|E'|$ is even and isomorphic to $\widetilde{\mathrm{CFI}(\Gamma)}$ otherwise.

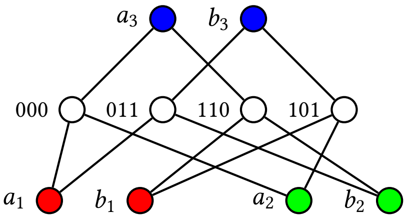

Figure 1. A depiction of the CFI-gadget F_3.

In the original construction the base graph is usually thought of as vertex colored with all vertices obtaining a different color. This makes all gadgets distinguishable. The colors can be removed by attaching gadgets retaining the property that the base graph is identified by 2-dimensional Weisfeiler-Leman. We want to record here the observation that it is possible to choose the base graph of WL-dimension 2 while maintaining the property that it is 3-regular.

Observation 2.1. *The 3-regular base graph Γ can be chosen to have Weisfeiler-Leman dimension at most 2.*

This can be seen in two ways, by adding gadgets on edges or by observing that random expanders, usually used in the construction, have this property.

2.3 First order logic with counting

There is a close connection between the Weisfeiler-Leman algorithm of dimension k and the $(k + 1)$-variable fragment of first order logic on graphs with counting quantifiers [8]. To obtain this logic we endow first order logic with counting quantifiers. The formula $\exists^{\geq i} x\varphi(x)$ expresses then the fact that there are at least i distinct elements that satisfy the formula φ. For example the formula $\exists^{\geq 3} x\exists^{\geq 4} yE(x,y)$ would express that the graph contains at least 3 vertices of degree at least 4. The logic $\mathbf{C}^k$ is the fragment of said logic which allows formulas to only use k distinct variables (that can however be reused an arbitrary number of times). We refer to [22] for a more thorough introduction to these logics and a proof that two graphs can be distinguished by k-dimensional WL exactly if there is a formula in $\mathbf{C}^{k+1}$ that holds on the one graph but not on the other. Often such logics are endowed with a fixed-point operator, but since we will only apply the formulas to structures of fixed size, this will not be necessary for us (see [32] for more information).

2.4 The pebble game

There is a third concept, the bijective pebble game [21], that has a deeper connection to the logic $\mathbf{C}^{k+1}$ and the k-WL. This game is often used to show that graphs cannot be distinguished by k-WL. The game is an Ehrenfeucht-Fraïssé-type game with two players Duplicator and Spoiler. Initially $k+1$ pairs of pebbles, each pair uniquely colored, are placed next to two given input graphs Γ_1, Γ_2. Each round proceeds as follows: Spoiler picks up a pebble pair $(p_i; p_i')$ of pebbles of the same color. Then Duplicator chooses a bijection φ from $V(\Gamma_1)$ to $V(\Gamma_2)$. Then Spoiler places pebble p_i on a vertex $v \in V(\Gamma_1)$ and places p_i' on $\varphi(v)$. Spoiler wins if at any point in time the graph induced by the vertices occupied by pebbles in $V(\Gamma_1)$ is not isomorphic to the graph induced by the vertices occupied by pebbles in $V(\Gamma_2)$ via a map that sends a pebble p_i to its corresponding pebble of the same color p_i' in the other graph. Spoiler also wins (in round 0) if $|V(\Gamma_1)| \neq |V(\Gamma_2)|$.

When using $k+1$ pebbles on two graphs, the game can be won by Spoiler exactly if k-WL distinguishes the graphs [21].

3 WL-type algorithms on groups

As with graphs we would like to be able to study combinatorial properties of finite groups using WL-type algorithms. The natural approach is to adapt the methods from the last section to suit (finite) groups. However, depending on the interpretation of these methods, we will obtain several different choices for initial colorings and refinement strategies for finite groups. We will argue that different methods are all in some sense natural and interesting in their own right. However, the different concepts (possibly) lead to different notions of Weisfeiler-Leman dimension for groups. In contrast to this, for graphs, all notions are equivalent. While we are not able to precisely determine whether for groups the different methods are equally powerful at this point, we do however show that exchanging one method for another changes the dimension by at most a constant factor.

3.1 Weisfeiler-Leman algorithms for groups

The following algorithms define color-refinement procedures on k-tuples of group elements. Since groups are (essentially) ternary relational structures, we will usually require for the dimension that $k \geq 2$. In the following let G be a group. We will define three versions of the WL-algorithm for groups.

Version I:. Define an initial coloring $\chi_0 \colon G^k \to C$ on k-tuples of group elements so that $(g_1, \dots, g_k)$ and $(h_1, \dots, h_k)$ obtain the same color if and only if for all indices $i, j \in \{1, \dots, k\}$ we have $g_i = g_j$ exactly if $h_i = h_j$ and for all indices $i, j, m \in \{1, \dots, k\}$ we have $g_i g_j = g_m$ exactly if $h_i h_j = h_m$. We iteratively define the refinement χ_i in the classical way just like it is defined for graphs, that is for $\bar{g} := (g_1, \dots, g_k) \in G^k$ we have $\chi_i(\bar{g}) :=$

$$(\chi_{i-1}(\bar{g}), \{\!\{(\chi_{i-1}(\bar{g}_{|1\leftarrow x}), \dots, \chi_{i-1}(\bar{g}_{|k\leftarrow x})) \mid x \in G\}\!\}).$$

Version II:. In the definition for graphs, the initial coloring of a tuple takes into account the subgraph induced by the tuple. In analogy to this, one might argue that for groups the initial coloring needs to take into account the subgroup generated by the tuple. Thus, in Version II, we define an initial coloring χ_0 on k-tuples of group elements such that $(g_1, \dots, g_k)$ and $(h_1, \dots, h_k)$ obtain the same color if and only if there is a map with $g_i \mapsto h_i$ which extends to an isomorphism from $\langle g_1, \dots, g_k\rangle$ to $\langle h_1, \dots, h_k\rangle$. In this case we will also say the tuples agree in their *marked isomorphism type*. The iterative refinement is again performed in the classical way.

Version III:. For Version III, we encode groups as graphs, execute the WL-algorithm for graphs and pull back the coloring. For this choose an isomorphism-preserving, invertible functor $\Gamma_\bullet$ that maps finite groups G to finite (simple) graphs Γ_G. Our working example will be the following construction but other choices are certainly possible and many choices will lead to equivalent or at least related algorithms.

To obtain the graph Γ_G (see Figure 2), start with a set of isolated nodes corresponding to elements of the group G. For each pair of group elements (g,h) add a multiplication gadget $M(g,h)$ by adding 4 nodes $a_{gh}, b_{gh}, c_{gh}, d_{gh}$ and add the edges $E(M(g,h)) =$

$$\{\{g,a_{gh}\}, \{h_{gh},b_{gh}\}, \{gh,d_{gh}\}, \{a_{gh},b_{gh}\}, \{b_{gh},c_{gh}\}, \{c_{gh},d_{gh}\}\}.$$

We then use the classical k-dimensional WL-algorithm on the graph Γ_G and pull back the colorings of k-tuples by simply restricting it to G^k.

By construction we have $|\Gamma_G| = \Theta(|G|^2)$ and due to vertex-degrees G is a canonical subset of the vertices of Γ_G. Thus for two groups G, H we have $\Gamma_G \cong \Gamma_H$ if and only if $G \cong H$.

Many other reductions of this form transforming groups to graphs are possible. However, some of them are artificial. For example, one could artificially ensure that the resulting graph has low WL-dimension by precomputing certain isomorphism-invariants not captured by the WL-algorithm.

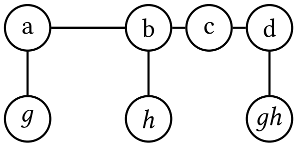

Figure 2. The multiplication gadget to encode the multiplication $g \cdot h = gh$

Thus, a unified treatment of all isomorphism-preserving constructions seems infeasible. However, it seems that for many 'well-behaved' functors the WL-dimension of the constructed graphs differ by a constant factor only. On another note, it would be interesting to obtain efficiently computable subquadratic reductions from groups to graphs, but we are not aware of such a construction.

For fixed k, each version of the WL-algorithm gives rise to a polynomial-time (possibly) partial isomorphism test on pairs of finite groups. Indeed, marked isomorphism of k-tuples in a group G can be checked in time $O(|G| \log(|G|))$ so we can compute initial colors in time $O(|G|^{k+1} \log(|G|))$ for Versions I and II. The refinement steps are the same as for graphs and thus we obtain the same $O(|G|^{k+1} \log(|G|))$ bound for the rest of the computation. For Version III we have a quadratic blowup yielding time $O(|G|^{2k+1} \log(|G|))$.

Definition 3.1. Groups G and H are equivalent with respect to k-WL in Version $J \in \{\text{I,II,III}\}$, in symbols $G \equiv_{WL_k^J} H$, if there is a bijection $f\colon G^k \to H^k$ preserving final colors of the respective color-refinement procedure. Furthermore we write $WL_k^J \preceq WL_{k'}^{J'}$ if it holds that $G \equiv_{WL_{k'}^{J'}} H \Rightarrow G \equiv_{WL_k^J} H$, i.e., the distinguishing power of WL_k^J is weaker than or equal to the distinguishing power of $WL_{k'}^{J'}$.

The main result of this section is that we can exchange one version for another when we multiply the dimension with a constant factor. When studying WL-type algorithms it is often useful to have equivalent pebble games at hand, so we first associate a pebble game to each of the variants above.

3.2 Bijective k-pebble games

We now define suitable pebble games for the different versions. Each of these games is played by two players Spoiler and Duplicator and in each case we will say that Duplicator wins the game if and only if there is a strategy for Duplicator to keep the game going on forever. The board consists of a pair of finite groups G, H of equal order (or rather their elements) or a pair of corresponding graphs Γ_G, Γ_H for Version III, respectively. There are k pairs of pebbles $(p_1; p_1'), \ldots (p_k; p_k')$. We think of pebbles in the same pair as having the same color, and pebbles from different pairs as having distinct colors. The pebbles can be placed beside the board or on the group elements (graph vertices in Version III), in which case we say a group element is pebbled. Pebbles p_i are placed on elements of G (vertices of Γ_G) and pebbles p_i' on H (Γ_H). At any point in time the pebbles $(p_1, \ldots, p_k)$ give us a pebbled tuple in $(G \cup \{\perp\})^k$ (or $(\Gamma_G \cup \{\perp\})^k$), where $\perp$ indicates that the pebble is placed besides the board.

Version I:. All k pairs of pebbles are initially placed beside the board. A round of the game consists of these steps:

1. Spoiler picks up a pair of pebbles $(p_i; p_i')$.
2. Duplicator chooses a bijection $f\colon G \to H$.
3. Spoiler pebbles some element $g \in G$ with p_i, the corresponding pebble p_i' is placed on $f(g)$.

The winning condition is always checked right after Step 1. At that moment, the pebbles not in Spoiler's hand then pebble a k-tuple over $G \cup \{\perp\}$ and a corresponding k-tuple over $H \cup \{\perp\}$. Spoiler wins if the pebbled tuples differ with respect to the initial coloring of Version I. (This implies that no more pebbles are placed beside the graph.)

Version II:. Version II differs from Version I only in that the winning condition uses the initial coloring of Version II rather than Version I. That is, Spoiler wins if the map induced by pairs of pebbles does not extend to an isomorphism between the subgroups generated by the pebbled group elements, and the game continues otherwise.

Version III:. Version III is the (classical) bijective k-pebble game for graphs played on Γ_G and Γ_H (see Subsection 2.4).

In the pebble games, when we say that "Duplicator has to do something", we mean that otherwise Spoiler wins the game. We say that Duplicator respects a certain property of group elements if Duplicator always has to pebble pairs of groups elements which agree in whether they have the property. One can show that Duplicator must respect the partial mapping given by the pairs of pebbles that are currently on the board. Indeed, otherwise Spoiler can win in the next round by pebbling the location where this is violated.

For each game we can also use initial configurations of pebbled tuples instead of starting from empty configurations.

Remark: Color refinement and pebble games are not necessarily restricted to finite groups. While not clear that the results are computable, they still may be of theoretical interest. The same goes for the logics defined next.

3.3 Logics with counting

As for graphs, the k-dimensional refinement on groups can also be interpreted in terms of first-order counting logic.

Recall the central aspects of first order logic. There is a countable set of variables $\{x_1, x_2, \ldots\}$. Formulas are inductively defined so that $x_i = x_j$ is a formula for all pairs of variables and if φ is a formula then $\varphi \wedge \varphi$, $\varphi \vee \varphi$,$\neg\varphi$, $\exists x_i \varphi$ and $\forall x_i \varphi$ are formulas. The semantics are defined in the obvious way. First order logic with counting allows additionally formulas of the form $\exists^{\geq t} x_i \varphi(x_i)$ with the semantic meaning that there are at least t distinct elements that satisfy φ.

To define logics on groups we need to additionally define a relation that relates to the group multiplication.

Version I:. In Version I we add a ternary relation R with which we can create terms of the form $R(x_i, x_j, x_\ell)$. The semantic interpretation is that $R(x_i, x_j, x_\ell)$ holds if $x_i \cdot x_j = x_\ell$. We call $\mathcal{L}_{\text{I}}$ the first order logic with counting on groups arising this way and let $\mathcal{L}^k_{\text{I}}$ be its k-variable fragment.

Version II:. For the logic $\mathcal{L}_{\text{II}}$ we use a different relation to access multiplication: The relation $R(x_{i_1}, x_{i_2}, \ldots, x_{i_t}; w)$ holds, where $w \in (\{x_{i_1}, x_{i_2}, \ldots, x_{i_t}\} \cup \{x^{-1}_{i_1}, x^{-1}_{i_2}, \ldots, x^{-1}_{i_t}\})^*$ is a word in the x_{i_j}, if multiplying the elements according to w gives the trivial element. For example in an abelian group G the relation $R(a, b; aba^{-1}b^{-1})$ would hold for all elements $a, b \in G$. The relation $R(a; aa)$ would only hold if a is the trivial element. We let $\mathcal{L}^k_{\text{II}}$ be the k-variable fragment of the logic. Note that for $\mathcal{L}^k_{\text{II}}$ it actually suffices to use only $k+1$ entries in the relation. In particular we can choose the relation R to be of arity $k+1$.

Version III:. The natural choice of logic for Version III is of course the classical first order logic with counting $\mathbf{C}$ on graphs as discussed in the preliminaries (Subsection 2.3), where we have the relation $E(u, v)$ to encode edges. For notational consistency we define $\mathcal{L}^k_{\text{III}} := \mathbf{C}^k$ to be the k-variable fragment of this logic.

3.4 Equivalence between the different concepts

For each of the versions we have defined, the expressive power will be equal comparing the WL-algorithm, the pebble game, and the corresponding logic. Let us fix groups G and H of the same order. The argument basically follows other well known arguments to show such equivalences (see e.g., [8]). We therefore defer the details to the full version [5].

Theorem 3.2. *Two groups G and H are distinguished by the k-WL-refinement (Version $J \in \{$I,II,III$\}$) if and only if the same holds for the bijective $k+1$-pebble game (Version J).*

Remark: Let us remark on a small detail where the group situation can differ from that of graphs. Note that in our definition of the game, the winning condition is only ever checked after Step 1. We could also check the winning condition when $k+1$ pebble pairs are situated on the group after a round is finished. For this we would need an initial coloring that works with $k+1$ tuples. For graphs this change does not make a difference, since the winning condition only ever depends on 2 pebble pairs. Similarly for Version I, where the winning condition occurs due to 3 pebble pairs, if $k > 3$ then it is irrelevant when we check the winning condition. However, for Version II we are not so sure how the power of the game changes, when altering the moment at which the winning condition is checked.

Theorem 3.3. *$G \not\equiv_{WL^J_k} H$ if and only if there is a sentence in $\mathcal{L}^{k+1}_J$ that holds on one of the groups but not the other.*

3.5 Relationship between the different WL-algorithm versions

Next, we want to relate different versions to each other and we will do so by exploiting the equivalence to pebble games.

Definition 3.4. Consider the k-pebble game on graphs Γ_G and Γ_H and assume that a pair of pebbles is placed on vertices corresponding to multiplication gadgets $M(g_1, g_2)$ and $M(h_1, h_2)$ (but not on vertices corresponding to group elements). Then the pairs (g_1, h_1) and (g_2, h_2) will be called *implicitly pebbled*. Note that implicit pebbles always induce a pairing of group elements.

Intuitively, pebbling a vertex in a multiplication gadget is as strong as pebbling two group elements simultaneously, hence the definition of implicit pebbles. It can be shown that Duplicator has to respect the gadget structure and in particular the multiplication structure of the implicitly pebbled elements. We obtain the following theorem whose proof can be found in the full version [5].

Theorem 3.5. *For all $k \in \mathbb{N}$ we have $WL^{\text{I}}_k \leq WL^{\text{II}}_k \leq WL^{\text{III}}_{k/2+2} \leq WL^{\text{I}}_{k+5}$.*

4 Embedding graphs into finite groups

Next, we describe a construction of finite groups from graphs such that structural properties of the resulting groups are primarily determined by the graphs. We will make this statement more precise in the following. For most proofs we refer to the full version [5]. From now on fix an odd prime p.

Definition 4.1. For each natural number n there is a relatively free group of exponent $p \neq 2$ and (nilpotency) class 2 generated by n elements. It admits a finite presentation

$$F_{n,p} = \langle x_1, \ldots, x_n \mid R(p, n) \rangle$$

where $R(p, n)$ consists of the following relations:

1. For all $1 \leq i \leq n$ there is a relation $x_i^p = 1$, and
2. for all $1 \leq i, j, k \leq n$ there is a relation $[[x_i,x_j],x_k] = 1$.

Thus, the group is generated by $x_1, \ldots, x_n$, each of these generators is an element of order p, and the commutator of two generators commutes with every generator and thus every element of the group. It follows from these properties that elements of $F_{n,p}$ can be uniquely written as

$$x_1^{d_1} \cdot \ldots \cdot x_n^{d_n} [x_1, x_2]^{d_{1,2}} [x_1, x_3]^{d_{1,3}} \cdot \ldots \cdot [x_{n-1}, x_n]^{d_{n-1,n}}$$

where exponents are defined modulo p. In particular, $|F_{n,p}| = p^{n+n(n-1)/2}$.

The main goal is to construct quotients of $F_{n,p}$ using graphs on vertex set $\{1, \ldots, n\}$ as templates in a way that translates combinatorial similarity of the graphs (with respect to Weisfeiler-Leman-refinement) to similar subgroup profiles. We will see that this affects other isomorphism invariants as well.

Definition 4.2. To each (simple, undirected) graph $\Gamma = (\{v_1, \ldots, v_n\}, E)$ and prime number p we assign a finite exponent p group of nilpotency class 2 via

$$G_\Gamma := \left\langle x_1, \ldots, x_n \mid R(p,n), [x_i, x_j] = 1 : \{v_i, v_j\} \in E \right\rangle.$$

Thus, in G_Γ two generators x_i, x_j commute, if the corresponding vertices form an edge in Γ. We usually identify x_i with v_i and use the latter to refer to the vertex as well as the respective element of G_Γ. We fix an order on generators $v_1, \ldots, v_n$ and call these the *standard generators* for G_Γ. The particular presentation above is called the *presentation of* G_Γ *from* Γ.

It turns out that this construction has also been used in other contexts. It is sometimes called *Mekler's construction* in the literature (see [28] for Mekler's original work) and has been primarily investigated for infinite graphs with respect to model theoretic properties. We first collect some possibly well known combinatorial and group theoretic properties.

Lemma 4.3. *We have* $\Phi(G_\Gamma) = G'_\Gamma$ *and the vertices of* Γ *form a generating set of* G_Γ *of minimal cardinality.*

Lemma 4.4. *Denote by* d *the number of non-edges in* Γ. *Then* $G'_\Gamma \cong \mathbb{F}_p^d$, *i.e., the set of non-edges of* Γ *forms a basis in* G'_Γ.

This also gives us normal forms for elements of G_Γ.

Corollary 4.5. *Let* Γ *be a (simple) graph. Then we have* $|G_\Gamma| = p^{|V(\Gamma)|+\left|\binom{V}{2}-E(\Gamma)\right|}$. *In particular, every element of* G_Γ *can be written in the form*

$$v_1^{d_1} \ldots v_n^{d_n} c_1^{d_{n+1}} \ldots c_k^{d_{n+k}}$$

where $\{c_1, \ldots, c_k\}$ *is the set of non-trivial commutators between generators (i.e., the non-edges of the graph* Γ*) and each* d_i *is uniquely determined modulo* p.

We will see that a lot of information on commutation and centralizers can be deduced from Γ directly.

Lemma 4.6. *We have* $Z(G_\Gamma) = G'_\Gamma \times \langle v : N[v] = V(\Gamma)\rangle$. *In particular, if no vertex of* Γ *is adjacent to all other vertices then* $Z(G_\Gamma) = G'_\Gamma$.

From now on let us fix a graph Γ on vertex set $\{v_1, \ldots, v_n\}$ and let $G := G_\Gamma$. We set $m := |\Phi(G)| = |G'|$. Then m is the number of non-edges in Γ and $|G| = p^{m+n}$. Furthermore, fix an ordering of non-trivial commutators $c_1, \ldots, c_m$ of pairs of standard generators $[v_i, v_j] \neq 1$ with $i < j$.

Definition 4.7. Let $x \in G_\Gamma$ be an element with normal form

$$x := v_1^{d_1} \ldots v_n^{d_n} c_1^{e_1} \ldots c_m^{e_m}$$

The **support** of x is $\{v_i \mid d_i \not\equiv_p 0\}$. For a subset of vertices $S \subseteq V(\Gamma)$ let x_S be the subword $v_{i_1}^{d_{i_1}} \ldots v_{i_s}^{d_{i_s}}$ where $S = \{v_{i_1}, \ldots, v_{i_s}\}$ with $i_1 < \cdots < i_s$.

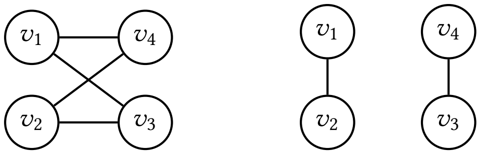

Figure 3. A complete bipartite graph on 4 vertices (left) and its complement (right).

Towards analyzing commutation in G_Γ we consider an example.

Example 4.8. Note that for two connected components C_1, C_2 of the complement graph $\operatorname{co}(\Gamma)$ and every group element $x \in G$ we always have $x_{C_1}x_{C_2} = x_{C_2}x_{C_1}$. Consider the complete bipartite graph Γ on parts $\{v_1, v_2\}$ and $\{v_3, v_4\}$ and its complement $\operatorname{co}(\Gamma)$ (see Figure 3). Then

$$C_{G_\Gamma}(v_1v_2v_3v_4) = \langle v_1v_2\rangle\langle v_3v_4\rangle Z(G_\Gamma).$$

The following theorem states that the example essentially captures how commutation works in general.

Lemma 4.9. *For* $x \in G_\Gamma$ *let* $C_1, \ldots, C_s$ *be the connected components of* $\operatorname{co}(\Gamma[\operatorname{supp}(x)])$. *Then* $x = x_{C_1} \cdots x_{C_s} c$ *with* $c \in G'_\Gamma \leq Z(G_\Gamma)$ *and* $y \in G_\Gamma$ *commutes with* x *if and only if* $y \in \langle x_{C_1}\rangle \cdots \langle x_{C_s}\rangle \cdot \langle w : [v,w] = 1 \text{ for all } v \in \operatorname{supp}(x)\rangle G'_\Gamma$.

Corollary 4.10. *Let* $x = v_{i_1}^{d_1} \ldots v_{i_r}^{d_r} c$ *with* $i_1 < i_2 < \cdots < i_r$, c *central in* G_Γ *and* $d_i \not\equiv_p 0$ *for all* i. *Then*

$$C_{G_\Gamma}(x) = \langle x_{C_1}\rangle \ldots \langle x_{C_s}\rangle \left\langle \{v_m \mid [v_m, v_{i_j}] = 1 \text{ for all } j\}\right\rangle G'_\Gamma.$$

Where, $C_1, \ldots, C_s$ *are the connected components of the complement graph* $\operatorname{co}(\Gamma[\operatorname{supp}(x)])$.

This (almost) distinguishes single support vertices.

Lemma 4.11. *For* $x \in G_\Gamma$ *and* $v \in \operatorname{supp}(x)$ *we have that* $|C_{G_\Gamma}(x)| \leq |C_{G_\Gamma}(v)|$. *Set* $M(x) := \{w \in V(\Gamma) \mid [w,y] = 1 \text{ for all } y \in \operatorname{supp}(x)\}$. *Then if* $|C_{G_\Gamma}(x)| = |C_{G_\Gamma}(v)|$ *either* $M(x) = M(v)$ *in which case* $\Gamma[\operatorname{supp}(x)]$ *is a complete graph, or* $M(x) = M(v) \setminus \{v\}$ *and in both cases all components of* $\operatorname{co}(\Gamma[\operatorname{supp}(x)])$ *not containing* v *are singletons.*

This means that elements of the form vz with $v \in V(\Gamma)$ and $z \in G'_\Gamma$ are almost canonical in G_Γ in the following sense: Define a set C as the union of all minimal generating sets $\{g_1, \ldots, g_n\}$ of G_Γ (so $n = |V(\Gamma)|$) for which the value of $\sum_i |C_{G_\Gamma}(g_i)|$ is maximal among minimal generating sets of G_Γ. Then C contains $V(\Gamma)$ since $V(\Gamma)$ is such a generating set itself. Furthermore C is canonical in G_Γ (invariant under all automorphisms) and we can use it to analyze isomorphisms.

In the following part we want to compare different groups presented on graphs. Let us fix graphs Γ_1 and Γ_2 on the vertex set $\{v_1, \ldots, v_n\}$ with edges given by E_1 and E_2 and corresponding groups $G_i := G_{\Gamma_i}$. The standard generators on which the G_i are presented will again be called $(v_j)_{1\leq j\leq n}$.

Theorem 4.12. *It holds that* $\Gamma_1 \cong \Gamma_2$ *if and only if* $G_1 \cong G_2$.

Proof. Let $\varphi\colon \Gamma_1 \to \Gamma_2$ be a graph isomorphism. Then φ induces an automorphism of $F_{n,p}$ by permuting generators and we have $G_{\Gamma_i} = F_{n,p}/N_i$ where N_i is the central subgroup generated by edges of Γ_i. Thus, as a group automorphism, φ maps N_1 to N_2 giving an isomorphism of the corresponding quotients.

For the other direction consider a group isomorphism $\varphi : G_1 \to G_2$. From Lemma 4.11 we see that for $x \in G_i$ and $v \in \mathrm{supp}(x)$ we have

$$\circledast:\ |C_{G_i}(x)| \leq |C_{G_i}(v)|.$$

As in the last lemma let $M(x) := \{v \in V(\Gamma_i) \mid [v, w] = 1 \text{ for all } w \in \mathrm{supp}(x)\}$ for $x \in G$ be the set of standard generators commuting with the entire support of x. In fact $M(x) = \bigcap_{w \in \mathrm{supp}(x)} N[w]$.

Our strategy is now to alter the group isomorphism $\varphi : G_1 \to G_2$ until we can extract sufficiently much information on the graphs. We do so by redefining the images $y_i = \varphi(v_i)$ and double checking that the new map is still a homomorphism onto a generating set and thus an isomorphism.

Consider the case that $y := y_j$ is supported in G_2 by more than one vertex for some index $j \leq n$. There must be some vertex $v \in \mathrm{supp}(y)$ such that replacing y with v still leaves us with a generating set for G_2. Indeed, this is true in the elementary abelian group $G_2/(G_2)'$ and commutators are non-generators in G_2. Furthermore, from $\circledast$ it follows that $(v_1, \ldots, v_n)$ is a generating set of G_1 which maximizes the sum of centralizer orders $\sum_i |C_{G_1}(v_i)|$ among minimal generating sets and since φ is an isomorphism, the same must be true for $(y_1, \ldots, y_n)$ in G_2. For $i > 1$, consider y_i such that $[y, y_i] = 1$. From Corollary 4.10 we see that (up to multiplication with commutators which can be ignored) $y_i = y_{C_1}^{t_1} \ldots y_{C_s}^{t_s} v_{i_1}^{e_1} \ldots v_{i_k}^{e_k}$ for some vertices $v_{i_j} \in M(y)$ and where $C_1, \ldots C_s$ are the components of $\mathrm{co}(\Gamma[\mathrm{supp}(y)])$ and we also get that $[y_i, v] = [y_{C_1}^{t_1}, v]$ where we, w.l.o.g., assume that $v \in C_1$. The last Corollary furthermore shows that $|C_i| = 1$ for $i > 1$, so actually we can write $y_i = y_{C_1}^{t_1} v_{i_1}^{e_1} \ldots v_{i_{k'}}^{e_k}$ for $v_{i_j} \in M(y)$. Using the same argument as for y and v there is some $w \in \mathrm{supp}(y_i)$ such that y_i can be replaced with w while still keeping a generating set and for this w we again have $|C_{G_2}(y_i)| = |C_{G_2}(w)|$. Also note that if $|C_1| = 1$ then $[v, y_i] = 1$ which is what we want to show. Similarly we are done if $t_1 \equiv_p 0$, so assume otherwise. If $|C_1| > 1$ there is some $v' \in C_1$ such that $[v, v'] \neq 1$ and in particular $v, v' \notin M(y_i)$ implying that $v, v' \in \mathrm{supp}(y_i)$ from the expression for y_i above. Now w can be chosen such that $w \notin C_1$ (since the exponents of y and y_i over elements of C_i agree this follows from rank considerations and the fact that $(y_1 G_2', \ldots, y_n G_2')$ forms a basis of G_2/G_2'). Thus $M(y_i) \subseteq M(w) \setminus \{v, v'\}$ contradicting the previous Corollary.

In conclusion, $[y, y_i] = 1$ implies $[v, y_i] = 1$ (And we even see that this only happens if $\mathrm{supp}(y)$ induces a complete graph or if $\mathrm{supp}(y_i) \cap C_1 = \emptyset$). Hence exchanging y for v gives us a generating set which is still a valid image of $(v_1, \ldots, v_n)$. We can iterate this process to obtain an isomorphism mapping vertices to elements supported by single vertices as well which gives rise to a bijection between vertices. The fact that the isomorphism respects commutators then translates to respecting edges of the graphs and we conclude that $\Gamma_1 \cong \Gamma_2$. □

It is not always the case that the original vertices $V(\Gamma)$ of the graph form a canonical subset of G_Γ. However, we can precisely describe the conditions under which they do.

Lemma 4.13. *Assume $\Gamma_1 \cong \Gamma_2$, then Γ_i is canonical in G_i, if and only if in Γ_1 (and thus Γ_2) there is no pair of distinct vertices v, w with $N(v) \subseteq N[w]$. If this is the case, there is a bijection between* $\mathrm{Iso}(\Gamma_1, \Gamma_2)$ *and* $\mathrm{Iso}(G_1, G_2)$.

5 Constructing groups with equal k-profiles

In this section we want to apply the construction from above to specific graphs. The idea is to start with a family of 3-regular base graphs such that the CFI-construction gives us two non-isomorphic graphs Γ_1 and Γ_2 for each of the base graphs which can be distinguished by k-WL only for k scaling linearly with the size of the CFI-graphs. We will then show that the resulting groups $G_i := G_{\Gamma_i}$ have equal $\Theta(k)$-profiles.

Definition 5.1. For a group G, a tuple $(g_1, \ldots, g_k) \in G^k$ is *minimal* if $\langle g_1, \ldots, g_k \rangle$ is not generated by $k - 1$ elements.

When working with $F := F_{n,p}$ we will fix a standard basis for $Z(F) = \Phi(F) \cong \mathbb{F}_p^{\binom{n}{2}}$. If F is presented on generators $v_1, \ldots, v_n$ we choose

$$([v_1, v_2], [v_1, v_3], \ldots, [v_1, v_n], [v_2, v_3], \ldots, [v_{n-1}, v_n])$$

as our fixed basis for the center of F. We call these commutators the *standard commutators*.

Definition 5.2. Let $\bar{g} := (g_1, \ldots, g_k) \in F_{n,p}^k$. We define two $\mathbb{F}_p$-matrices. In the $(k \times n)$-matrix $B_1(\bar{g})$ the i-th row corresponds to (the exponents of) g_i expressed in normal form in terms of standard generators. In the $\left(\binom{k}{2} \times \binom{n}{2}\right)$-matrix $B_2(\bar{g})$ the rows correspond to $[g_1, g_2], [g_1, g_3], \ldots, [g_{k-1}, g_k]$ expressed in terms of standard commutators in this order. We will sometimes refer to their columns by these labels, i.e., the column belonging to $[v_i, v_j]$ will be referenced as $B_2(\bar{g})([v_i, v_j])$.

Example 5.3. For example assume $n = 3$ and $k = 2$ and assume $\bar{g} := (g_1, g_2)$ with $g_1 = v_1 v_2^5 v_3$ and $g_2 = v_1^2 v_2 [v_1, v_2]$. Then $[g_1, g_2] =$

$$[v_3, v_1^2][v_2^5, v_1^2][v_3, v_2][v_1, v_2] = [v_1, v_2]^{-9}[v_1, v_3]^{-2}[v_2, v_3]^{-1}.$$

In this case $B_1(\bar{g}) = \begin{pmatrix} 1 & 5 & 1 \\ 2 & 1 & 0 \end{pmatrix}$ and $B_2(\bar{g}) = (-9, -2, -1)$

where entries are to be read modulo p.

Lemma 5.4. *Let $\bar{g} := (g_1, \dots, g_t) \in (F_{n,p})^t$. Then $B_2(\bar{g}) = B_1(\bar{g}) \wedge B_1(\bar{g})$ where $\wedge$ describes the exterior product with respect to our chosen orderings for the standard bases.*

Proof. Express the commutator $c_{i,j} := [g_i, g_j]$ in terms of the standard commutators. Then $c_{i,j} = \left([v_k, v_\ell]^{m(k,\ell)}\right)_{k<\ell}$ where $m(k,\ell) = (B_1(\bar{g}))_{i,k}(B_1(\bar{g}))_{j,\ell} - (B_1(\bar{g}))_{i,\ell}(B_1(\bar{g}))_{j,k} =$

$$\det\begin{pmatrix} (B_1(\bar{g}))_{i,k} & (B_1(\bar{g}))_{i,\ell} \\ (B_1(\bar{g}))_{j,k} & (B_1(\bar{g}))_{j,\ell} \end{pmatrix}.$$

Thus, the row of $B_2(\bar{g})$ belonging to $c_{i,j}$ corresponds to the row of $B_1(\bar{g}) \wedge B_1(\bar{g})$ belonging to rows $B_1(\bar{g})_{i,-}$ and $B_1(\bar{g})_{j,-}$ □

In particular, this shows that subgroups of $F_{n,p}$ are direct products of relatively free groups and central groups. In the following we will use the fact that for $M \in \mathbb{F}_p^{k\times n}$ we have $\operatorname{rank}(M \wedge M) = \binom{\operatorname{rank}(M)}{2}$, see for example [9, Section 10.1].

Lemma 5.5. *Let $G \leq F_{n,p}$ be generated by $\bar{g} := (g_1, \dots, g_t)$ and set $r := \operatorname{rank}(B_1(\bar{g}))$. Then there are r elements g_{i_j} among $\{g_1, \dots, g_t\}$ and central elements $c_1, \dots, c_k \in Z(F_{n,p})$ for some $0 \leq k \leq (n-r)$ such that $G = \langle g_{i_1}, \dots, g_{i_r}\rangle \times \langle c_1, \dots, c_k\rangle$. Furthermore, G' has $\mathbb{F}_p$-dimension $\binom{r}{2}$.*

Proof. If $B_1(\bar{g})$ has rank r, we can choose r linearly independent rows corresponding to certain generators g_{i_j}. Other rows can then be expressed via these chosen rows which by definition of $B_1(\bar{g})$ means that all other generators can be replaced by central elements $c_1, \dots, c_{n-r}$ without changing G. Set $G_r := \langle g_{i_1}, \dots, g_{i_r}\rangle$. The corresponding rows in $B_1(\bar{g})$ are now independent meaning that no set of cardinality less than r can generate G_r. Since all other generators are now central we have $[G, G] = [G_r, G_r]$ and the latter is of dimension $\operatorname{rank}(B_2(\bar{g})) = \binom{r}{2}$. Choose a subset of c_i's that is maximal with respect to the property $G_r \cap \langle c_{i_1}, \dots, c_{i_k}\rangle = \emptyset$. Then $G = \langle G_r, c_{i_1}, \dots, c_{i_k}\rangle$ as desired. □

The following observation is elementary but will help us compare subgroups of G_Γ for different values of Γ.

Lemma 5.6. *Let $H := \langle g_1, \dots, g_t, z_1, \dots, z_r\rangle \leq G_\Gamma$ and $R := \dim(\Phi(H))$. Assume that all z_i are central in G_Γ, that $\langle g_1, \dots, g_t\rangle / Z(G_\Gamma) \cong \mathbb{F}_p^t$, and that H is not generated by less than $t + r$ elements. Let $c_1, \dots, c_R$ be generators of $\Phi(H)$ of the form $c_i = [g_{i_1}, g_{i_2}]$ and express all other commutators $c_{R+1}, \dots, c_{\binom{t}{2}}$ between the g_i as words $w_{R+1}, \dots, w_{\binom{t}{2}}$ in the c_i. Then*

$$H \cong \langle g_1, \dots, g_t \mid \text{exponent } p, \text{class } 2\ , w_{R+1}, \dots, w_{\binom{t}{2}}\rangle \times C_p^r.$$

Proof. By assumption $|\langle g_1, \dots, g_t\rangle| = p^{t+R}$. Clearly the presentation above defines a group admitting an epimorphism onto $\langle g_1, \dots, g_t\rangle$. Due to the given relations its order is at most p^{t+R}. Since $(g_1, \dots, g_t, z_1, \dots, z_r)$ is assumed to be minimal, the central group $\langle z_1, \dots, z_r\rangle \cong C_p^r$ splits from H. □

Let $\Gamma_0 = (\{V_1, \dots, V_t\}, E)$ be a 3-regular graph with $N := |E|$ edges and such that $\Gamma_1 := \mathrm{CFI}(\Gamma_0)$ and $\Gamma_2 := \widetilde{\mathrm{CFI}(\Gamma_0)}$ are not isomorphic (cf. Theorem 2.1). Let $n := 10t$ be the number of vertices of Γ_1 and Γ_2.

To improve readability, we use capital letters for the vertices of the base graph in the following.

We assume Γ_0 and $\mathrm{co}(\Gamma_0)$ to be connected and then the same holds for the corresponding CFI-graphs. Recall that the CFI-graphs are again 3-regular. In the following we will call a pair of edges between two CFI-gadgets together with their adjacent vertices a *link* and *twisting* will be understood as replacing the edges in a link with their twisted version. Note that two gadgets or two links are always disjoint or equal and that links correspond bijectively to edges in the base graph Γ_0. As before we call vertices of links external (w.r.t. their gadget) and other vertices internal. We fix $F := F(n, p)$, the relatively free group on vertices of the CFI-graphs above. We also fix normal subgroups $N_1, N_2 \leq F$ corresponding to edges of Γ_1 and Γ_2, respectively. Thus $G_i := G_{\Gamma_i} = F/N_i$. Finally, let e be any edge in the base graph and let ${}^{(e)}\colon F \to F$ be the following map: Say $e = (V, W) \in E(\Gamma_0)$ (so we actually chose an orientation). Then twisting along e can be seen as swapping in all normal forms the standard commutators $[a_i^V, a_j^W]$ and $[a_i^V, b_j^W]$ and also swapping all occurrences of $[b_i^V, b_j^W]$ and $[b_i^V, a_j^W]$.

This is of course not a group isomorphism but it induces an automorphism $\varphi\colon Z(F) \to Z(F)$. If $x \in F$ has a normal form that factors as vc where v is the part of x in standard generators and c is the product of standard commutators then $x^{(e)} := v\varphi(c)$ and this defines a bijection of F into itself.

Definition 5.7. A group $H \leq F$ is *essentially k-generated* if

1. $F' = Z(F) \leq H$ and
2. $\dim_{\mathbb{F}_p}(H/F') = k$.

Intuitively this means that the group is k generated modulo the center. Define $\mathcal{H}_k \subseteq \mathrm{Sub}(F)$ to be the set of all essentially k-generated subgroups of F.

Lemma 5.8. *For every subgroup $S \leq G_i = F/N_i$ for which $\dim_{\mathbb{F}_p}(S/G_i') = k$ there is a unique essentially k-generated subgroup $H \leq F$ such that $S \leq H/N_i$.*

Proof. Let $\nu : F \to F/N_i$ be the natural epimorphism then H can be uniquely defined as $\nu^{-1}(S)F'$. □

Set $\mathcal{H}_k^{N_i} := \{H/N_i \mid H \in \mathcal{H}_k\}$. Our goal is for various k to construct a bijection $\mathcal{H}_k^{N_1} \to \mathcal{H}_k^{N_2}$ that preserves isomorphism-types of groups. Since all k-generated subgroups have the property that $\dim_{\mathbb{F}_p}(S/G_i') = k$, the lemma above then gives an isomorphism-type preserving bijection between k-generated subgroups of G_1 and G_2. Note that $\mathcal{H}_k^{N_i} = \{S \leq G_i \mid \dim_{\mathbb{F}_p}(S/G_i') = k \text{ and } G_i' = Z(G_i) \leq S\}$.

Lemma 5.9. *Let $1 \leq k < N/10$ where N is the number of edges in Γ_0. For $H \in \mathcal{H}_k$ there is some edge e in the base graph such that $H/N_1 \cong H/N_1^{(e)}$.*

Proof. Let $H := \langle f_1, \dots, f_\ell \rangle$ and for each i set $g_i := f_i N_1 \in G_1$. We want to investigate the group $(H/N_1)' = H'/N_1$. Since it is generated by commutators between the g_i its structure is mostly described by $B_2(\bar{f}) = B_1(\bar{f})^{\wedge 2}$ after replacing columns indexed by elements of N_1 with zero-columns. Call this new matrix $B_2(\bar{g})$.

Twisting along edge $e = (V, W) \leq E(\Gamma_0)$ results in mapping $([a_i^V, a_j^W], [b_i^V, b_j^W])$ to $([a_i^V, b_j^W], [b_i^V, a_j^W])$ (and vice versa, see Section 2.2). This can also be interpreted in terms of the matrices from above as replacing the two zero-columns $B_2(\bar{g})([a_i^V, a_j^W])$ and $B_2(\bar{g})([b_i^V, b_j^W])$ by the original columns in $B_2(\bar{f})$ and by replacing the columns corresponding to $[a_i^V, b_j^W]$ and $[b_i^V, a_j^W]$ with zero-columns instead. This defines a matrix $B_2(\bar{g}^{(e)})$ that describes linear dependencies between commutators among the $(f_1^{(e)} \dots, f_k^{(e)})$ modulo $N_1^{(e)}$.

We will now argue that e can be chosen in such a way that $B_2(\bar{g})$ and $B_2(\bar{g}^{(e)})$ have the same column spaces. For this, we argue that we can fix a system of columns of rank r in $B_2(\bar{g})$ that does not contain the columns affected by twisting along e, then for e as above these columns also form a system of maximal rank in $B_2(\bar{g}^{(e)})$ and thus linear dependency relations for rows of the two matrices are exactly the same. Using Lemma 5.6 we see that $H/N_1 \cong H/N_1^{(e)}$ for this choice of e.

By assumption the rank of $B_1(\bar{f})$ is k and $k < N$. We assume w.l.o.g. that the first k columns of $B_1(\bar{f})$ are linearly independent. Then the same holds for the first $\binom{k}{2}$ columns in $B_2(\bar{f}) = B_1(\bar{f}) \wedge B_1(\bar{f})$. Now these columns may not contain a system of full rank anymore in $B_2(\bar{g})$ but they belong to commutators of the form $[i, j]$ for $1 \leq i < j \leq k$. Since Γ_i is 3-regular, for a fixed i at most three of these commutators are contained in N_1. Thus the rank of the first $\binom{k}{2}$ columns in $B_2(\bar{g})$ is at least $\binom{k}{2} - 3k$ and we may choose $r' \leq 3k$ additional columns such that they contain a system of full rank together with the first $\binom{k}{2}$ columns. Now every such column belongs to a pair of vertices and the number of relevant vertices for the full rank system in total is smaller than $2r' + k \leq 7k < N$ and thus there are still links in Γ_1 that are not adjacent to any of these vertices. Let us say these links correspond at least to edges $e_1, \dots, e_{N-7k}$. For each of these links there are two zero-columns in $B_2(\bar{g})$ and two columns agreeing with $B_2(\bar{f})$ corresponding to the twisted/non-twisted version of this link. Due to the choice of the edges we can now replace all four of these columns by zero-columns without reducing the rank of the resulting matrix. We will argue that among the edges $e_1, \dots, e_{N-7k}$ there are some edges where twisting also does not change the rank.

For this, note that for vertices v, w, column $(B_2(\bar{f}))([v, w])$ is a linear combination of columns $(B_2(\bar{f}))([v, y])$ and also a linear combination of columns $(B_2(\bar{f}))([y, w])$ where y runs through the first k columns of $B_1(\bar{f})$ since we assumed the first k columns of $B_1(\bar{f})$ to be linearly independent and since the entries of $B_2(\bar{f})$ are subdeterminants of $B_1(\bar{f})$. Say the first k columns of $B_1(\bar{f})$ correspond to vertices $v_1, \dots, v_k$ in the CFI-graphs. We say that $1 \leq i \leq k$ is *bad* for some link if v_i is adjacent to this link. Since each index is bad for at most three links and $3k < N - 7k$, there exist links over the edges $e_1, \dots, e_{N-7k}$ for which no index is bad. For such a link, belonging to edge e say, all columns in the linear combination described above are still present in $B_2(\bar{g})$ and thus the rank of this matrix is the same as for $B_2(\bar{g}^{(e)})$. □

Definition 5.10. Set $\mathcal{V} := V(\Gamma_1)$ and identify $\mathrm{Sym}(\mathcal{V})$ as a subgroup of $\mathrm{Aut}(F)$ in the natural way. We set A to be the group of permutations of $\mathrm{Sym}(\mathcal{V})$ that map each gadget to itself with an automorphism. (I.e., A consists of the graph automorphisms after link edges have been removed.)

Note that the group A is abelian. It is generated by the permutations of $\mathcal{V}$ twisting two incident links in Γ_1 while permuting the inner vertices of their common gadget accordingly to a graph automorphism of the gadget. In particular, A stabilizes all links and gadgets setwise.

If $H \in \mathcal{H}_k$ then for any edge e of Γ_0 we have $H^{(e)} = H$ (even if ${}^{(e)}$ is not a group isomorphism). Lemma 5.9 shows that for $H/N_1 \leq G_1$ there is some edge e of Γ_0 such that $H/N_1 \cong H/N_1^{(e)}$ and by the properties of the CFI-construction the twist ${}^{(e)}$ can be altered to become the original twist via suitable elements from A. More precisely, in the situation above there is some $\sigma_e \in A$ (only depending on e) such that $H/N_1^{(e)} \cong \sigma_e(H_1)/N_2$. This defines an isomorphism-type preserving map

$\Phi : \mathcal{H}_k^{N_1} \to \mathcal{H}_k^{N_2},\ H/N_1 \mapsto \sigma_e(H_1)/N_2$ where e depends on H and we will show that the edges can be chosen in a way that makes Φ bijective.

Definition 5.11. Let $i \in \{1, 2\}$. We say that subgroups $H_1/N_i, H_2/N_i \in \mathcal{H}_k^{N_i}$ are of the same *type* if there is some $\sigma \in A$ such that $H_1 = \sigma(H_2)$.

An inspection of Lemma 5.9's proof shows the choice of edge e only depends on the type of the subgroups involved.

Lemma 5.12. *If $k < N/10$, the edge e in Lemma 5.9 can be chosen to be the same for all subgroups of a fixed type.*

Proof. Since A fixes links setwise, positions where twisting preserves the isomorphism type are the same for groups that get mapped to each other via elements from A. □

Lemma 5.13. *For each edge e compatible with Lemma 5.9, Φ maps subgroups of different types to subgroups of different types.*

Proof. Assume that 5.9 gives edges e_1 and e_2 for groups $S, \tilde{S} \leq G_1$. Write $S = H/N_1$, $\tilde{S} = \tilde{H}/N_1$ and assume that $\sigma_{e_1}(H)/N_2$ and $\sigma_{e_2}(\tilde{H})/N_2$ have the same type. Then there is some $\sigma \in A$ with $(\sigma_{e_2}^{-1}\sigma\sigma_{e_1})(H) = \tilde{H}$ and thus S and $\tilde{S}$ have the same type. □

Lemma 5.14. *For a fixed type and a fixed edge e (as in Lemma 5.9), Φ is isomorphism-type preserving and injective.*

Proof. Keep the notation from the last lemma but assume $S_1 \neq S_2$ are of the same type. Then $H_1 \neq H_2$. Thus $\sigma_e(H_1) \neq \sigma_e(H_2)$ which is equivalent to $\sigma_e(H_1)/N_2 \neq \sigma_e(H_2)/N_2$ due to $\sigma_e(H_i)$ containing $Z(F)$ and in particular N_2. □

All arguments also work for interchanged roles of G_1 and G_2. In particular this shows that $|\mathcal{H}_k^{N_1}| = |\mathcal{H}_k^{N_2}|$ for each k.

Corollary 5.15. *G_1, G_2 have equal k-profiles for $k < N/10$.*

Proof. Since bijection Φ is isomorphism-type preserving, the collection of subgroups in $\mathcal{H}_k^{N_1}$ is mapped bijectively and isomorphism-type preservingly to $\mathcal{H}_k^{N_2}$. Every k-generated subgroup is contained in a unique factor of an essentially k-generated subgroup (Lemma 5.8) so this induces a bijection from k-generated subgroups to k-generated subgroups. □

For the commuting graphs of G_1 and G_2, note that non-central elements in G_1 that are not powers of one another cannot commute if one of the elements has a support of 4 or larger. Whether the Weisfeiler-Leman algorithm of a particular dimension distinguishes the graphs therefore does not change when restricting the commuting graphs to group elements with support size at most 3. In particular, the commuting graphs cannot be distinguished by the $O(\log(n))$-dimensional Weisfeiler-Leman algorithm.

6 The Weisfeiler-Leman Dimension of groups constructed from CFI-graphs is 3

In the previous section we constructed groups $G_i := G_{\Gamma_i}$ based on two CFI-graphs Γ_1 and Γ_2. The groups agree in terms of traditional group theoretical invariants (such as exponent, nilpotency class, and the combinatorics of their conjugacy classes) and also with respect to their k generated subgroups for large k. On first sight this might indicate that these groups should be hard to distinguish by combinatorial means but as we will see in this section their WL-dimension is only 3. Throughout this section we exclusively use WL-algorithms and pebble games of Version II. The main theorem of this section is the following.

Theorem 6.1. *Let Γ_0 be a 3-regular connected graph and let $\Gamma_1 := \mathrm{CFI}(\Gamma_0)$ and $\Gamma_2 := \widetilde{\mathrm{CFI}(\Gamma_0)}$ be the corresponding CFI-graphs. The 3-dimensional WL-algorithm distinguishes G_1 from G_2. If additionally Γ_0 has (graph) WL-dimension at most 3 then 3-dimensional WL-algorithm identifies G_1 as well as G_2.*

Requiring that Γ_0 has WL-dimension at most 3 is not a severe restriction (Observation 2.1). Towards proving the theorem we collect several observations on the pebble game that are particular to the groups arising from CFI-graphs. The proofs of these observations can be found in the full version [5].

Lemma 6.2. *For each $k \geq 3$, throughout the k-pebble game on G_1 and G_2 Duplicator has to choose bijections that respect the set of elements with single-vertex support $\{x \mid |\operatorname{supp}(x)| = 1\}$. Moreover $\operatorname{supp}(x) = \operatorname{supp}(y)$ and $|\operatorname{supp}(x)| = |\operatorname{supp}(y)| = 1$ must imply $\operatorname{supp}(f(x)) = \operatorname{supp}(f(y))$.*

Lemma 6.3. *If Duplicator does not respect support sizes at some point then Spoiler can win with three pebbles.*

Lemma 6.4. *For each $k \geq 4$, throughout the k-pebble game on G_1 and G_2 Duplicator has to choose bijections respecting internal vertices and gadgets of the underlying CFI-graphs. Here, elements corresponding to a gadget vertex v are all elements of $vZ(G_i)$. Moreover pairs of vertices lying in a common gadget have to be mapped to pairs in a common gadget.*

Using these observation we can finally prove the theorem.

Proof of Theorem 6.1. We first define a set $\mathcal{V}$ of special vertices in Γ_1: For each gadget put exactly one internal vertex in $\mathcal{V}$ and add all adjacent external vertices. Let $v \in G_1$ denote the ordered product of all vertices in $\mathcal{V}$. By Lemma 6.3 Duplicator must choose a bijection for which $f(v)$ has the same support size as v. Spoiler puts a pebble on v. The Lemma furthermore shows that all future bijections have to map $\operatorname{supp}(v)$ to $\operatorname{supp}(f(v))$ or otherwise Spoiler can pebble some $v_i \in \operatorname{supp}(v)$ with $f(v_i) \notin \operatorname{supp}(f(v))$ and Duplicator will not be able to respect support sizes from here on. Using Lemma 6.4 we see that $\operatorname{supp}(f(v)) =: \mathcal{V}'$ has to be composed exactly as $\operatorname{supp}(v) = \mathcal{V}$, that is, $\mathcal{V}'$ can also be constructed by choosing set of internal vertices, one per gadget, and adding all their adjacent external vertices. The set $\mathcal{V}$ induces a subgraph of Γ_1 and similarly $\mathcal{V}'$ induces a subgraph of Γ_2. We argue these subgraphs have a different number of edges modulo 2. For this observe the following: if we alter $\mathcal{V}$ by replacing one internal vertex with another one in the same gadget, this changes exactly two neighbors among the external vertices. The new induced subgraph differs then in exactly two locations of two different links. Thus the number of edges in the induced subgraph remains the same modulo 2. By induction this is true for all possible choices of $\mathcal{V}$. We can thus assume that $\mathcal{V} = \mathcal{V}'$. However, this implies that $\Gamma_1[\mathcal{V}]$ and $\Gamma_2[\mathcal{V}']$ disagree in exactly one edge, namely at the twisted link. This shows the graphs have a different number of edges modulo 2.

However, we already argued that Duplicator has to map $\mathcal{V}$ to $\mathcal{V}'$. Since the number of edges of $\Gamma_1[\mathcal{V}]$ and $\Gamma_2[\mathcal{V}']$

disagree, for any suitable bijection some vertex is mapped to a vertex of incorrect degree, which can be exploited by Spoiler. This shows G_1 can be distinguished from G_2.

We assume now that additionally the base graph Γ_0 has Weisfeiler-Leman dimension at most 3. Suppose that G is any group with $|G| = |G_1|$ that is indistinguishable from G_1. The vertices of Γ_1 form a canonical copy of Γ_1 inside of G_1 (up to central elements), so there must be a corresponding set in G as well. If the induced commutation graph Γ on this set is distinguishable from Γ_1 then G_1 is distinguishable from G. From the commutation graph, we can reconstruct a corresponding base graph Γ. Which must be indistinguishable by 3-WL from Γ_0. This means it is isomorphic to Γ_0 since its Weisfeiler-Leman dimension is at most 3. Thus Γ is isomorphic to Γ_1 or Γ_2. This gives a presentation of G isomorphic to a presentation of G_1 or G_2. □

7 Conclusion

We defined several versions of the Weisfeiler-Leman algorithm for groups and showed their dimension concepts are linearly related. We then gave a construction of groups from graphs that preserves isomorphism. We can recover combinatorics of the original graph by analyzing commutation in the groups. This allowed us to construct pairs of non-isomorphic groups with the same k-subgroup profiles for k logarithmic in the order of the groups. These groups are nevertheless identified by the 3-dimensional Weisfeiler Leman algorithm.

The strategy in the pebble we employed to show this exploits that by pebbling products of standard generators Spoiler can essentially force Duplicator to fix an arbitrary set of standard generators simultaneously. Abstractly, on graphs or groups, one could define a new pebble game where pebbles can be placed on sets of vertices. Spoiler now additionally wins the game if the subset relations between pebbled sets disagree in the two structures. This game corresponds to a monadic second order logic where there is still a bound on the number of variables that may be used. While they seem to resemble each other, we are not sure what the precise relationship between this game and the pebble games on groups is. It also seems to be unknown what the expressive power of this game (or the corresponding logic) is.

On another note, it also remains a central open question whether k-WL solves the Group Isomorphism Problem for some constant dimension k. While a positive answer would place Group Isomorphism in polynomial time, a negative answer would prove the existence of groups of unbounded Weisfeiler-Leman dimension which would provide interesting examples of groups that are even harder to distinguish.

References

[1] László Babai. 2016. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, Daniel Wichs and Yishay Mansour (Eds.). ACM, 684–697. https://doi.org/10.1145/2897518.2897542

[2] László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. 2011. Code Equivalence and Group Isomorphism. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, Dana Randall (Ed.). SIAM, 1395–1408. https://doi.org/10.1137/1.9781611973082.107

[3] László Babai, Paolo Codenotti, and Youming Qiao. 2012. Polynomial-Time Isomorphism Test for Groups with No Abelian Normal Subgroups - (Extended Abstract). In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I (Lecture Notes in Computer Science)*, Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer (Eds.), Vol. 7391. Springer, 51–62. https://doi.org/10.1007/978-3-642-31594-7_5

[4] László Babai and Youming Qiao. 2012. Polynomial-time Isomorphism Test for Groups with Abelian Sylow Towers. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France (LIPIcs)*, Christoph Dürr and Thomas Wilke (Eds.), Vol. 14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 453–464. https://doi.org/10.4230/LIPIcs.STACS.2012.453

[5] Jendrik Brachter and Pascal Schweitzer. 2020. On the Weisfeiler-Leman Dimension of Finite Groups. *CoRR* abs/2003.13745 (2020). arXiv:arXiv:2003.13745 arXiv.

[6] Peter A. Brooksbank, Joshua A. Grochow, Yinan Li, Youming Qiao, and James B. Wilson. 2019. Incorporating Weisfeiler-Leman into algorithms for group isomorphism. *CoRR* abs/1905.02518 (2019). arXiv:1905.02518 http://arxiv.org/abs/1905.02518 arXiv.

[7] Peter A. Brooksbank, Joshua Maglione, and James B. Wilson. 2017. A fast isomorphism test for groups whose Lie algebra has genus 2. *J. Algebra* 473 (2017), 545–590. https://doi.org/10.1016/j.jalgebra.2016.12.007

[8] Jin-Yi Cai, Martin Fürer, and Neil Immerman. 1992. An optimal lower bound on the number of variables for graph identifications. *Combinatorica* 12, 4 (1992), 389–410. https://doi.org/10.1007/BF01305232

[9] Peter J. Cameron. 1992. *Projective and polar spaces.* QMW Maths Notes, Vol. 13. Queen Mary and Westfield College, School of Mathematical Sciences, London. viii+147 pages.

[10] John J. Cannon and Derek F. Holt. 2003. Automorphism Group Computation and Isomorphism Testing in Finite Groups. *J. Symb. Comput.* 35, 3 (March 2003), 241âĂŞ267. https://doi.org/10.1016/S0747-7171(02)00133-5

[11] Bireswar Das and Shivdutt Sharma. 2019. Nearly Linear Time Isomorphism Algorithms for Some Nonabelian Group Classes. In *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019, Novosibirsk, Russia, July 1-5, 2019, Proceedings (Lecture Notes in Computer Science)*, René van Bevern and Gregory Kucherov (Eds.), Vol. 11532. Springer, 80–92. https://doi.org/10.1007/978-3-030-19955-5_8

[12] Heiko Dietrich and James B. Wilson. 2018. Polynomial-time isomorphism testing of groups of most finite orders. *CoRR* abs/1806.08872 (2018). arXiv:arXiv:1806.08872 arXiv.

[13] Bettina Eick, Charles R. Leedham-Green, and Eamonn A. O'Brien. 2002. Constructing automorphism groups of p-groups. *Comm. Algebra* 30, 5 (2002), 2271–2295. https://doi.org/10.1081/AGB-120003468

[14] François Le Gall. 2009. Efficient Isomorphism Testing for a Class of Group Extensions. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings (LIPIcs)*, Susanne Albers and Jean-Yves Marion (Eds.), Vol. 3. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 625–636. https://doi.org/10.4230/LIPIcs.STACS.2009.1830

[15] George Glauberman and Łukasz Grabowski. 2015. Groups with Identical k-Profiles. *Theory Comput.* 11, 15 (2015), 395–401. https://doi.org/10.4086/toc.2015.v011a015

[16] Walid Gomaa. 2010. Descriptive Complexity of Finite Abelian Groups. *Int. J. Algebra Comput.* 20, 8 (2010), 1087–1116. https://doi.org/10.1142/S0218196710006047

[17] Timothy Gowers. Blog entry: November 7, 2011, Comment: November 12, 2011. Comment on Dick Lipton's blog: The Group isomorphism Problem: A Possible Polymath Problem? (Blog entry: November 7, 2011, Comment: November 12, 2011). https://rjlipton.wordpress.com/2011/11/07/the-group-isomorphism-problem-a-possible-polymath-problem/ https://rjlipton.wordpress.com/2011/11/07/the-group-isomorphism-problem-a-possible-polymath-problem/.

[18] Joshua A. Grochow and Youming Qiao. 2017. Algorithms for Group Isomorphism via Group Extensions and Cohomology. *SIAM J. Comput.* 46, 4 (2017), 1153–1216. https://doi.org/10.1137/15M1009767

[19] Martin Grohe. 2000. Isomorphism testing for embeddable graphs through definability. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, STOC 2000, May 21-23, Portland, OR, USA*, F. Frances Yao and Eugene M. Luks (Eds.). ACM, 63–72. https://doi.org/10.1145/335305.335313

[20] Martin Grohe. 2017. *Descriptive complexity, canonisation, and definable graph structure theory.* Lecture Notes in Logic, Vol. 47. Association for Symbolic Logic, Ithaca, NY; Cambridge University Press, Cambridge.

[21] Lauri Hella. 1996. Logical Hierarchies in PTIME. *Inf. Comput.* 129, 1 (1996), 1–19. https://doi.org/10.1006/inco.1996.0070

[22] Neil Immerman and Eric Lander. 1990. Describing graphs: a first-order approach to graph canonization. In *Complexity theory retrospective.* Springer, New York, 59–81.

[23] Neil Immerman and Rik Sengupta. 2019. The k-Dimensional Weisfeiler-Leman Algorithm. *CoRR* abs/1907.09582 (2019). arXiv:arXiv:1907.09582 arXiv.

[24] Gábor Ivanyos and Youming Qiao. 2019. Algorithms Based on *-Algebras, and Their Applications to Isomorphism of Polynomials with One Secret, Group Isomorphism, and Polynomial Identity Testing. *SIAM J. Comput.* 48, 3 (2019), 926–963. https://doi.org/10.1137/18M1165682

[25] Sandra Kiefer, Ilia Ponomarenko, and Pascal Schweitzer. 2019. The Weisfeiler-Leman Dimension of Planar Graphs Is at Most 3. *J. ACM* 66, 6 (2019), 44:1–44:31. https://doi.org/10.1145/3333003

[26] Yinan Li and Youming Qiao. 2017. Linear Algebraic Analogues of the Graph Isomorphism Problem and the Erdős-Rényi Model. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, Chris Umans (Ed.). IEEE Computer Society, 463–474. https://doi.org/10.1109/FOCS.2017.49

[27] Eugene M. Luks. 2015. Group Isomorphism with Fixed Subnormal Chains. *CoRR* abs/1511.00151 (2015). arXiv:arXiv:1511.00151 http://arxiv.org/abs/1511.00151 arXiv.

[28] Alan H. Mekler. 1981. Stability of nilpotent groups of class 2 and prime exponent. *Journal of Symbolic Logic* 46, 4 (1981), 781âĂŞ788. https://doi.org/10.2307/2273227

[29] Gary L. Miller. 1978. On the $n^{\log n}$ Isomorphism Technique: A preliminary report. In *Proceedings of Tenth Annual ACM Symposium on Theory of Computing, STOC 1978* (San Diego, California, USA) *(STOC âĂŹ78)*. ACM, Association for Computing Machinery, New York, NY, USA, 51âĂŞ58. https://doi.org/10.1145/800133.804331

[30] André Nies and Katrin Tent. 2017. Describing finite groups by short first-order sentences. *Israel J. Math.* 221, 1 (2017), 85–115. https://doi.org/10.1007/s11856-017-1563-2

[31] Eamonn A. O'Brien. 1994. Isomorphism testing for p-groups. *J. Symbolic Comput.* 17, 2 (1994), 131, 133–147. https://doi.org/10.1006/jsco.1994.1007

[32] Martin Otto. 2017. *Bounded Variable Logics and Counting: A Study in Finite Models.* Lecture Notes in Logic, Vol. 9. Cambridge University Press. https://doi.org/10.1017/9781316716878

[33] Youming Qiao, Jayalal Sarma, and Bangsheng Tang. 2012. On Isomorphism Testing of Groups with Normal Hall Subgroups. *J. Comput. Sci. Technol.* 27, 4 (2012), 687–701. https://doi.org/10.1007/s11390-012-1255-7

[34] David J. Rosenbaum. 2013. Bidirectional Collision Detection and Faster Deterministic Isomorphism Testing. *CoRR* abs/1304.3935 (2013). arXiv:arXiv:1304.3935 http://arxiv.org/abs/1304.3935 arXiv.

[35] David J. Rosenbaum and Fabian Wagner. 2015. Beating the generator-enumeration bound for p-group isomorphism. *Theor. Comput. Sci.* 593 (2015), 16–25. https://doi.org/10.1016/j.tcs.2015.05.036

[36] Michael J. Smith. 1996. Computing automorphisms of finite soluble groups. *B. Aust. Math. Soc.* 53, 1 (1996), 169âĂŞ171. https://doi.org/10.1017/S0004972700016841

[37] Boris Weisfeiler. 1976. *On construction and identification of graphs.* Springer-Verlag, Berlin-New York. xiv+237 pages.

[38] Daniel Wiebking. 2020. Normalizes and permutational isomorphisms in simply-exponential time. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, Shuchi Chawla (Ed.). SIAM, 230–238. https://doi.org/10.1137/1.9781611975994.14

[39] James B. Wilson. 2008. *Group Decompositions, Jordan Algebras, and Algorithms for p-groups.* Ph.D. Dissertation. University of Oregon.

[40] James B. Wilson. 2016. The threshold for subgroup profiles to agree is $\Omega(\log n)$. *CoRR* abs/1612.01444 (2016). arXiv:1612.01444 https://arxiv.org/abs/1612.01444 arXiv.