Walk refinement, walk logic, and the iteration number of the Weisfeiler-Leman algorithm

Moritz Lichter
TU Kaiserslautern
lichter@cs.uni-kl.de

Ilia Ponomarenko
St. Petersburg Department of Steklov Mathematical Institute of the Russian Academy of Sciences
inp@pdmi.ras.ru

Pascal Schweitzer
TU Kaiserslautern
schweitzer@cs.uni-kl.de

***Abstract*—We show that the 2-dimensional Weisfeiler-Leman algorithm stabilizes n-vertex graphs after at most $\mathcal{O}(n \log n)$ iterations. This implies that if such graphs are distinguishable in 3-variable first order logic with counting, then they can also be distinguished in this logic by a formula of quantifier depth at most $\mathcal{O}(n \log n)$.**

For this we exploit a new refinement based on counting walks and argue that its iteration number differs from the classic Weisfeiler-Leman refinement by at most a logarithmic factor. We then prove matching linear upper and lower bounds on the number of iterations of the walk refinement. This is achieved with an algebraic approach by exploiting properties of semisimple matrix algebras. We also define a walk logic and a bijective walk pebble game that precisely correspond to the new walk refinement.

***Index Terms*—first order logic, counting quantifiers, quantifier depth, Weisfeiler-Leman algorithm**

I. Introduction

The classic Weisfeiler-Leman algorithm is a tool that lies at the heart of algebraic combinatorics. Developed first in 1968 to investigate symmetries of highly regular graphs, it is routinely employed for the purpose of isomorphism testing. Its development recently culminated in the WL2018 conference on "Symmetry vs Regularity" in algebraic graph theory in Pilsen [1], marking the 50 year anniversary of the algorithm. Roughly speaking, the core idea of the classic (2-dimensional) algorithm is to propagate structural information regarding pairs (u, w) of vertices in a graph by considering for each possible third vertex v the information already known for the pairs (u, v) and (v, w). To say equivalently, the algorithm repeatedly classifies pairs (u, w) of vertices according to the multiset of walks of length 2 from u to w. This process necessarily stabilizes after a finite number of iterations and the corresponding stabilization is used to classify pairs of vertices.

There is a close connection to a specific logic namely the 3-variable fragment of first order logic with counting [2]. Not only do the distinguishing power of the logic and the distinguishing power of the algorithm agree, there is also a close correspondence between the number of iterations required by the algorithm and the quantifier depth required in the logic.

In this paper we are therefore interested in the number of iterations after which the algorithm stabilizes. Equivalently, we are interested in the maximum quantifier depth needed in said first order logic to capture its expressibility on a graph of given size. There is a trivial upper bound of n^2 for this number, since there are only n^2 pairs of vertices and thus a proper chain of partitions on the vertex pairs (each partition finer than the previous one) cannot be longer than n^2. With regard to lower bounds, Fürer [3] proved that there are graphs on which the stabilization number is in $\Omega(n)$. In [4], using combinatorial techniques and a case distinction into small and large vertex-color classes, the currently best upper bound of $\mathcal{O}(n^2/\log n)$ for the iteration number was proven. In this paper we take an algebraic approach to the problem and show the following upper bound.

Theorem 1. *The 2-dimensional Weisfeiler-Leman algorithm stabilizes after $\mathcal{O}(n \log n)$ iterations on graphs with n vertices.*

Via the above-mentioned correspondence between the Weisfeiler-Leman (WL) algorithm and the logic [2] we obtain the following corollary.

Corollary 2. *If two n-vertex graphs can be distinguished by a sentence in 3-variable first order logic with counting $\mathcal{C}_3$, then there is also a $\mathcal{C}_3$ sentence of quantifier depth at most $\mathcal{O}(n \log n)$ that distinguishes the two graphs.*

To prove Theorem 1, we take an algebraic point of view and use a one to one correspondence between coherent configurations and coherent algebras [5]. The WL algorithm produces the former as output, whereas the latter are semisimple matrix algebras closed with respect to the Hadamard multiplication.

A. Our technique

Generalizing the idea of considering walks of length 2, we consider a new type of refinement, which we call the walk refinement. In one iteration it distinguishes pairs of vertices not only according to the multiset of 2-walks between them, but rather considers the multiset of all walks of arbitrary length between the two vertices. Naturally, considering all walks cannot be weaker than considering 2-walks. However, using arguments from linear algebra it can be proven that it suffices

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 (grant agreement No. 820148).

978-1-7281-3608-0/19/$31.00

to consider walks of bounded length. This in turn can be used to argue that the walk refinement is subsumed by a logarithmic number of traditional 2-walk refinements. In particular the two kinds of refinement yield the same stabilization and their iteration numbers differ by at most a logarithmic factor.

The cornerstone of our argument is then to show that the number of iterations of the walk refinement is at most linear in the number of vertices. This is done by observing that the result of walk refinement corresponds to a semisimple matrix algebra. Multiple iterations of walk refinement must therefore correspond to an increasing chain of semisimple subalgebras of a full matrix algebra. We can show that the length of such a chain is at most $\mathcal{O}(n)$, which gives a linear upper bound for the iteration number of repeated walk refinement.

Since our upper bound on the iteration number of the WL refinement is tight up to a logarithmic factor, the question arises whether the factor of $\log n$ can be removed. We show that walk refinement requires $\Theta(n)$ iterations on the same graphs, for which Fürer [3] showed that the WL refinement requires $\Theta(n)$ iterations. This leaves the problem open, whether our method can be pushed further.

In our paper, we also associate the walk refinement with a special Ehrenfeucht–Fraïssé type duplicator-spoiler game and a variant of a counting logic. We call them bijective walk pebble game and walk counting logic, respectively. They are suitable adaptations of a game and the logic $\mathcal{C}_3$ that are associated with the classic Weisfeiler-Leman (2-walk) refinement. The close correspondences between aspects of refinement algorithm, game, and logic translate to our scenario (Theorem 19). In particular, we prove tight bounds on the length of shortest winning strategies and optimal quantifier depth, respectively, of $\Theta(n)$ (Theorem 26, Corollaries 27 and 28).

We should remark that while the combinatorial techniques from [4] showing the upper bound of $\mathcal{O}(n^2/\log n)$ also translate to the setting without counting, our techniques seem to strongly rely on counting, since only the counting itself ensures the correspondence to matrix algebras that we exploit.

B. Related work

Deep Stabilization (see [6]), developed by Weisfeiler and Leman, is a generalization of the classic 2-dimensional WL algorithm. It can in turn be seen as a restricted form of the k-dimensional WL algorithm (for a suitable k) in the sense of Babai (see [2]). For each $k \in \mathbb{N}$ both generalizations give a polynomial-time algorithm that, as k increases, can distinguish more and more non-isomorphic graphs.

Over the course of the years striking connections have been drawn between the Weisfeiler-Leman algorithm and seemingly unrelated areas of research. While at first it was unclear whether the algorithm (for some k) solves the graph isomorphism in polynomial time, the seminal paper of Cai, Fürer, and Immerman [2] answered this question in the negative. Not only did they construct for each $k \in \mathbb{N}$ graphs that cannot be distinguished by the k-dimensional version, but they also exhibited a close connection to a logic with counting and described the Ehrenfeucht–Fraïssé type duplicator-spoiler game mentioned above. Optimal strategies in the game reflect precisely the outcome of the algorithm. The precise logic in question here is the k-variable fragment of first order logic with counting.

Babai employs the k-dimensional WL algorithm, with k logarithmic in the input, as a subroutine in his quasi-polynomial time algorithm for graph isomorphism testing [7].

In the language of Grohe [8], the Weisfeiler-Leman dimension of a graph G is the least number k for which the k-dimensional WL algorithm distinguishes G from every graph non-isomorphic to G. Equivalently, it is the number of variables needed in said fixed-point logic with counting to distinguish the graph from every non-isomorphic graph. Grohe shows [8] that graphs with a forbidden minor have bounded Weisfeiler-Leman dimension, reassuring the polynomial-time solvability of the isomorphism problem of such graphs [9]. For recent developments that relate techniques of the WL algorithm to group-CSP (constrained satisfaction problems in which the constraints are cosets of a group) we refer to [10].

Regarding bounds, Berkholz and Nordström [11] proved a lower bound on the number of iterations of the k-dimensional WL algorithm for finite structures. Specifically, they show for sufficiently large k the existence of n-element relational structures distinguished by the k-dimensional WL algorithm but for which $n^{o(k/\log k)}$ iterations do not suffice. For a different logic, namely the 3-variable existential negation-free fragment of first-order logic, Berkholz also developed techniques to prove tight bounds [12]. In contrast to these bounds, Fürer's lower bound [3] of $\Omega(n)$ mentioned above is applicable to graphs and in fact also applies to all fixed dimensions k.

II. Preliminaries

We denote with $[n]$ the set of numbers $\{1, \dots, n\}$ and with $\{\!\{x_1, \dots, x_k\}\!\}$ the multiset containing the elements x_1 to x_k. In this paper we work with colored directed graphs $G = (V, E, \chi)$, where $\chi\colon E \to C$ is a coloring function into some set C of colors. We will often consider complete directed graphs (with loops), i.e., the case $E = V^2$. We always require that χ assigns different colors to loops than it does to other edges (i.e., $\chi(v, v) \neq \chi(u, w)$ whenever $u \neq w$). For a tuple of $m > 1$ vertices $(v_1, \dots, v_m) \in V^m$ we set

$$\overline{\chi}(v_1, \dots, v_m) := (\chi(v_1, v_2), \chi(v_2, v_3), \dots, \chi(v_{m-1}, v_m))$$

and for a single vertex $v \in V$ we set $\chi(v) := \chi(v, v)$.

A coloring χ induces a partition $\pi(\chi)$ of the vertex pairs. For two colorings χ and χ' we write $\pi(\chi) \preceq \pi(\chi')$ to say that $\pi(\chi)$ is finer than $\pi(\chi')$. If not ambiguous we may only write $\chi \preceq \chi'$. If $\pi(\chi) = \pi(\chi')$ we also write $\chi \equiv \chi'$.

We say that χ respects *converse equivalence* if $\chi(u_1, v_1) = \chi(u_2, v_2)$ implies $\chi(v_1, u_1) = \chi(v_2, u_2)$ for all $u_1, u_2, v_1, v_2 \in V$, i.e., the color $\chi(u_1, v_1)$ determines $\chi(v_1, u_1)$.

A *k-walk* or walk of *length* k from v_1 to v_{k+1} is a tuple $(v_1, \dots, v_{k+1}) \in V^{k+1}$. Its color is $\overline{\chi}(v_1, \dots, v_{k+1})$. We say

that tuples in C^k are (potential) *k-walk colors* in χ and omit the coloring if it is clear from the context.

A *refinement* r is a function that for each graph G and coloring χ yields a new coloring χ' such that $\chi' \preceq \chi$. Additionally, r is required to be isomorphism invariant: if we apply r to two isomorphic, complete, and colored graphs (i.e., there is an isomorphism respecting the partitions induced by the colorings) then we obtain two new colorings that make the graphs isomorphic again.

We write χ_r for the application of the refinement r to the coloring χ and χ_r^m for m applications of r, i.e., $\chi_r^{m+1} = (\chi_r^m)_r$. We denote with χ_r^∞ the stable coloring, i.e., the χ_r^m for the smallest m such that $\chi_r^m \equiv \chi_r^{m+1}$. Let $G' = (V', E', \chi')$ be another colored complete graph, and suppose $u, v \in V$ and $u', v' \in V'$. We say that the refinement r *distinguishes* (u, v) from (u', v') in m iterations if $\chi_r^m(u, v) \neq (\chi')_r^m(u', v')$. The refinement r *distinguishes* G and G' in m iterations, if the multiset of colors after m iterations is different, that is

$$\{\!\{\chi_r^m(u, v) \mid u, v \in V\}\!\} \neq \{\!\{(\chi')_r^m(u', v') \mid u', v' \in V'\}\!\}.$$

We call the number of applications of r needed to obtain the stable partition the *iteration number* of r.

Now let $G = (V, E)$ be an undirected and uncolored graph. We turn G to a colored complete graph by defining a coloring $\chi\colon V^2 \to \{-1, 0, 1\}$ setting $\chi(v, v) = -1$ for all $v \in V$, as well as $\chi(v, u) = 1$ if $v \neq u$ and $(v, u) \in E(G)$, and setting $\chi(v, u) = 0$ otherwise. We refer to χ as the *initial coloring* of G. By construction the initial coloring respects converse equivalence. A refinement distinguishes two undirected and uncolored graphs if the respective initial colorings are distinguished by the refinement. The analogous definition applies to vertex pairs.

III. The Weisfeiler-Leman Refinement

In this section we recall the 2-dimensional WL refinement and its connections to the counting logic with 3 variables and the bijective 3-pebble game. A reader familiar with these notions should feel free to proceed to the next section.

Of particular interest in this paper is the 2-dimensional *Weisfeiler-Leman* refinement WL:

$$\chi_{\mathsf{WL}}(u, v) := \{\!\{\overline{\chi}(u, w, v) \mid w \in V\}\!\}.$$

Intuitively, WL refines the color of a vertex pair with the colors of all triangles containing this pair. This definition gives indeed a refinement: because loops and non-loops always have distinct colors, the presence of the color $\overline{\chi}(u, u, v)$ (or $\overline{\chi}(u, v, v)$) in the multiset ensures that pairs colored differently remain colored differently after applying WL. In particular, we do not need to include the color $\chi(u, v)$ of the previous iteration in the new color explicitly to ensure that WL is a refinement. In this paper, we refer with "the Weisfeiler-Leman refinement" always to the 2-dimensional WL refinement.

There is a close connection between the WL refinement, the counting logic with three variables $\mathcal{C}_3$, and the so called bijective 3-pebble game.

The logic $\mathcal{C}_3$ provides counting existential quantification and is limited to use three variables (but they may be bound multiple times). For $k \in \mathbb{N}$ in general, $\mathcal{C}_k$ formulas are defined for a variable set $\mathcal{V}$ of size k by the following grammar:

$$\varphi ::= x = y \mid x \sim y \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists^j x.\ \varphi \qquad x, y \in \mathcal{V}, j \in \mathbb{N}.$$

The variables will be interpreted as vertices of an undirected graph, $x = y$ expresses equality and $x \sim y$ the edge relation of the graph. A counting quantifier $\exists^j x.\ \varphi$ is satisfied if there are at least j distinct vertices that satisfy φ.

The bijective 3-pebble game is played by two players called Spoiler and Duplicator on two undirected graphs $G = (V, E)$ and $G' = (V', E')$. There are three pebble pairs $(p_1, q_1), (p_2, q_2), (p_3, q_3)$, where the pebbles of the i-th pair are labeled with number i. Initially, all pebbles are placed beside the graphs. During the game the p_i pebbles will be placed on the vertices of G and the q_i pebbles on the vertices of G'. A round of the game consists of the following moves:

1) Spoiler picks up a pair of pebbles (p_i, q_i).
2) Duplicator chooses a bijection $f\colon V \to V'$.
3) Spoiler places p_i on a vertex $v \in V$ and q_i on $f(v) \in V'$.

Spoiler wins the game after the m-th round, if mapping the vertex covered by pebble p_i to the vertex covered by pebble q_i is not an isomorphism of the subgraphs of G and G' induced by the vertices covered by pebbles. Duplicator wins the game if Spoiler never wins the game.

The connection between the 2-dimensional WL refinement, the logic $\mathcal{C}_3$, and the bijective 3-pebble game is the following: Let $(u, v) \in V^2$ and $(u', v') \in (V')^2$ be vertex pairs. Then the following statements are equivalent [2], [13]:

- If u, u', v, v' are covered by p_1, q_1, p_2, q_2, respectively, then Spoiler has a winning strategy finished after at most m rounds (i.e., a way to win in at most m rounds whatever Duplicator does).
- There is a $\mathcal{C}_3$ formula $\varphi(x, y)$ of quantifier depth m with exactly two free variables, such that φ holds on G when assigning u to x and v to y, but not on G' when assigning u' to x and v' to y.
- After m iterations of the Weisfeiler-Leman refinement (starting with the initial colorings of G and G') the pairs (u, v) and (u', v') are colored differently.

It follows that the WL refinement distinguishes exactly the same graphs as the logic $\mathcal{C}_3$ (there is a sentence holding on one graph but not the other) and as the bijective 3-pebble game (Spoiler has a winning strategy).

IV. Walk Refinement

We now introduce a new refinement. Suppose that $G = (V, E, \chi)$ is a complete and colored graph and recall that $\overline{\chi}(v_1, \ldots, v_m) = (\chi(v_1, v_2), \chi(v_2, v_3), \ldots, \chi(v_{m-1}, v_m))$. We define for $k \geq 2$ the *k-walk refinement* to be the function that for G and χ gives the new coloring $\chi_{\mathcal{W}[k]}$ defined by

$$\chi_{\mathcal{W}[k]}(u, v) := \{\!\{\overline{\chi}(u, w_1, \ldots, w_{k-1}, v) \mid w_i \in V\}\!\}.$$

Intuitively, the k-walk refinement refines the color of a vertex pair (u, v) with the color sequence of the traversed

vertex pairs along walks, taken over all possible walks of length k from u to v (taken as multiset). Note that, since χ assigns different colors to loops, the refinement implicitly also refines with respect to walks of shorter lengths $k' < k$ (indeed, the information is contained in the walks whose first $k - k'$ steps are of color $\chi(u)$, i.e., which are stationary at u). So the k-walk refinement is indeed a refinement, i.e. $\chi_{\mathcal{W}[k]} \preceq \chi$, because walks of length 1 are just the old colors. It is easy to see that the k-walk refinement is isomorphism invariant and preserves converse equivalence.

From what we just argued, obviously $\chi_{\mathcal{W}[k]} \preceq \chi_{\mathcal{W}[j]}$ if $k \geq j$. Also note that 2-walk refinement is exactly the 2-dimensional Weisfeiler-Leman refinement. Thus, for $k \geq 2$

$$\chi_{\mathcal{W}[k]} \preceq \chi_{\mathsf{WL}} \preceq \chi.$$

We argue next that k-walk refinement can be simulated with a logarithmic number of Weisfeiler-Leman refinements.

Lemma 3. *If $k \geq 2$ then $\chi_{\mathsf{WL}}^{\lceil \log k \rceil} \preceq \chi_{\mathcal{W}[k]}$.*

Proof: Let C be the set of colors of $\chi\colon V^2 \to C$ and let C_i be the set of colors of $\chi_{\mathsf{WL}}^i\colon V^2 \to C_i$.

We show by induction that after $i \geq 1$ iterations of the Weisfeiler-Leman refinement for each color $d \in C_i$ there is a function $\bar{d}\colon C^{(2^i)} \to \mathbb{N}$ with the following property: For all $u, v \in V$ of color $\chi_{\mathsf{WL}}^i(u,v) = d$ and for all 2^i-walk colors $(c_1, \dots c_{2^i}) \in C^{(2^i)}$ in χ there are exactly $\bar{d}(c_1, \dots, c_{2^i})$ many $(c_1, \dots, c_{2^i})$ colored walks between u and v in χ. In particular, the number of $(c_1, \dots, c_{2^i})$ colored walks is the same for all such u and v. This implies $\chi_{\mathsf{WL}}^i \preceq \chi_{\mathcal{W}[2^i]}$.

For $i = 1$, the Weisfeiler-Leman refinement assigns colors such that every color just contains the possible 2-walk colors. So assume $i > 1$.

Let $u, v \in V$ be vertices, $d_1, e_1, \dots, d_n, e_n \in C_i$ be colors of χ_{WL}^i, $d = \{\!\{(d_1, e_1), \dots, (d_n, e_n)\}\!\} \in C_{i+1}$ be a color of χ_{WL}^{i+1}, and $(c_1, \dots, c_{2^{i+1}}) \in C^{(2^{i+1})}$ be a $2^{(i+1)}$-walk color. We set

$$\bar{d}(c_1, \dots, c_{2^{i+1}}) := \sum_{j \in [n]} \bar{d}_j(c_1, \dots, c_{2^i}) \cdot \bar{e}_j(c_{2^i+1}, \dots, c_{2^{i+1}}).$$

By induction hypothesis $\bar{d}_i$ and $\bar{e}_i$ yield the correct number of 2^i-walks and so $\bar{d}$ yields the correct number of 2^{i+1} walks. ∎

This lemma corresponds to the known fact that in $\mathcal{C}_3$ walks of length k can be defined by a formula of quantifier depth $\lceil \log k \rceil$. These walks can be counted using counting quantifiers similarly. Considering paths we see that the k-walk refinement cannot be simulated with less than a logarithmic number of Weisfeiler-Leman refinements, and in that sense the bound in the lemma is tight. On the other hand the relation can be strict, that is $\chi_{\mathsf{WL}}^{\lceil \log k \rceil} \prec \chi_{\mathcal{W}[k]}$. However, the Weisfeiler-Leman and k-walk refinement produce the same stable partition because finitely many steps of one subsume a single step of the other.

Lemma 4. *If $k \geq 2$, then $\chi_{\mathsf{WL}}^\infty \equiv \chi_{\mathcal{W}[k]}^\infty$.*

Proof: Suppose that $\chi_{\mathcal{W}[k]}^\infty \equiv \chi_{\mathcal{W}[k]}^j$ and $\chi_{\mathsf{WL}}^\infty \equiv \chi_{\mathsf{WL}}^j$ for some suitable j. Then $\chi_{\mathcal{W}[k]}^j \preceq \chi_{\mathsf{WL}}^j \equiv \chi_{\mathsf{WL}}^{j \cdot \lceil \log k \rceil} \preceq \chi_{\mathcal{W}[k]}^j$ by Lemma 3, and hence $\chi_{\mathsf{WL}}^\infty \equiv \chi_{\mathcal{W}[k]}^\infty$. ∎

We remark that it is possible that the partitions produced by the Weisfeiler-Leman refinement and the partitions produced by k-walk refinement all disagree except for the stable partitions in the end (for example this is the case for the graphs $X(G_n^2)$ for $2 \leq n \leq 10$ defined in Section VIII as shown by computer calculations with $k = n$).

We define the *walk refinement* $\chi_{\mathcal{W}}$ as the finest k-walk refinement. More precisely, we define it as $\chi_{\mathcal{W}[k]}$ for the smallest k, for which $\chi_{\mathcal{W}[k]}$ induces the finest partition over all choices of k. We will prove in Section V that n^2-walk refinement always produces this finest partition, thus $k \leq n^2$. From that we will conclude that $\chi_{\mathcal{W}} \equiv \chi_{\mathcal{W}[n^2]}$ and $\chi_{\mathsf{WL}}^{\mathcal{O}(\log n)} \preceq \chi_{\mathcal{W}}$. This will allow us to bound the iteration number of the Weisfeiler-Leman refinement by bounding the iteration number of walk refinement.

V. Iteration Number of Walk Refinement

In this section we show that walk refinement stabilizes after $\mathcal{O}(n)$ iterations. We interpret the partitions produced by walk refinement as matrix algebras. If walk refinement strictly refines the partition then the algebra is strictly enlarged. We obtain the linear bound by observing that these algebras can be nested at most a linear number of times.

Throughout this section, let $G = (V, E, \chi)$ be a complete and colored graph with $V = [n]$ and let $\chi\colon E \to C$ respect converse equivalence.

A. Background on Matrix Algebras

In this section we make use of standard material from representation theory, see e.g. [14]. Let S be a set of $n \times n$ matrices over $\mathbb{C}$. We denote with $\mathbb{C}S$ the $\mathbb{C}$-linear span of S and with

$$S^{\leq k} := \{M_1 \cdot \ldots \cdot M_j \mid j \leq k, M_i \in S \text{ for all } i \in [j]\}$$

the set of all products of matrices in S with at most k factors. Clearly $S^{\leq k} \subseteq S^{\leq k+1}$. We write $\widehat{S}$ for the union of all $S^{\leq k}$.

For a color $c \in C$ we denote with M_c the $n \times n$ *color c adjacency matrix*, that is $(M_c)_{ij} = 1$ if $\chi(i,j) = c$ and $(M_c)_{ij} = 0$ otherwise. The set of all color adjacency matrices is denoted by M_χ. The coloring χ thereby induces an $n \times n$ matrix algebra over the complex numbers:

$$\langle \chi \rangle := \mathbb{C}\widehat{M_\chi}.$$

The algebra $\langle \chi \rangle$ is closed under (conjugate) transposition because χ respects converse equivalence.

We write $\mathsf{M}_k(\mathbb{C})$ for the (full) matrix algebra of all $k \times k$ matrices over the complex numbers. It is a well-known fact that a matrix algebra $\mathcal{A} \subseteq \mathsf{M}_n(\mathbb{C})$ closed under conjugate transposition is always semisimple. Indeed, if M is in the Jacobson radical of $\mathcal{A}$, so is M^*M. But M^*M is diagonalizable (because it is Hermitian) and nilpotent (because the radical is nilpotent, Lemma 1.6.6 in [14]) and hence $M^*M = 0$ and so $M = 0$. Then the radical itself is 0, which is one characterization of semisimplicity.

By the theorem of Wedderburn (Corollary 1.4.17 in [14]) a semisimple matrix algebra $\mathcal{A} \subseteq \mathsf{M}_n(\mathbb{C})$ is always isomorphic to a direct sum of full matrix algebras, that is

$$\mathcal{A} \cong \mathsf{M}_{a_1}(\mathbb{C}) \oplus \cdots \oplus \mathsf{M}_{a_k}(\mathbb{C}),$$

for some positive integers k and $a_1, \ldots, a_k$. The direct sum decomposition is unique up to reordering. We will prove a bound on the length of proper chains $\mathcal{A}_1 \subsetneq \cdots \subsetneq \mathcal{A}_m \subseteq \mathsf{M}_n(\mathbb{C})$ of semisimple matrix algebras. This is the essential theorem to bound the iteration number of walk refinement:

Theorem 5. *Let $\mathcal{A}_1 \subsetneq \cdots \subsetneq \mathcal{A}_m \subseteq \mathsf{M}_n(\mathbb{C})$ be a chain of semisimple strict subalgebras. Then $m \leq 2n$.*

To prove the theorem we need several auxiliary lemmas, which may be self-evident for a reader familiar with the theory of semisimple algebras. They show that such chains behave well with respect to the direct sum decompositions of the $\mathcal{A}_i$.

Lemma 6. *If there is an algebra monomorphism*

$$\mathsf{M}_{a_1}(\mathbb{C}) \oplus \cdots \oplus \mathsf{M}_{a_k}(\mathbb{C}) \to \mathsf{M}_m(\mathbb{C}),$$

then $\sum_{i=1}^{k} a_i \leq m$.

Proof: For each i, there are exactly a_i diagonal matrices in $\mathsf{M}_{a_i}(\mathbb{C})$ with one entry 1 and all other 0. These matrices are nonzero, idempotent, and pairwise orthogonal (i.e., the product of any two of them is 0). It follows that the direct sum, and hence the monomorphism image, contains a set of $d = \sum_{i=1}^{k} a_i$ nonzero, idempotent, and pairwise orthogonal elements.

Let $M_1, \ldots, M_d \in \mathsf{M}_m(\mathbb{C})$ be the matrices of this set. Since the M_i are nonzero, each has rank at least 1. Suppose $i \neq j \in [d]$. Then $M_i + M_j$ is idempotent, orthogonal to all other M_ℓ, and has $\operatorname{rank}(M_i + M_j) = \operatorname{rank}(M_i) + \operatorname{rank}(M_j)$ (see e.g. Theorem IV.12 in [15]). Because the maximal rank of an $m \times m$ matrix is m, it follows by induction that $d = \sum_{i=1}^{k} a_i \leq \operatorname{rank}(\sum_{i=1}^{k} M_i) \leq m$. ∎

Lemma 7. *Suppose $a_1, \ldots, a_k, b_1, \ldots, b_\ell \in \mathbb{N}$ and let*

$$\varphi \colon \bigoplus_{i=1}^{k} \mathsf{M}_{a_i}(\mathbb{C}) \to \bigoplus_{j=1}^{\ell} \mathsf{M}_{b_j}(\mathbb{C})$$

be an algebra monomorphism. Then for every $i \in [k]$ there is a $j \in [\ell]$ such that $\pi_j \circ \varphi$ maps $\mathsf{M}_{a_i}(\mathbb{C})$ injectively into $\mathsf{M}_{b_j}(\mathbb{C})$, where π_j is the projection onto the j-th component $\mathsf{M}_{b_j}(\mathbb{C})$.

Proof: Full matrix algebras are simple, i.e., contain no proper nontrivial two-sided ideals. For a simple algebra $\mathcal{A}$, any homomorphism $\psi \colon \mathcal{A} \to \mathcal{B}$ into some algebra $\mathcal{B}$ is either injective or zero (otherwise, $\ker(\psi)$ is a proper nontrivial two-sided ideal of $\mathcal{A}$). Suppose $i \in [k]$. The map $\pi_j \circ \varphi_i \colon \mathsf{M}_{a_i}(\mathbb{C}) \to \mathsf{M}_{b_j}(\mathbb{C})$ is an algebra monomorphism for all $j \in [\ell]$, where φ_i is the restriction of φ to the i-th component $\mathsf{M}_{a_i}(\mathbb{C})$. Now, $\pi_j \circ \varphi_i$ must be injective for some j, because if all $\pi_j \circ \varphi_i$ were zero, φ was not injective. ∎

Lemma 8. *Let $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathsf{M}_n(\mathbb{C})$ be two semisimple matrix algebras with direct sum decompositions*

$$\mathcal{A} \cong \mathsf{M}_{a_1}(\mathbb{C}) \oplus \cdots \oplus \mathsf{M}_{a_k}(\mathbb{C}) \text{ and}$$
$$\mathcal{B} \cong \mathsf{M}_{b_1}(\mathbb{C}) \oplus \cdots \oplus \mathsf{M}_{b_\ell}(\mathbb{C}).$$

Then $2(\sum_{i=1}^{k} a_i) - k \leq 2(\sum_{j=1}^{\ell} b_j) - \ell$ with equality exactly if $\mathcal{A} = \mathcal{B}$.

Proof: First, we pick an algebra monomorphism $\varphi \colon \bigoplus_{i=1}^{k} \mathsf{M}_{a_i}(\mathbb{C}) \to \bigoplus_{j=1}^{\ell} \mathsf{M}_{b_j}(\mathbb{C})$. Second, for each $i \in [k]$ we choose an $f(i) = j$ such that $\pi_j \circ \varphi$ maps $\mathsf{M}_{a_i}(\mathbb{C})$ injectively into $\mathsf{M}_{b_j}(\mathbb{C})$. Such choices exist by Lemma 7. For each $j \in [\ell]$ we obtain a monomorphism $\bigoplus_{i \in f^{-1}(j)} \mathsf{M}_{a_i}(\mathbb{C}) \to \mathsf{M}_{b_j}(\mathbb{C})$ by restricting $\pi_j \circ \varphi$. From Lemma 6 it now follows that $b_j \geq \sum_{i \in f^{-1}(j)} a_i$. Then, to show that $2(\sum_{i=1}^{k} a_i) - k \leq 2(\sum_{j=1}^{\ell} b_j) - \ell$, simply observe that for each $j \in [\ell]$ we have $2b_j - 1 \geq 2(\sum_{i \in f^{-1}(j)} a_i) - |f^{-1}(j)|$ (since $b_j > 0$) and that summing up over all j yields the desired equation. Finally, consider the case of equality and let $j \in [\ell]$. Then $2b_j - 1 = 2(\sum_{i \in f^{-1}(j)} a_i) - |f^{-1}(j)|$ and because $b_j \geq \sum_{i \in f^{-1}(j)} a_i$ it follows that $|f^{-1}(j)| = 1$ and $b_j = a_{f^{-1}(j)}$. Thus f is a bijection satisfying $a_i = b_{f(i)}$ for all $i \in [k]$ and φ is an isomorphism. ∎

We now conclude the proof of Theorem 5.

Proof of Theorem 5: For all $i \in [m]$ suppose that $\mathcal{A}_i \cong \bigoplus_{j=1}^{n_i} \mathsf{M}_{a_{ij}}(\mathbb{C})$ and define $s_i := 2(\sum_{j=1}^{n_i} a_{ij}) - n_i$. Then $s_m \leq 2n - 1$ by Lemma 6. Moreover, by Lemma 8 for all $i \in [m-1]$ we have $s_i \leq s_{i+1}$ and in fact even $s_i < s_{i+1}$ since equality would imply $\mathcal{A}_i = \mathcal{A}_{i+1}$. Thus $m \leq 2n$. ∎

B. Matrix Algebras and Walk Refinement

We say that a matrix $M \in \mathsf{M}_n(\mathbb{C})$ distinguishes (u_1, v_1) from (u_2, v_2) if $M_{u_1,v_1} \neq M_{u_2,v_2}$ and that a set $S \subseteq \mathsf{M}_n(\mathbb{C})$ distinguishes (u_1, v_1) from (u_2, v_2) if S contains a matrix distinguishing them.

We now show that with one iteration of walk refinement we can distinguish the same vertex pairs as with the induced algebra $\langle \chi \rangle$.

Let $c_1, c_2 \in C$ be colors. Then $(M_{c_1} \cdot M_{c_2})_{u,v}$ is the number of (c_1, c_2) colored walks from u to v. In general, let $c_1, \ldots, c_k$ be colors. Then $(M_{c_1} \cdot \ldots \cdot M_{c_k})_{u,v}$ is the number of $(c_1, \ldots, c_k)$ colored walks from u to v. Because walk refinement and the induced algebra essentially count colored walks, they distinguish the same vertex pairs:

Lemma 9. *Let $u_1, v_1, u_2, v_2 \in V$. The walk refinement $\chi_{\mathcal{W}}$ distinguishes (u_1, v_1) from (u_2, v_2) if and only if the induced algebra $\langle \chi \rangle$ distinguishes them.*

Proof: On the one hand suppose that walk refinement distinguishes the vertices, i.e. $\chi_{\mathcal{W}}(u_1, v_1) \neq \chi_{\mathcal{W}}(u_2, v_2)$. Then there is a sequence of colors $c_1, \ldots, c_k$ such that the number of $(c_1, \ldots, c_k)$ colored walks between u_1 and v_1 is different from the number of such walks between u_2 and v_2. Hence $M_{c_1} \cdot \ldots \cdot M_{c_k}$ distinguishes the two vertex pairs.

On the other hand let $M \in \langle \chi \rangle$ distinguish (u_1, v_1) and (u_2, v_2). The matrix M is a linear combination of products of color adjacency matrices:

$$M = \sum_{i=1}^{m} z_i \prod_{j=1}^{k_i} M_{c_{ij}}$$

where $z_i \in \mathbb{C}$ and $c_{ij} \in C$ for all $i \in [m]$ and $j \in [k_i]$. There must be an i such that $\prod_{j=1}^{k_i} M_{c_{ij}}$ distinguishes (u_1, v_1) and (u_2, v_2), because M distinguishes them. Hence the number of $(c_{i1}, \ldots, c_{ik_i})$ colored walks between u_1 and v_1 is different from the number of such walks between u_2 and v_2 and the pairs are distinguished by walk refinement. ∎

Corollary 10. *Either* $\langle \chi \rangle \subsetneq \langle \chi_{\mathcal{W}} \rangle$ *or* $\chi \equiv \chi_{\mathcal{W}}$.

The induced algebra gets strictly larger if the partition induced by walk refinement gets strictly finer. We obtain the bound on the walk refinement iterations, because the algebras can be nested only $2n$ many times.

Theorem 11. *Walk refinement stabilizes in* $2n$ *iterations.*

Proof: Assume that m is the smallest number such that $\chi_{\mathcal{W}}^{m} \equiv \chi_{\mathcal{W}}^{\infty}$. Because walk refinement preserves converse equivalence, the algebras $\langle \chi \rangle, \langle \chi_{\mathcal{W}} \rangle, \ldots, \langle \chi_{\mathcal{W}}^{m} \rangle$ are semisimple. Then from Corollary 10 it follows that $\langle \chi \rangle \subsetneq \langle \chi_{\mathcal{W}} \rangle \subsetneq \cdots \subsetneq \langle \chi_{\mathcal{W}}^{m} \rangle$ and from Theorem 5 that $m \leq 2n$. ∎

To obtain a bound on the iteration number of the Weisfeiler-Leman refinement, it remains to relate the Weisfeiler-Leman refinement and the walk refinement.

Lemma 12. $\chi_{\mathcal{W}} \equiv \chi_{\mathcal{W}[n^2]}$ *and* $\chi_{\mathsf{WL}}^{\mathcal{O}(\log n)} \preceq \chi_{\mathcal{W}}$.

Proof: We first show $\chi_{\mathcal{W}} \equiv \chi_{\mathcal{W}[n^2]}$. A close inspection of the proof of Lemma 9 shows that $\mathbb{C}M_{\chi}^{\leq k}$ distinguishes the same vertex pairs as k-walk refinement. It suffices to show that $\mathbb{C}\widehat{M}_{\chi} = \mathbb{C}M_{\chi}^{\leq n^2}$, which implies $\langle \chi \rangle = \mathbb{C}M_{\chi}^{\leq n^2}$ and $\chi_{\mathcal{W}} \equiv \chi_{\mathcal{W}[n^2]}$.

The argument is well-known: Let S be a set of $n \times n$ matrices, then clearly $\dim \mathbb{C}S^{\leq k} \leq \dim \mathbb{C}S^{\leq k+1}$. If $\dim \mathbb{C}S^{\leq k} = \dim \mathbb{C}S^{\leq k+1}$, then $\mathbb{C}S^{\leq k} = \mathbb{C}S^{\leq j}$ for all $j \geq k$. Hence $\mathbb{C}\widehat{S} = \mathbb{C}S^{\leq n^2}$ because the dimension can be at most n^2.

Now $\chi_{\mathsf{WL}}^{\mathcal{O}(\log n)} \preceq \chi_{\mathcal{W}}$ follows by Lemma 3. ∎

Proof of Theorem 1: Finally, combining Theorem 11 with Lemmas 4 and 12 proves Theorem 1, namely that the Weisfeiler-Leman refinement stabilizes in $\mathcal{O}(n \log n)$ iterations. ∎

We argued that the length of the involved matrix algebras (the smallest number k, such that $\mathbb{C}S^{\leq k} = \mathbb{C}\widehat{S}$) is at most n^2. We remark that there is even an $\mathcal{O}(n \log n)$ bound [16] for the length of matrix algebras. But this bound does not improve our bound on Weisfeiler-Leman iterations asymptotically.

VI. Walk Counting Logic

The Weisfeiler-Leman refinement can distinguish the same graphs as the counting logic $\mathcal{C}_3$. More strongly the number of Weisfeiler-Leman iterations needed to distinguish two vertex pairs equals the minimum quantifier depth of a formula to distinguish them. As we have already seen, for $k \geq 2$ the k-walk refinement distinguishes the same vertex pairs, too. But the required iterations of walk refinement do not correspond to the quantifier depth of $\mathcal{C}_3$. We now introduce a logic $\mathcal{W}_k$ we call *k-walk counting logic* for which such a correspondence holds. The logic is defined for undirected and uncolored graphs. We could relax the restriction to directed and colored graphs respecting converse equivalence as in the previous section but this is not needed in this paper.

The logic $\mathcal{W}_k$ uses a set $\mathcal{V}$ of $k+1$ variables and every $\mathcal{W}_k$ formula φ has at most two free variables, which we indicate using the notation $\varphi(x, y)$. The $\mathcal{W}_k$ formulas with free variables $z_1, z_{k+1} \in \mathcal{V}$ are defined according to the grammar

$$\begin{aligned}\varphi(z_1, z_{k+1}) ::= {} & z_1 = z_{k+1} \mid z_1 \sim z_{k+1} \mid \\ & \varphi(z_1, z_{k+1}) \wedge \varphi(z_1, z_{k+1}) \mid \neg\varphi(z_1, z_{k+1}) \mid \\ & \exists^{j}(z_2, \ldots, z_k).\ \bigwedge_{i \in [k]} \varphi(z_i, z_{i+1})\end{aligned}$$

where $z_i \in \mathcal{V}$ and $j \in \mathbb{N}$. The variables $z_1, \ldots, z_{k+1} \in \mathcal{V}$ in the existential quantifier are required to be pairwise distinct. We call the existential quantifier above a *k-walk quantifier*. As usual with grammars, the subformulas of a k-walk quantifier $\varphi(z_i, z_{i+1})$, which are non-terminals, can be replaced by different formulas. The k-walk quantifier above is satisfied, if there are at least j distinct tuples $(v_2, \ldots, v_k)$ of vertices satisfying the rest of the formula.

Note that sentences can for example be obtained by setting $\varphi(z_1, z_2)$ to be the formula $z_2 = z_2$ and setting $\varphi(z_k, z_{k+1})$ to be $z_k = z_k$. This restricts the top most walk quantifier to quantify over walks of length $k-2$. The restriction could be relaxed, but that would complicate the definition and is not needed for our purpose.

Syntactically, $\mathcal{W}_k$ is not a subset of $\mathcal{W}_{k+1}$, but obviously for every $\mathcal{W}_k$ formula there is an equivalent $\mathcal{W}_{k+1}$ formula.

Let $\varphi(x, y)$ be a $\mathcal{W}_k$ formula, $G = (V, E)$ an undirected graph, and $u, v \in V$ be vertices. By $\varphi_G(u, v)$ we denote the truth value of φ on G when assigning u to x and v to y. We omit the subscript if the graph is clear from the context.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two undirected graphs, $u_1, v_1 \in V_1$, and $u_2, v_2 \in V_2$. We say that

- a $\mathcal{W}_k$ formula $\varphi(x, y)$ *distinguishes* (u_1, v_1) from (u_2, v_2), if $\varphi_{G_1}(u_1, v_1)$ is different from $\varphi_{G_2}(u_2, v_2)$,
- a $\mathcal{W}_k$ sentence φ *distinguishes* G_1 from G_2 if φ has different truth values on G_1 and G_2, and
- $\mathcal{W}_k$ distinguishes G_1 from G_2 if there is a $\mathcal{W}_k$ sentence distinguishing them.

For a coloring $\chi \colon V^2 \to C$, we also say that a $\mathcal{W}_k$ formula $\varphi(x, y)$ *identifies* a color $c \in C$ in χ, if $\varphi(u, v)$ holds if and only if $\chi(u, v) = c$ for all $u, v \in V$.

We call the union of the $\mathcal{W}_k$ logics for all $k \in \mathbb{N}$ *walk counting logic*. Consequently, the number of variables in the walk counting logic is unbounded. The definitions for

distinguishing vertex pairs and graphs for walk counting logic are analogous to $\mathcal{W}_k$.

We now show that with $\mathcal{W}_k$ formulas of quantifier depth m one can distinguish at least as many vertex pairs as with m iterations of k-walk refinement.

Lemma 13. *Let $G = (V, E)$ be an undirected graph, χ the initial coloring for G, and c a color produced by m iterations of k-walk refinement. Then there is a $\mathcal{W}_k$ formula $\varphi(x, y)$ of quantifier depth m identifying c in $\chi^m_{\mathcal{W}[k]}$. Moreover, $\varphi(x, y)$ only depends on n and c (but not on G).*

Proof: If $m = 0$, then c stands either for loop, edge, or non edge, which is identified by the formulas $x = y$, $x \sim y$, and $x \not\sim y$.

Let c be a color in the $(m + 1)$-th iteration. Hence c is a multiset of k-walk colors in $\chi^m_{\mathcal{W}[k]}$. Let $(c_1, \ldots, c_k)$ occur with multiplicity j in c. Then there are formulas φ_{c_i} of quantifier depth m identifying c_i in $\chi^m_{\mathcal{W}[k]}$ by induction hypothesis. The formula

$$\varphi_{(c_1,\ldots,c_k)}(z_1, z_{k+1}) := \exists^j z_2, \ldots, z_k. \bigwedge_{i\in[k]} \varphi_{c_i}(z_i, z_{i+1})$$

holds for vertices u and v assigned to z_1 and z_{k+1} respectively if and only if there are at least j many $(c_1, \ldots, c_k)$ colored walks from u to v in $\chi^m_{\mathcal{W}[k]}$.

Then the conjunction over all $(c_1, \ldots, c_k)$ in c identifies c in $\chi^{m+1}_{\mathcal{W}[k]}$ and is of quantifier depth $m + 1$. ∎

There is a technical detail that, when one is interested in distinguishing graphs rather than distinguishing vertex pairs, one (sometimes) needs an additional quantifier. This is the case because a refinement distinguishes two graphs after m applications, if the multisets of colors of both graphs are different. Hence, there is a hidden quantifier (saying that there is a color, that occurs with different multiplicity in both graphs).

Lemma 14. *If m iterations of k-walk refinement distinguish two graphs G_1 and G_2, then a $\mathcal{W}_k$ sentence of quantifier depth $m+1$ (respectively $m+2$ if $k = 2$) distinguishes G_1 and G_2.*

Proof: Assume m iterations of k-walk refinement distinguish the two graphs and let χ_i be the coloring obtained for G_i for $i \in [2]$. Then there is a color c occurring for a different number of vertex pairs, say n_1 and n_2 with $n_1 > n_2$, in χ_1 and χ_2, respectively. Let $\varphi(x, y)$ be the formula from Lemma 13 of quantifier depth m that identifies vertex pairs of color c in both colorings. Now the formula $\exists^{n_1} x, y.\ \varphi(x, y)$ is of quantifier depth $m + 1$ and distinguishes the graphs.

Suppose now that $k = 2$ (and hence the prior formula is not a valid $\mathcal{W}_2$ formula). Let D be the multiset of c-outdegrees of all vertices in G_1. Then the sum of all c-outdegrees (respecting the multiplicity) is n_1. Let $\mathcal{D}$ be the set of all possible c-outdegree multisets with sum n_1. Then the formula

$$\bigvee_{D\in\mathcal{D}} \bigwedge_{(i,d)\in D} \exists^i x.\ \exists^d y.\ \varphi(x, y)$$

distinguishes G_1 and G_2, where $(i, d) \in D$ says that d occurs with multiplicity i in D. ∎

VII. Bijective Walk Pebble Game

We now describe a game called the *bijective k-walk pebble game*, which corresponds to k-walk refinement and k-walk counting logic. It is an adaption of the bijective 3-pebble game to agree with the k-walk refinement.

There are two players, Spoiler and Duplicator. The game is played on two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Spoiler obtains $k + 1$ pairs of pebbles $(p_1, q_1), \ldots, (p_{k+1}, q_{k+1})$ labeled with numbers 1 to $k+1$. We say that the pebble pairs (p_i, q_i) and (p_{i+1}, q_{i+1}) for all $i \in [k]$ and the pairs (p_{k+1}, q_{k+1}) and (p_1, q_1) are *consecutive*. Given a pebble pair (p_i, q_i) we will for simplicity write (p_{i+1}, q_{i+1}) for the next consecutive pebble pair, in particular in the case $i = k + 1$, where (p_1, q_1) is meant.

If $|V_1| \neq |V_2|$, Spoiler wins immediately. Otherwise, all pebbles are placed beside the graphs. The game is played in multiple rounds. One round consists of the following three moves:

1) If there are pebbles already placed on the graphs, Spoiler can choose a pair of pebbles (p_i, q_i), replace it with (p_1, q_1) and then must also replace the next pair (p_{i+1}, q_{i+1}), if it is placed on the graph, with (p_{k+1}, q_{k+1}). In either case, she (Spoiler) then picks up all pebble pairs apart the first and last.
2) Duplicator chooses a bijection $f\colon V_1^{k-1} \to V_2^{k-1}$.
3) Spoiler places the pebbles p_i for $2 \leq i \leq k$ onto vertices of G_1. She may place multiple pebbles on the same vertex. Assume pebble p_i is placed onto vertex u_i and $f(u_2, \ldots, u_k) = (v_2, \ldots, v_k)$. Then, Spoiler also places the pebbles q_i onto v_i for all $2 \leq i \leq k$.

Thus, as opposed to a bijection between vertices in the classic game, Duplicator chooses a bijection from the k-walks in G_1 from p_1 to p_{k+1} to the k-walks from q_1 to q_{k+1} in G_2 (hence the name walk pebble game).

We say that Spoiler wins the game after the i-th round, if there are consecutive pebble pairs (p_i, q_i) and (p_{j+1}, q_{j+1}) placed on vertices $(u, v) \in V_1^2$ and $(u', v') \in V_2^2$, such that the induced subgraphs $G_1[\{u, v\}]$ and $G_2[\{u', v'\}]$ are not isomorphic. Duplicator wins the game if Spoiler never wins the game.

We say that Spoiler can *force a win after the i-th round* or has *a winning strategy in i rounds*, if she can always win the game after the i-th round for all possibles moves of Duplicator.

With *bijective walk pebble game* we refer to the game in which Spoiler is allowed to choose the number k in the beginning (after she has seen the two graphs).

Note that, similar to the toplevel quantifier of a $\mathcal{W}_k$ sentence, in the first round, only $k - 1$ pebbles are placed on the graph and hence they describe a $(k-2)$-walk. Besides keeping definitions simpler, in the case $k = 2$ this also ensures that the game becomes the bijective 3-pebble game (modulo some irrelevant replacements of pebble pairs).

In the following, let G_1 and G_2 be two undirected graphs. Furthermore let $u, v \in V_1$, and $u', v' \in V_2$. We say that the bijective k-walk pebble game *distinguishes* (u, v) from (u', v') in m rounds, if Spoiler has a winning strategy in m rounds in the game that has been altered as follows: instead of her making her first move in the first round, the pebble pairs (p_1, q_1) and (p_{k+1}, q_{k+1}) are placed on (u, u') and (v, v'), respectively. Afterwards the game proceeds normally with Duplicator choosing a bijection and so on. In the special case that (u, v) is an edge, non-edge, or a loop but (u', v') is not of the same type, we say that the vertex pairs are distinguished in 0 rounds.

Lemma 15. *Let $u, v \in V_1$ and $u', v' \in V_2$. If there is a $\mathcal{W}_k$ formula $\varphi(x, y)$ of quantifier depth m that distinguishes (u, v) from (u', v'), then so does the bijective k-walk pebble game in m rounds.*

Proof: Assume that $\varphi(x, y)$ distinguishes (u, v) from (u', v') and that the pebble pairs (p_1, q_1) and (p_{k+1}, q_{k+1}) are placed on these vertices. The proof proceeds by induction on m.

If $m = 0$, $\varphi(x, y)$ is quantifier free, hence p_1 and p_{k+1} cover an edge, non-edge, or the same vertex, where q_1 and q_{k+1} cover something different, and Spoiler wins the game immediately.

Assume $\varphi(z_1, z_{k+1})$ has quantifier depth $m+1$. If $\varphi = \neg\varphi'$, then φ' distinguishes (u, v) from (u', v'), too. If $\varphi = \varphi_1 \wedge \varphi_2$, one formula of φ_1 and φ_2 distinguishes (u, v) from (u', v'). Hence we can assume that φ is a walk quantifier:

$$\varphi(z_1, z_{k+1}) = \exists^j z_2, \ldots, z_k. \bigwedge_{i \in [k]} \varphi_i(z_i, z_{i+1}).$$

Assume w.l.o.g. that $\varphi_{G_1}(u, v)$ is true but $\varphi_{G_2}(u', v')$ not.

Duplicator chooses a bijection $f\colon V_1^{k-1} \to V_2^{k-1}$. There must be a tuple $(w_2, \ldots, w_k) \in V_1^{k-1}$ serving as witness of the quantifier in G_1 but $f(w_2, \ldots, w_k) = (w'_2, \ldots, w'_k)$ does not serve as witness for G_2, because otherwise $\varphi_{G_2}(u', v')$ was true. Then Spoiler places for $2 \leq i \leq k$ the pebble p_i on w_i and pebble q_i on w'_i.

Now, there must be an $i \in [k]$ such that $\varphi_i(w_i, w_{i+1})$ is true but $\varphi_i(w'_i, w'_{i+1})$ is not, because otherwise $(w'_2, \ldots, w'_k)$ is a witness. Now φ_i is of quantifier depth m and the i-th and $(i+1)$-th pebble pairs are placed on the correct vertices. Thus Spoiler removes all other pebbles and wins the game in additional m rounds by induction hypothesis. ∎

Lemma 16. *If there is a $\mathcal{W}_k$ sentence φ of quantifier depth m distinguishing G_1 and G_2, then Spoiler has a winning strategy in m rounds in the bijective k-walk pebble game.*

Proof: Assume φ is a sentence of quantifier depth $m > 0$ distinguishing G_1 and G_2. For the same reasons as in Lemma 15, we can assume that φ is a walk-quantifier:

$$\varphi = \exists^j z_1, \ldots, z_{k-1}. \bigwedge_{i \in [k-2]} \varphi_i(z_i, z_{i+1}).$$

Again as in Lemma 15, for each bijection $f\colon V_1^{k-1} \to V_2^{k-1}$ there is a witness $(w_1, \ldots, w_{k-1}) \in V_1^{k-1}$ of G_1 such that $f(w'_1, \ldots, w'_{k-1}) = (v_1, \ldots, v_{k-1})$ is not a witness of G_2. Again, there is a $j \in [k-2]$ such that φ_i distinguishes (w_j, w_{j+1}) and (w'_j, w'_{j+1}).

When Spoiler places the p_i pebbles on the w_i and the q_i pebbles on the w'_i, Spoiler can force a win in additional $m-1$ rounds by Lemma 15. So overall she has a winning strategy in m rounds. ∎

Lemma 17. *Let $u, v \in V_1$ and $u', v' \in V_2$. If the bijective k-walk pebble game distinguishes (u, v) from (u', v') in m rounds, then m iterations of k-walk refinement distinguish them.*

Proof: Assume that the pebble pairs (p_1, q_1) and (p_{k+1}, q_{k+1}) are placed on (u, v) and (u', v') and Spoiler has a winning strategy in additional m rounds. Let χ and χ' be the initial colorings of the graphs G_1 and G_2. The proof proceeds by induction on m.

If $m = 0$, (u, v) is an edge, non-edge, or loop, (u', v') is not of the same type, and hence $\chi(u, v) \neq \chi'(u', v')$.

Assume Spoiler can force a win of the game in additional $m+1$ rounds. Whatever bijection Duplicator chooses, Spoiler can place the pebbles such that she can force a win in m additional rounds. That means, by inductive hypothesis, that for every bijective mapping between the k-walks from u to v in $\chi^m_{\mathcal{W}[k]}$ and the k-walks from u' to v' in $(\chi')^m_{\mathcal{W}[k]}$ there is a walk that is mapped to a walk of different color.

Hence, there is a k-walk color that occurs with different multiplicity from u to v in $\chi^m_{\mathcal{W}[k]}$ than from u' to v' in $(\chi')^m_{\mathcal{W}[k]}$. This just says that the vertex pairs obtain different colors in $\chi^{m+1}_{\mathcal{W}[k]}$ respectively$(\chi')^{m+1}_{\mathcal{W}[k]}$. ∎

Lemma 18. *If Spoiler can force a win in the bijective k-walk pebble game in m rounds, then the k-walk refinement distinguishes the graphs G_1 and G_2 after m iterations.*

Proof: At the beginning of the game, Duplicator chooses a bijection. For every such bijection, Spoiler can place the pebbles such that she can force a win in additional $m-1$ rounds. As in and by Lemma 17 there is no bijective mapping between the walks on vertices of G_1 and those on G_2 such that assigned walks have the same color after $m-1$ iterations of k-walk refinement.

But then m iterations distinguish the graphs because if not, such a mapping would always exist. ∎

Theorem 19. *Two graphs G_1 and G_2 are distinguished by k-walk refinement if and only if they are distinguished by $\mathcal{W}_k$ if and only if Spoiler has a winning strategy in the bijective k-walk pebble game.*

Corollary 20. *Two graphs G_1 and G_2 are distinguished by walk refinement if and only if they are distinguished by walk counting logic if and only if Spoiler has a winning strategy in the bijective walk pebble game.*

These equivalences in particular imply that the upper bound for the walk refinement (Theorem 11) translates to the game and logic scenarios as follows.

Corollary 21. *If Spoiler has a winning strategy in the bijective walk-pebble game on two graphs G_1 and G_2, then she has a winning strategy requiring $\mathcal{O}(n)$ rounds.*

Corollary 22. *If two graphs G_1 and G_2 are distinguished by walk counting logic, then they can be distinguished by a walk counting logic sentence of quantifier depth $\mathcal{O}(n)$.*

VIII. A Linear Lower Bound for Walk Refinement

In this section we show that there are graphs on which walk refinement stabilizes only after $\Omega(n)$ iterations. Specifically, we show this for the same graphs, for which Fürer already showed that the WL refinement requires $\Omega(n)$ iterations [3]. We do this by demonstrating that Duplicator has a strategy in the bijective walk pebble game played on these graphs that delays the win of Spoiler for at least $\Omega(n)$ rounds.

In the following we recall well-known constructions and their properties from [2] and [3]. For proofs of these properties we refer the reader to the original papers.

A. CFI-Construction

The graphs used by Fürer in [3] to prove the lower bound are obtained by taking suitable base graphs and replacing each vertex with a special gadget. We first describe these gadgets and their properties.

Let $G = (V, E)$ be a simple connected *base graph*. We call the vertices and edges in the base graph *base vertices* and *base edges*, respectively. Each base vertex will be replaced by a gadget (a small graph) and each base edge e will be represented by edges between the gadgets corresponding to the endpoints of e.

Cai, Fürer, and Immerman introduced the so called CFI-gadgets [2] consisting of outer and middle vertices, where a base edge results in edges between the outer vertices of two gadgets. However, in [3] Fürer uses a variant of these gadgets only consisting of the middle vertices. He directly connects the middle vertices of two gadgets. In this paper we follow this approach because it simplifies our reasoning for the bijective walk pebble game (see Figures 1 and 2).

A gadget F_d of degree d consists of all d tuples $\{0,1\}^d$ with an even number of ones as vertices. The gadget has no edges.

Let $v \in V$ be a base vertex of degree d. When replacing v with a gadget of degree d, we denote with $v(a_1, \dots, a_d)$, where $a_i \in \{0,1\}$, the vertices of the gadget (of course we have $a_i = 1$ for an even number of a_i).

We fix arbitrarily for each base vertex v an ordering of its incident edges, so that we can speak of the i-th base edge incident to v. The undirected graph $X(G)$ is obtained from the base graph G by replacing every base vertex v with a gadget $F(v)$ of degree $d(v)$ and connecting gadgets arising from adjacent base vertices as follows: Let $e = \{u, v\}$ be a base edge, let u and v have degree d and d' respectively, and assume that e is the i-th incident edge of u and the j-th incident edge of v. We then insert the edges $\{\{u(a_1, \dots, a_d), v(b_1, \dots, b_{d'})\} \mid a_i = b_j\}$ between vertices that agree on the i-th and j-th component, respectively. That is, a base edge is represented in $X(G)$ by two complete bipartite induced subgraphs, one for $a_i = b_j = 0$ and one for $a_i = b_j = 1$ (these subgraphs may be empty if one of the both base vertices has degree 1).

To *twist* a base edge $\{u, v\} \in E$ means to replace every edge between a vertex of the gadget $F(u)$ and a vertex of the gadget $F(v)$ by a nonedge and vice versa. We obtain from $X(G)$ another graph $\tilde{X}(G)$ by *twisting* some arbitrary base edge. The graph $\tilde{X}(G)$ is well defined up to isomorphism, because the graph obtained from $X(G)$ by twisting another edge is always isomorphic to $\tilde{X}(G)$. If there is at least one base edge, then $X(G)$ is not isomorphic to $\tilde{X}(G)$.

We say that $v(a_1, \dots, a_d)$ *originates* from v or that the *origin* of $v(a_1, \dots, a_d)$ is v. Likewise, we say that an edge $\{u(a_1, \dots, a_d), v(b_1, \dots, b_{d'})\}$ *originates* from $\{u, v\}$ or has *origin* $\{u, v\}$. We extend this notion to sets and walks: A set of vertices in $X(G)$ (or $\tilde{X}(G)$) originates from the set of origins and a walk originates from the walk consisting of the origins of the visited vertices.

In the following, we will use $\bar{u}, \bar{v}$, and $\bar{w}$ for vertices of $X(G)$ with origins u, v, and w respectively. Similarly, we use $\tilde{u}, \tilde{v}$, and $\tilde{w}$ for vertices of $\tilde{X}(G)$. Let $X(G) = (\bar{V}, \bar{E})$ and $\tilde{X}(G) = (\tilde{V}, \tilde{E})$ and note that $\bar{V} = \tilde{V}$ and hence we can reinterpret an automorphism of $\tilde{X}(G)$ as a mapping between $X(G)$ and $\tilde{X}(G)$. We say that an automorphism φ of $\tilde{X}(G)$ *moves the twist* to the base edge $\{w, w'\} \in E$ if

$$\{\bar{u}, \bar{v}\} \in \bar{E} \Leftrightarrow \{\varphi(\bar{u}), \varphi(\bar{v})\} \in \varphi(\tilde{E})$$

for all $\{\bar{u}, \bar{v}\}$ not originating from $\{w, w'\}$ and

$$\{\bar{u}, \bar{v}\} \in \bar{E} \Leftrightarrow \{\varphi(\bar{u}), \varphi(\bar{v})\} \notin \varphi(\tilde{E})$$

for all $\{\bar{u}, \bar{v}\}$ originating from $\{w, w'\}$.

That is, φ behaves like an isomorphism except on vertex pairs originating from the base edge $\{w, w'\}$, on which φ inverts adjacency. For every base edge, there is an automorphism of $\tilde{X}(G)$ moving the twist to that edge (which is just another way of saying that the graph obtained by twisting some edge in $X(G)$ is always isomorphic to $\tilde{X}(G)$). Assume that φ moves the twist to $\{u, v\}$ and we want to move the twist to $\{v, w\}$. Then there is another automorphism ψ which possibly permutes the vertices of $F(v)$ but is otherwise constant such that $\psi \circ \varphi$ moves the twist to $\{v, w\}$. In general, not every automorphism moves the twist to a single base edge, but to an odd number of bases edges (on which it inverts adjacency). In the following we only consider automorphisms moving the twist to a single base edge.

B. Lower Bound for the Weisfeiler-Leman Refinement

We recall the necessary parts of Fürer's lower bound on the iteration number of the d-dimensional WL refinement. We only deal with the 2-dimensional case.

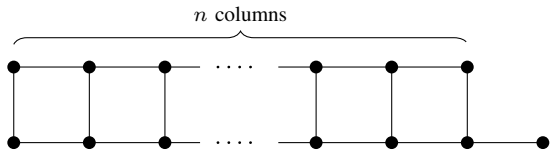

Figure 1. The base graph G_n^2.

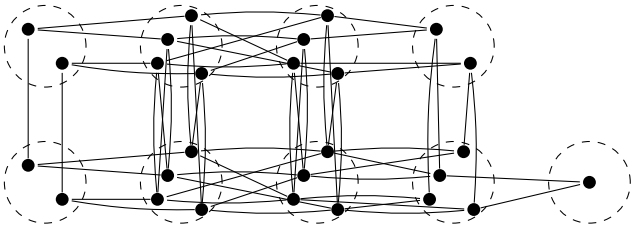

Figure 2. The graph G_4^2. A circle indicates a base vertex and hence contains one gadget.

Let G_n^2 be a $2 \times n$ grid with an additional vertex attached to one corner (depicted in Figure 1). In this graph all vertices can be uniquely identified by their distance to the unique vertex of degree 1 as well as the distance to the two adjacent vertices of degree 2 (given that $n \geq 3$). By replacing with gadgets as described, we obtain two graphs $X(G_n^2)$ and $\tilde{X}(G_n^2)$. The graph $X(G_4^2)$ is shown in Figure 2. The graphs $X(G_n^2)$ and $\tilde{X}(G_n^2)$ are not isomorphic and can be distinguished by the Weisfeiler-Leman refinement. Hence, Spoiler has a winning strategy in the bijective 3-pebble game and consequently also in the bijective walk pebble game. Note that, since vertices of G_n^2 have degree at most 3, $X(G_n^2)$ and $\tilde{X}(G_n^2)$ have $\Theta(n)$ vertices.

We recall some facts for the bijective 3-pebble game played on the graphs from [3]: Assume that in the progress of the game some pebble pairs are placed on the graphs. Then we call an isomorphism φ between two graphs *pebble respecting* if v is covered by pebble p_i if and only if $\varphi(v)$ is covered by q_i. In case φ is an automorphism it necessarily maps all vertices covered by pebbles to themselves.

Intuitively, since Duplicator can move the twist to different edges using automorphisms, Spoiler needs to "catch" the twist with her pebbles. If a vertex is placed on a vertex $\tilde{v}$ of $\tilde{X}(G_n^2)$ with origin v, all pebble respecting automorphisms of $\tilde{X}(G_n^2)$ fix all vertices in $F(v)$. Hence, it does not matter on which vertex originating from v Spoiler places a pebble and we can simply say that Spoiler places a pebble on v.

A set of vertices of $X(G_n^2)$ (or $\tilde{X}(G_n^2)$) is called a *wall* if its origin is a separator of G_n^2, that is, a set of vertices whose removal separates the graph into at least two connected components. We say that Spoiler *builds a wall*, if the vertices covered by the pebbles form a wall. To avoid a quick win for Spoiler, Duplicator picks the bijection on her turn so that it is origin respecting. That is, Duplicator maps a vertex $\bar{u}$ to a vertex $\tilde{u}$ with the same origin. Our strategy for Duplicator in the bijective walk-pebble game described below has this property, too. Consequently, when asking whether the pebbles form a wall or not, it does not matter whether we consider the pebbles on $X(G_n^2)$ or $\tilde{X}(G_n^2)$). Since we will only consider origin respecting strategies, we will often simply think of the origins being pebbled.

Suppose now some vertices of G_n^2 have been pebbled. A *component* of the graph G_n^2, w.r.t. the pebbled vertices, is an inclusion-wise maximal and nonempty set of base edges C satisfying the following property: For every two edges $e_1, e_2 \in C$, there is a walk $(v_1, \ldots, v_j)$ only using edges $\{v_i, v_{i+1}\} \in C$ for all $i \in [j-1]$ such that $e_1 = \{v_1, v_2\}$, $e_2 = \{v_{j-1}, v_j\}$, and $v_2, \ldots, v_{j-1}$ are not covered by a pebble. A component C *contains* a base vertex v, if all base edges incident to v are contained in C. The *size* of the component is the number of vertices it contains. We call a component *nontrivial*, when its size is nonzero.

Intuitively, one can think of components as the parts of the graph G_n^2 obtained by deleting only the vertices covered by pebbles, but not the edges incident to these vertices. This results in "dangling" edges in nontrivial components (edges, which were incident to only one vertex covered by a pebble) and edges not incident to any vertex forming the trivial components (edges, both endpoints of which where covered by a pebble).

We call a component C *twisted*, if there is a pebble respecting automorphism that moves the twist to an edge in C. If C is twisted, every pebble respecting automorphism moves the twist to an edge in C, i.e., a twisted component contains precisely the edges, to which Duplicator can move the twist with pebble respecting automorphisms.

With 3 pebbles Spoiler can build at most one wall and hence in the bijective 3-pebble game there are at most two nontrivial components. When a trivial component is twisted, Spoiler wins the game. To delay the win of Spoiler, Duplicator maintains a single twisted component, whose size only decreases by a constant per round.

C. Lower Bound for Walk Refinement

The situation changes in the bijective walk pebble game, since the game does not have a bound on the number of pebbles that are used. In the situation where more than two pebbles are placed on the graph, there can be many components (Spoiler may in particular cover every vertex and every edge). But, once she has to remove all but two pebbles, there can be at most one wall again. We describe a strategy of Duplicator with the following properties:

1) If the size of the twisted component is at most $n-2$, its size reduces by at most 2 after one round.
2) If the size of the twisted component is greater than $n-2$ (e.g. in the beginning of the game), the size of the twisted component is at least $n-4$ after one round.

Combining these properties, Spoiler needs at least $\Omega(n)$ rounds to win. Intuitively, the existence of such a strategy comes from the fact that in the bijective walk pebble game Duplicator needs to preserve adjacency and/or equality only for consecutive pebble pairs. This allows her to fix the twist

locally, possibly introducing global inconsistencies but never introducing inconsistencies between consecutive pebble pairs. We describe a strategy how Duplicator can introduce these inconsistencies only on edges incident to a chosen base vertex.

Suppose we are in the bijective k-walk pebble game for an arbitrary k and we are in the state of the game in which only two pebble pairs are placed on the graphs. Assume that the pebbles are placed on $\bar{u}_1$ and $\bar{u}_2$ in $X(G_n^2)$ and on $\tilde{u}_1$ and $\tilde{u}_2$ in $\tilde{X}(G_n^2)$ respectively. We assume, justifying our notation, that $\bar{u}_i$ and $\tilde{u}_i$ have the same origin for $i \in \{1,2\}$ and that Spoiler has not won the game already, i.e., the pebbles define an isomorphism between the subgraphs induced by the pebbled vertices.

Let v be a base vertex of degree 3 in the twisted component and e_1 and e_2 be two distinct base edges incident to v. We define a bijection $f^v_{e_1,e_2} \colon \bar{V}^{k-1} \to \tilde{V}^{k-1}$ as follows:

First, pick a pebble respecting isomorphism φ of $\tilde{X}(G_n^2)$ moving the twist to e_1 (it exists because v is in the twisted component). Let $(\bar{w}_1, \dots, \bar{w}_{k-1}) \in \bar{V}^{k-1}$ and set $\bar{w}_0 := \bar{u}_1$ and $\bar{w}_k := \bar{u}_2$. We define $f^v_{e_1,e_2}(\bar{w}_1, \dots, \bar{w}_{k-1}) = (\tilde{w}_1, \dots, \tilde{w}_{k-1})$ entry-wise for $i \in [k-1]$ by the following case distinction:

1) If $w_i \neq v$, we set $\tilde{w}_i := \varphi(\bar{w}_i)$.
2) If $w_i = v$, let $j < i$ and $\ell > i$ be the unique indices such that $w_j \neq w_{j+1} = \dots = w_i = \dots = w_{\ell-1} \neq w_\ell$. We pick an edge e incident to v by a second case distinction:
   a) If $e_1 := \{w_j, w_i\}$ and $e_2 := \{w_i, w_\ell\}$ are distinct base edges, let $e \notin \{e_1, e_2\}$ be the third base edge incident to v.
   b) Otherwise at least one of $\{w_j, w_i\}$ and $\{w_i, w_\ell\}$ is not a base edge or $\{w_j, w_i\} = \{w_i, w_\ell\}$. Hence, when passing through v at position i, the walk uses at most one base edge e' incident to v. Let the edge $e \in \{e_1, e_2\} \setminus \{e'\}$ be smallest one according to the fixed order of base edges incident with v.

   Finally, let ψ be the automorphism of $\tilde{X}(G_n^2)$ such that $\psi \circ \varphi$ moves the twist to e and ψ is the identity on all vertices apart from those in the gadget $F(v) = F(w_i)$. We set $\tilde{w}_i := \psi(\varphi(\bar{w}_i))$.

Lemma 23. *The function $f^v_{e_1,e_2}$ is a bijection.*

Proof: The function $f^v_{e_1,e_2}$ maps a walk in $X(G_n^2)$ to a walk in $\tilde{X}(G_n^2)$ with the same origin. For two walks in $X(G_n^2)$ with the same origin, Duplicator chooses for each of the walks the same additional automorphisms ψ. Since φ and all chosen ψ are bijections, the map $f^v_{e_1,e_2}$ is a bijection. ∎

Lemma 24. *If Duplicator chooses $f^v_{e_1,e_2}$ as bijection, then Spoiler does not win in the current round.*

Proof: Assume Spoiler picks some $(\bar{w}_1, \dots, \bar{w}_{k-1})$ and $f^v_{e_1,e_2}(\bar{w}_1, \dots, \bar{w}_{k-1}) = (\tilde{w}_1, \dots, \tilde{w}_{k-1})$. Then the p_i pebbles are placed on the $\bar{w}_i$ and the q_i pebbles are placed on the $\tilde{w}_i$. We set $\bar{w}_0 := \bar{u}_1$, $\tilde{w}_0 := \varphi(\bar{u}_1)$ and likewise $\bar{w}_k := \bar{u}_2$ and $\tilde{w}_k = \varphi(\bar{u}_2)$. Note that $\bar{w}_0$ and $\tilde{w}_0$ are covered by the first pebble pair and $\bar{w}_k$ and $\tilde{w}_k$ by the last one because φ was chosen pebble respecting. Suppose $i \in \{0, \dots, k-1\}$. To show that Spoiler does not win in this round, we show that the i-th and $(i+1)$-th pebble pair define an isomorphism between the subgraphs of $X(G_n^2)$ and $\tilde{X}(G_n^2)$ induced by the vertices covered by the i-th and $(i+1)$-th pebble pair.

- Suppose $\bar{w}_i$ and $\bar{w}_{i+1}$ do no originate from v and thus $\tilde{w}_i = \varphi(\bar{w}_i)$ and $\tilde{w}_{i+1} = \varphi(\bar{w}_{i+1})$. Then $\{\bar{w}_i, \bar{w}_{i+1}\}$ in particular does not originate from e_1 and hence by the choice of φ the pebbles define an isomorphism of the induced subgraphs.
- Suppose $\bar{w}_i$ and $\bar{w}_{i+1}$ both originate from v. In this case $\tilde{w}_i = \psi(\varphi(\bar{w}_i))$ and $\tilde{w}_{i+1} = \psi(\varphi(\bar{w}_{i+1}))$ for some automorphism ψ. Because both φ and ψ are bijections and inside a gadget there are no edges, the pebbles define an isomorphism.
- Lastly, suppose $\bar{w}_i$ originates from v but $\bar{w}_{i+1}$ does not. In this case $\tilde{w}_i = \psi(\varphi(\bar{w}_i))$ and $\tilde{w}_{i+1} = \varphi(\bar{w}_{i+1})$ for another automorphism ψ such that $\psi \circ \varphi$ moves the twist to an edge other than $\{v, w_{i+1}\}$. Therefore $\{\bar{w}_i, \bar{w}_{i+1}\} \in \bar{E}$ holds if and only if $\{\psi(\varphi(\bar{w}_i)), \psi(\varphi(\bar{w}_{i+1}))\} = \{\tilde{w}_i, \tilde{w}_{i+1}\} \in \varphi(\tilde{E})$. Recall that ψ was chosen constant on all vertices apart $F(v)$ and thus $\tilde{w}_{i+1} = \varphi(\bar{w}_{i+1}) = \psi(\varphi(\bar{w}_{i+1}))$. The case where $\bar{w}_{i+1}$ originates from v but $\bar{w}_i$ does not is symmetric. ∎

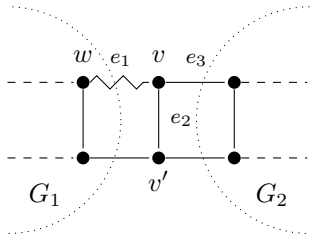

Figure 3. The situation in Lemma 25: dashed edges may exist but not have to and the two subgraphs G_1 and G_2 are indicated by dotted lines. The isomorphism φ moves the twist to e_1.

Lemma 25. *Let $e_1 = \{v, w\}$ and $e_2 = \{v, v'\}$ be base edges, $v' \neq w$, and $\{v, v'\}$ be a separator of G_n^2 separating G_n^2 into two subgraphs G_1 and G_2. Assume that w is contained in G_1 and that G_2 has ℓ vertices.*

If Duplicator chooses $f^v_{e_1,e_2}$ as bijection, Spoiler pebbles two corresponding walks and then removes all pebble pairs apart from two consecutive ones, then the new twisted component has size at least $\min\{\ell, 2n - \ell - 2\}$.

Proof: The situation of the lemma is shown in Figure 3. If G_2 contains ℓ vertices, then G_1 contains $\ell' := 2n - \ell - 1$ vertices. We first note that if there is no wall, the twisted component has size $2n - 1$. In the case of a wall, we make the following case distinction:

- The vertex v is not covered by a pebble. Then φ is still a pebble respecting automorphism moving the twist to e_1 and thus v is in the twisted component. It has size at least $\min\{\ell, \ell'\} + 1$.

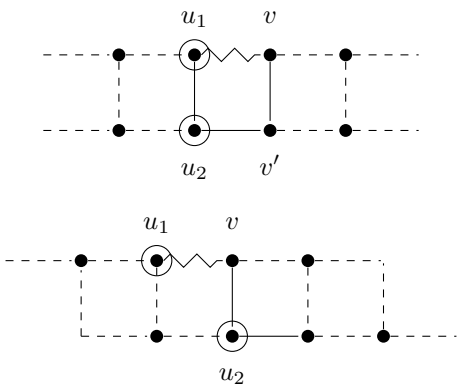

Figure 4. The two possible situations in Theorem 26: the base vertices u_1 and u_2 are covered by pebbles (indicated by circles) and form a wall. Dashed edges can exists, but do not have to. The automorphism φ moves the twist to $\{u_1, v\}$ and the twisted component is to the right of the wall.

- The vertex v and an adjacent vertex u are covered by pebbles. If $\{v, u\} \in \{e_1, e_3\}$ then there is no wall. So $u = v'$ and thus the twist can be moved to e_1 or e_3 by the choice of ψ in the construction of $f^v_{e_1,e_2}$. Then the twisted component has size $\min\{\ell, \ell'\}$.
- The vertex v and a non-adjacent vertex u are covered by pebbles. In the construction of $f^v_{e_1,e_2}$ the automorphism ψ is chosen such that the twist is moved to e_1 or e_2. If u is in G_1, the twisted component has size $\ell + 1$ if the twist was moved to e_2 and size $\ell' - 1$ is the twist was moved to e_1. Otherwise u is in G_2. Since the twist can be moved to e_1 or e_2, the twisted component has size $\ell + 1$. ■

Theorem 26. *For every $k \geq 2$ and $n \geq 3$ Duplicator has a strategy in the bijective k-walk pebble game played on the graphs $X(G_n^2)$ and $\tilde{X}(G_n^2)$ such that Spoiler wins in $\Omega(n)$ rounds at the earliest.*

Proof: We show that Duplicator has a strategy such that after m rounds the twisted component of G_n^2 is of size at least $n-2(m+1)$ and Spoiler can win only if its size is at most 2.

Assume that after m rounds the twisted component has size at least $n - 2(m + 1) > 2$, that there are only two pebbles on the graphs, and its Duplicator's turn to pick a bijection. There are two cases:

- The twisted component has size at most $n - 2$. Then in particular Spoiler builds a wall. Let the pebbles be placed on base vertices u_1 and u_2. Duplicator picks v as depicted in Figure 4: If u_1 and u_2 are adjacent, v is the neighbor of u_1 in the twisted component and v' is the common neighbor of v and u_2. Duplicator chooses $f^v_{\{u_1,v\},\{v,v'\}}$ as bijection. If otherwise u_1 and u_2 are not adjacent, v is the neighbor of both u_1 and u_2 in the twisted component. We possibly exchange names of u_1 and u_2 so that $\{u_2, v\}$ is a separator of the graph. Then Duplicator chooses $f^v_{\{u_1,v\},\{u_2,v\}}$ as bijection. By Lemma 24, Spoiler does not win in the current round and by Lemma 25 the size of the new twisted component is at least $n - 2(m + 2)$.
- The twisted component has size greater than $n - 2$. Let $\{u_1, u_2\}$ be a base edge and a separator of G_n^2 separating G_n^2 into two subgraphs containing at least $n - 2$ vertices and let u_1 have a neighbor of degree 3 in the twisted component. Such a separator exists because the size of the twisted component is greater than $n - 2$. Duplicator proceeds with this choice of u_1, u_2, and v as in the case before[1]. After the current round, the twisted component has size at least $n - 4$. ■

With Lemmas 14 and 15 we obtain the following corollaries:

Corollary 27. *A walk counting logic formula distinguishing $X(G_n^2)$ and $\tilde{X}(G_n^2)$ has quantifier depth $\Omega(n)$.*

Corollary 28. *Walk refinement distinguishes $X(G_n^2)$ and $\tilde{X}(G_n^2)$ in $\Omega(n)$ iterations. In particular, walk refinement stabilizes on $X(G_n^2)$ as well as $\tilde{X}(G_n^2)$ in $\Omega(n)$ iterations.*

Recall that $X(G_n^2)$ and $\tilde{X}(G_n^2)$ have $\Theta(n)$ vertices, so both corollaries give bounds that are linear in the number of vertices.

We want to remark that 4 pebble pairs already suffice for Spoiler to reduce the size of the twisted component by 2 in each round: Spoiler places the pebbles always on four vertices originating from a 4-cycle starting in the middle of the graph. Then she moves two of them so that together the 4 pebbles cover a 4-cycle that shares an edge with the 4-cycle of the previous round. Overall, this strategy requires $n/2 + 1$ rounds if n is even and $(n + 1)/2 + 1$ rounds otherwise. In particular, 4-walk refinement has the same iteration number as walk-refinement on these graphs. Nevertheless, after only one iteration, n-walk refinement distinguishes vertices of different gadgets, but 4-walk refinement does not.

With only 3 pebble pairs the size reduces by at most one per iteration (as already shown in [3]).

IX. Conclusion

We showed that the 2-dimensional Weisfeiler-Leman refinement stabilizes an n-vertex graph after $\mathcal{O}(n \log n)$ iterations. Hence in the counting logic $\mathcal{C}_3$ we only require a quantifier depth of $\mathcal{O}(n \log n)$. This matches the best known lower bound of the form $\Omega(n)$ up to a logarithmic factor. Thus the question remains what the precise bound is, and whether the iteration number can be superlinear. At least for the walk refinement we have now matching linear lower and upper bounds.

It remains also an open problem whether our techniques can be applied to counting first order logic with more than three variables (equivalently higher dimensional Weisfeiler-Leman refinement) or to three variable first order logic without counting.

For all of these mentioned avenues of investigation it could be interesting to find a combinatorial argument for the $\mathcal{O}(n)$ bound for walk refinement. Finally, we also introduced walk counting logic and the bijective walk pebble game and studying these remains as future work.

[1]Formally, our construction of $f^v_{e_1,e_2}$ requires two pebble pairs to be placed. If they are not placed already, we just pretend that they are placed on u_1 and u_2.

References

[1] "Symmetry vs regularity the first 50 years since Weisfeiler-Leman stabilization," https://www.iti.zcu.cz/wl2018/index.html, accessed: 2018-11-26.

[2] J. Cai, M. Fürer, and N. Immerman, "An optimal lower bound on the number of variables for graph identification," *Combinatorica*, vol. 12, no. 4, pp. 389–410, 1992.

[3] M. Fürer, "Weisfeiler-Lehman refinement requires at least a linear number of iterations," in *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2076. Springer, 2001, pp. 322–333.

[4] S. Kiefer and P. Schweitzer, "Upper bounds on the quantifier depth for graph differentiation in first order logic," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*. ACM, 2016, pp. 287–296.

[5] D. G. Higman, "Coherent algebras," *Linear Algebra and its Applications*, vol. 93, pp. 209–239, 1987.

[6] B. Weisfeiler, *On construction and identification of graphs*, ser. Lecture Notes in Mathematics, Vol. 558. Springer-Verlag, Berlin-New York, 1976.

[7] L. Babai, "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*. ACM, 2016, pp. 684–697.

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

[9] I. N. Ponomarenko, "The isomorphism problem for classes of graphs that are invariant with respect to contraction," *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, vol. 174, no. Teor. Slozhn. Vychisl. 3, pp. 147–177, 182, 1988.

[10] C. Berkholz and M. Grohe, "Linear diophantine equations, group CSPs, and graph isomorphism," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*. SIAM, 2017, pp. 327–339.

[11] C. Berkholz and J. Nordström, "Near-optimal lower bounds on quantifier depth and Weisfeiler-Leman refinement steps," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, 2016, pp. 267–276.

[12] C. Berkholz, "The propagation depth of local consistency," in *Principles and Practice of Constraint Programming - 20th International Conference, CP 2014, Lyon, France, September 8-12, 2014. Proceedings*, 2014, pp. 158–173.

[13] L. Hella, "Logical hierarchies in PTIME," *Information and Computation*, vol. 129, no. 1, pp. 1–19, 1996.

[14] A. Zimmermann, *Representation Theory: A Homological Algebra Point of View*, ser. Algebra and Applications. Springer International Publishing, 2014.

[15] A. A. Albert, *Modern higher algebra*. Chicago, Ill.: Univ. of Chicago Press, 1937.

[16] Y. Shitov, "An improved bound for the length of matrix algebras," *ArXiv e-prints*, Jul. 2018, https://arxiv.org/abs/1807.09310.