Weisfeiler-Lehman Refinement Requires at Least a Linear Number of Iterations

Martin Fürer*

Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802, USA
`furer@cse.psu.edu`,
`http://www.cse.psu.edu/~furer`

Abstract. Let $\mathcal{L}_{k,m}$ be the set of formulas of first order logic containing only variables from $\{x_1, x_2, \ldots, x_k\}$ and having quantifier depth at most m. Let $\mathcal{C}_{k,m}$ be the extension of $\mathcal{L}_{k,m}$ obtained by allowing counting quantifiers $\exists i x_j$, meaning that there are at least i distinct x_j's.

It is shown that for constants $h \geq 1$, there are pairs of graphs such that h-dimensional Weisfeiler-Lehman refinement (h-dim W-L) can distinguish the two graphs, but requires at least a linear number of iterations. Despite of this slow progress, $2h$-dim W-L only requires $O(\sqrt{n})$ iterations, and $3h-1$-dim W-L only requires $O(\log n)$ iterations. In terms of logic, this means that there is a $c > 0$ and a class of non-isomorphic pairs (G_n^h, H_n^h) of graphs with G_n^h and H_n^h having $O(n)$ vertices such that the same sentences of $\mathcal{L}_{h+1,cn}$ and $\mathcal{C}_{h+1,cn}$ hold ($h+1$ variables, depth cn), even though G_n^h and H_n^h can already be distinguished by a sentence of $\mathcal{L}_{k,m}$ and thus $\mathcal{C}_{k,m}$ for some $k > h$ and $m = O(\log n)$.

Keywords: Graph Isomorphism Testing, Weisfeiler-Lehman Refinement, Games, Descriptive Complexity

1 Introduction

A simple and important preprocessing procedure for the graph isomorphism problem is the k-dimensional Weisfeiler-Lehman refinement (k-dim W-L). The algorithm tries to color k-tuples of vertices with different colors, if they belong to different orbits of the automorphism group. This goal is not always achieved. If two k-tuples have the same color, it is still possible that no automorphism maps one to the other, but the algorithm has not discovered a significant difference between the two k-tuples. On the other hand, if two k-tuples have different colors, then they always belong to different orbits.

For $k = 1$, this is the straightforward vertex classification algorithm where vertices are initially colored by their degrees. During every later refinement step,

---

* Research supported in part by NSF Grant CCR-9700053

F. Orejas, P.G. Spirakis, and J. van Leeuwen (Eds.): ICALP 2001, LNCS 2076, pp. 322–333, 2001.

each vertex is colored by the multi-set of the colors of its neighbors. The process stops, when no color class is split anymore.

The case $k = 2$ has also been well studied. It is edge coloring. The algorithm starts with three classes of pairs of vertices: pairs (u, v) with or without an edge, and pairs (u, u). During each refinement step, every directed edge (u, v) is colored by the multi-set of pairs of colors on paths of length two from u to v.

As an example, consider the path of length $n - 1$. Applying 1-dim W-L, the vertices of distance d from an endpoint receive their unique color during step d. The algorithm stops when every vertex "knows" its distance from its closer endpoint. Obviously, this requires $\Theta(n)$ iterations. Using 2-dim W-L, distances up to 2^s are measured in s steps. After only $\log n$ steps, the color of (u, u) (which may be interpreted as the color of vertex u) determines the distance of u from the closer endpoint.

This and other examples suggest, that for $k > 1$, k-dim W-L might always run in just $O(\log n)$ rounds for graphs of size n. In particular, it is very suggestive to make this conjecture for $k = 2$, because this case allows an algebraic treatment. Indeed, it has initiated a vast development in algebra (cellular algebras [11,4] and coherent configurations [7]). It is easy to see that 2-dim W-L corresponds to squaring a matrix A of indeterminates and replacing identical expressions by the same new indeterminate (starting with a modified adjacency matrix where 3 different indeterminates are used for edges, non-edges and diagonal elements).

Assume, instead of this special "squaring" operation, one would do a sequence of corresponding "multiplications" by A. As there can be at most n^2 colors of vertex pairs, this process would stop at the latest with A^{n^2-2}. All higher "powers" would be equal to this one. As a result of this reasoning, one might jump to the conclusion that $O(\log n)$ squaring operations were always sufficient. We will show in this paper that this is not at all the case. This somewhat counterintuitive result is possible, because the just described matrix "product" is not associative.

Section 2 reviews some background information on the basic techniques connecting Weisfeiler-Lehman refinement to logic and games. Section 3 presents the examples for which upper and lower bounds will be proved in Section 5. A simplified view of the pebble games is discussed in Section 4.

2 The Cai-Fürer-Immerman Method

The strength of k-dim W-L has long been an open problem. It has been difficult to find graphs, for which k-dim W-L does not succeed immediately. Already 1-dim W-L identifies random graphs in linear time [1]. For regular graphs, 1-dim W-L cannot even get started. But 2-dim W-L is strong enough to identify shortest cycles and classify the vertices by their distance from the set of vertices covered by shortest cycles. Refining this classification is likely to identify random regular graphs [10] in linear time. It seemed reasonable to conjecture that $f(k)$-dim W-L could identify all degree k graphs for some slow growing function f, e.g., $f(k) = k$. Cai, Fürer, and Immerman [2] have shown that this is very far from the truth. Indeed $k = \Omega(n)$ is required for graphs of degree 3. We use a modification of their counter-examples to produce graphs which can be identified by k-dim W-L, but only after a linear number of iterations.

Cai, Fürer, and Immerman [2] exhibit an intimate connection between three different approaches to the graph isomorphism problem. These approaches are based on Weisfeiler-Lehman refinement, descriptional complexity, and a version of Ehrenfeucht-Fraïssé games [3,5].

To understand the present paper, it is required to know many definitions and techniques from the Cai, Fürer, and Immerman [2] paper. We start by reviewing some of these notions and their applications.

2.1 Logic Background

Definition 1. For a given language $\mathcal{L}$, the graphs G and H are *$\mathcal{L}$-equivalent* ($G \equiv_{\mathcal{L}} H$) iff the same sentences of $\mathcal{L}$ hold for G and H. Formally, this is expressed as

$$G \models \varphi \;\Leftrightarrow\; H \models \varphi\,.$$

for all sentences $\varphi \in \mathcal{L}$.

We say that $\mathcal{L}$ *identifies* the graph G, if $G \equiv_{\mathcal{L}} H$ implies G and H are isomorphic.

We define $\mathcal{L}_k$ to be the set of first-order formulas φ, such that the variables in φ are a subset of $x_1, x_2, \ldots, x_k$. To see the full power of $\mathcal{L}_k$, one has to reuse the same variable many times for different purposes in the same formula — a practice that is not very common in everyday mathematics.

For example, consider the following sentence in $\mathcal{L}_2$.

$$\psi \quad\equiv\quad \forall x_1 \exists x_2 \Big(E(x_1, x_2) \land \exists x_1 \big(\neg E(x_1, x_2) \big) \Big)$$

The sentence, ψ, says that every vertex is adjacent to some vertex which is itself not adjacent to every vertex. Note that the first quantifier ($\forall x_1$) refers only to the free occurrence of x_1 within its scope.

The language $\mathcal{L}_k$ is weak in expressing quantitative properties. For example, it is impossible to say that there are k vertices of degree k. On the other hand, it is possible to say that three are $k-3$ vertices of degree 2, even though it has to be formulated somewhat cumbersome.

The language $\mathcal{C}_k$ is a natural extension of $\mathcal{L}_k$, enabling such statements or making them more elegant. For every positive integer i, $\mathcal{C}_k$ allows a quantifier $(\exists i\, x)$ with a straightforward meaning. For example, $(\exists 3\, x)\varphi(x)$ means that there are at least 3 distinct vertices with property φ.

As an example, the following formula in $\mathcal{C}_2$ says that x_i is adjacent to at least two vertices of degree 7.

$$(\exists 2\, x_2)(E(x_1, x_2) \land (\exists 7\, x_1) E(x_1, x_2))$$

2.2 Pebbling Games

Let G and H be two graphs, and let m and k be natural numbers. Define the m-move $\mathcal{L}_k$ *game* on G and H as follows. There are two players, and for each variable x_i ($i = 1, \ldots, k$), there is a pair of pebbles labeled x_i. Initially, the pebbles lie outside the game board containing the graph.

In each move, Player I starts by selecting an $i \in \{1, \ldots, k\}$ and picking up the pair of x_i pebbles. Then he places one of them on a vertex in one of the graphs. Player I is free to select pebbles that have or have not already been placed on the board. Player II must then place the other x_i pebble on a vertex of the other graph.

To define win or loss, consider the subgraphs G' and H' of G and H induced by the pebbled vertices. The pebble respecting mapping f (if it exists) assigns the vertex of G' pebbled by x_i to the vertex of H' pebbled by x_i. Player II loses, if after some move, f does not exist or is not an isomorphism of G' and H'. Player I loses, if Player II plays m moves without losing. Player II has a winning strategy for the $\mathcal{L}_k$ game (without restriction) on the number of moves) if she can play indefinitely without losing against any strategy of Player I.

Some authors call Player II the duplicator, because she wants the two graphs to look the same. They call Player I the spoiler, as he tries to interfere with this goal.

Theorem 1. *[9] Player II has a winning strategy for the $\mathcal{L}_k$ game on G, H iff $G \equiv_{\mathcal{L}_k} H$.*

A modification of the $\mathcal{L}_k$ games provides a combinatorial tool for analyzing the expressive power of $\mathcal{C}_k$. The game board looks the same, and inning is defined as for $\mathcal{L}_k$. Just as in the $\mathcal{L}_k$ game, the two players use k pairs of pebbles. The difference is that each move now has two parts.

- Player I picks up the x_i pebble pair for some i and selects a set A of vertices from one of the graphs. Player II answers with a set B of vertices from the other graph such that $|B| = |A|$.
- Player I places one of the x_i pebbles on some vertex $v \in B$. Player II answers by placing the other x_i pebble on some $u \in A$.

We interpret the first part of a move as an assertion of Player I that there exist $|A|$ vertices in G with a certain property. Player II answers with the same number of such vertices in H. Player I challenges one of the vertices in B and Player II replies with an equivalent vertex from A. Note that it is never an advantage for Player I to include vertices with obviously different properties in A. Again, games and logic are just two sides of the same coin.

Theorem 2. *[8] Player II has a winning strategy for the $\mathcal{C}_k$ game on G, H if and only if $G \equiv_{\mathcal{C}_k} H$.*

2.3 Weisfeiler-Lehman Refinement

One-dimensional Weisfeiler-Lehman refinement (1-dim W-L) is just vertex classification, first by the degree and then by the multi-set of colors of the neighbors, until no color class is split anymore.

For $k > 1$, k-dim W-L is defined as follows. Let G be a graph and let $u = (u_1, \ldots, u_k)$ be a k-tuple of vertices of G. The initial color $W^0(u)$ is defined according to the isomorphism type of u. That is, $W^0(u) = W^0(v)$ iff

$$\forall i \, \forall j \, ((u_i, u_j) \in E \Longleftrightarrow (v_i, v_j) \in E)$$

For each vertex w, we define

$$\begin{aligned}\operatorname{sift}_t(u, w) = \langle & W^t(w, u_2, u_3, \ldots, u_{k-1}, u_k), W^t(u_1, w, u_3, \ldots, u_{k-1}, u_k), \ldots \\ & \ldots, W^t(u_1, u_2, u_3, \ldots, w, u_k), W^t(u_1, u_2, u_3, \ldots, u_{k-1}, w)\rangle\end{aligned}$$

Thus $\operatorname{sift}_t(u, v)$ is the k-tuple of W^t-colors of the k-tuples of vertices obtained by substituting vertex w in turn for each of the k occurrences of a vertex in the k-tuple u.

At time $t+1$, the new colors $W^{t+1}(u)$ and $W^{t+1}(v)$ are the same, if $W^t(u) = W^t(v)$ and the number of w's for which $\operatorname{sift}_t(u, w)$ has any specific value is the same as the number of w's for which $\operatorname{sift}_t(v, w)$ has that same value.

Finally $\overline{W}(u)$ is the stable color of u. It is obtained after at most n^k iterations, i.e., $\overline{W}(u) = W^{n^k}(u)$.

Building on previous work [9,8] the following result has shown the close connection between logic, games, and Weisfeiler-Lehman refinement. Here, the formulas are allowed to have free variables, which are interpreted by the k-tuples u and v respectively.

Theorem 3. *[2] Let G, H be a pair of colored graphs and let (u, v) be a k-configuration on G, H, where $k \geq 1$. Then the following are equivalent:*

1. *$W^m(u) = W^m(v)$ for k-dim W-L*
2. *$G, u \equiv_{\mathcal{C}_{k+1,m}} H, v$*
3. *Player II has a winning strategy for the m-move $\mathcal{C}_{k+1}$ game on (G, H), whose initial configuration is (u, v).*

3 An Example Where k-Dim W-L Is Slow

Our construction of counter-examples starts with a graph G_n^h (see Figure 1), which we call the global graph. We modify G_n^h to obtain 2 graphs $X(G_n^h)$ and $\tilde{X}(G_n^h)$ ("X twist of G_n^h") which are difficult to distinguish by k-dim W-L. For the purpose of forcing k to be big, the global graph has been chosen as an expander [2]. For this paper, we choose the pretty simple grid graph G_n^h.

Now we describe how to modify G_n^h to obtain $X(G_n^h)$. Every vertex of degree d of G_n^h is replaced by 2^{d-1} vertices, which we want to view as the four corners

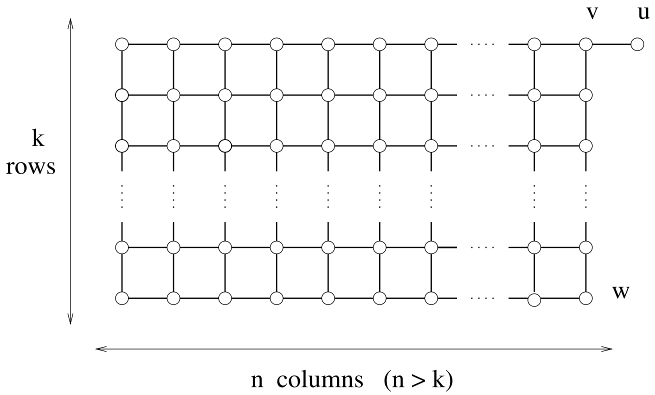

Fig. 1. The global graph G_n^h with $kn + 1$ vertices, where $k \geq 1$ is a constant

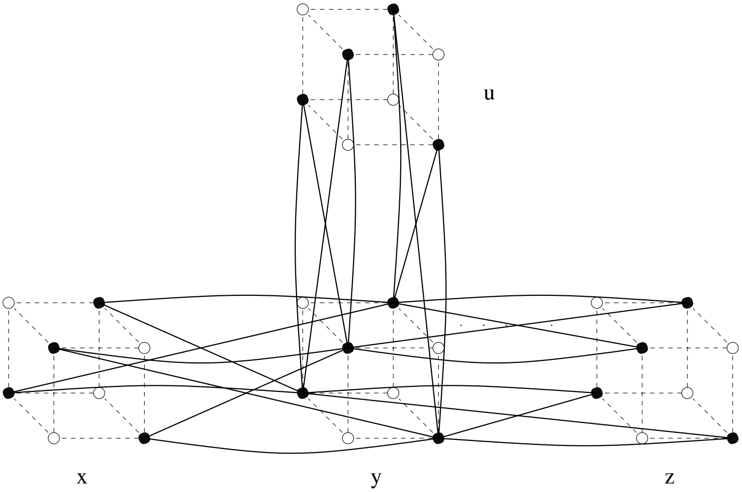

Fig. 2. This figure shows a meta-vertex y and its 3 neighbors u, x, and z. All 4 meta-vertices correspond to vertices of degree 3 in G_n^h. They are therefore represented by 3-dimensional half-cubes. For each meta-vertex, only the 4 dark points are vertices. The 4 white points and the dashed lines are just there to illustrate the cubes. Note the 3 different types of connections of y to its neighbors. The connections to u are left to left and right to right. The connections to x are front to front and back to back. The connections to z are top to top and bottom to bottom. A top to left and bottom to right connection would also be fine, as long as every vertex of degree 3 is represented by a meta-vertex whose connections represent the 3 basic partitions: left-right, top-bottom, and front-back.

of a d-dimensional cube with an even number of coordinates being 1. We refer to these vertices as a half-cube or meta-vertex (see Figure 2).

We might denote the vertices of the half-cube at a vertex v of G_n^h by $v(0,0,0)$, $v(0,1,1), v(1,0,1), v(1,1,0)$. If two vertices u and v of G_n^h are adjacent, then their half-cubes are connected as follows. Say, u is of degree 4 and and $\{u,v\}$ is the third edge of u, and v is of degree 3 and $\{u,v\}$ is the first edge of v. Then, for all $i_1, i_2, i_4, j_2, j_3, \ell \in \{0,1\}$, the vertex $u(i_1, i_2, \ell, i_4)$ is adjacent to $v(\ell, j_2, j_3)$ (provided these vertices exist, i.e., the sum of their coordinates is even).

$\tilde{X}(G_n^h)$ is constructed almost exactly as $X(G_n^h)$ with one exception. We say one edge of G_n^h is twisted.

Definition 2. *To* twist an edge $\{u,v\}$ *of the global graph* G_n^h *means to replace every edge between the meta-vertex* u *and the meta-vertex* v *by a non-edge, and every non-edge by an edge.*

It is not difficult to see that $\tilde{X}(G_n^h)$ and $X(G_n^h)$ are not isomorphic. We cannot make the twist disappear, but we can move it around to any edge of the connected global graph G_n^h. For example, mapping $u(i_1, i_2, i_3, i_4)$ to $u(1-i_1, i_2, i_3, 1-i_4)$ moves a twist from a the first edge of u to its fourth edge.

4 The Global Game

The graphs $X(G_n^h)$ and $\tilde{X}(G_n^h)$ are nicely structured. Nevertheless it is somewhat complicated to analyze the games played on them. Therefore, we investigate a simpler game $\mathcal{G}_k$ that can still adequately describe the original game $\mathcal{C}_k$. The new game is played on the global graph G_n^h rather than the pair $(X(G_n^h), \tilde{X}(G_n^h))$. We therefore call it the global game.

The moves of Player I are very much the same as before. He picks up one of his k pebbles and puts it on a vertex of G_n^h. The moves of Player II are of a very different kind. To describe them, we introduce the following notion of connected components of edges in G_n^h.

Definition 3. *The edges* e, e' *are* connected *if there is a path* $v_0, v_1, \ldots, v_{\ell-1}, v_\ell$ *in* G_n^h *with* $e = (v_0, v_1)$, $e' = (v_{\ell-1}, v_\ell)$, *and none of the interior vertices* $v_1, v_2, \ldots, v_{\ell-1}$ *is holding a pebble.*

We just use the term *component* when we mean a connected component of edges.

A move of Player II just consists of declaring certain components as twisted. The game $\mathcal{G}_k$ starts with no pebbles on the board, and the only component being twisted. At any time, the number of twisted components is odd. When Player I picks up a pebble from the board, two or more components might merge into one component. The new component is twisted iff the number of merged twisted components was odd. When Player I places a pebble on the board, then one component might be replaced by two or more newly formed components. Player II declares the new components as twisted or straight, with the only restriction that the parity of the number of twisted components does not change. When a move of Player I does not split any component, then the answer of Player II consists of doing nothing.

Definition 4. *If a twisted component has size 1, then we say the twist is* trapped.

Player II loses the global game $\mathcal{G}_k$ as soon as any twist is trapped. Player II wins the m-move game, if she can do m moves without losing.

Intuitively, the original game $\mathcal{C}_k$ and the new global game $\mathcal{G}_k$ are equivalent, because of the following reasoning.

- Player I does not really have a reason to place a pebble on any node other than the origin $u(0, \ldots, 0)$ of a meta-vertex u. So we might just view him as placing the pebble on the meta-vertex (or the corresponding vertex of the global graph G_n^h).
- Unless risking inevitable defeat, Player II better place her pebble on the corresponding meta-vertex. Thus, no selection of a meta-vertex has to be done by Player II. She just selects among the vertices of the given meta-vertex. She does this by selecting twists to move her choice on u into the origin of u.
- Here, we only consider graphs G_n^h without any non-trivial automorphisms. Furthermore, every vertex can easily be identified. Therefore, the global game can be played $\mathcal{L}$-like rather than $\mathcal{C}$-like. No player makes any claims about the existence of more than one vertex with certain properties.

In summary, we are not claiming that every play of the original game $\mathcal{C}_k$ could be simulated by the global game $\mathcal{G}_k$, but we will show that it is of no significant disadvantage for a player to play in a way that can be so simulated.

Definition 5. *Player II plays* proper *in the game $\mathcal{C}_k$, if after any of her moves, it is possible to apply an odd number of twists to $\tilde{X}(G_n^h)$ such that there is a pebble respecting isomorphism between $X(G_n^h)$ and the modified graph $\tilde{X}(G_n^h)$.*

In particular, if Player II plays proper, then she answers every move by a move in the corresponding meta-vertex. Likewise, she answers any set of potential moves by a set of potential moves in corresponding meta-vertices. She is further restricted in placing a pebble within a meta-vertex, should Player I place more than one pebble on the same meta-vertex.

Our graphs G_n^h have the property that there is a unique vertex u of degree 1, distinguishing its neighbor v, and the unique vertex w of degree 2 at distance h from u. All the other vertices are characterized by their distances from v and w.

Lemma 1. *Let the number of pebbles be at least 3. If at any time, Player II does not play proper in $\mathcal{C}_k$, then Player I can force a win in $O(\log n)$ additional moves.*

Proof. The unique characterization of the vertices in G_n^h implies that some distance is wrong, whenever Player II selects a non-matching meta-vertex. With 3 pebbles, Player I can easily exhibit the shorter distance in $O(\log n)$ moves by a divide-and-conquer approach. Hereby, Player I might have a need to identify the vertices u or w of G_n^h. As these vertices are (partly) characterized by their degrees, Player I will use the full power of $\mathcal{C}_k$-moves as follows. When Player II matches a low degree vertex by a high degree vertex, then Player I proposes the set of neighbors of the high degree vertex, and Player II has no appropriate answer.

Assume now that Player II has always played in the correct meta-vertices, but no set of twists can produce a pebble respecting isomorphism. Then it is not hard to see that there has to be an inconsistency within a meta-vertex containing multiple pebbles. E.g., $X(G_n^h)$ might have 2 pebbles in the front of the half-cube, while $\tilde{X}(G_n^h)$ has one of the corresponding pebbles in the front and one in the back. By selecting in that neighboring meta-vertex which distinguishes front from back, Player I wins in one move.

□

As it does not pay off for Player II to play improper, we can focus now on the case where Player II always plays proper.

Theorem 4. *Assume Player II is restricted to play proper in the game $\mathcal{C}_k$ on the pair $(X(G_n^h), \tilde{X}(G_n^h))$. Then a player has a strategy to win the m-move $\mathcal{C}_k$ game on the pair $(X(G_n^h), \tilde{X}(G_n^h))$ if and only if that player has a strategy to win the m-move global $\mathcal{G}_k$-game.*

Proof. We have to prove four parts.

(a) Player I wins the m-move $\mathcal{C}_k$ game on the pair $(X(G_n^h), \tilde{X}(G_n^h))$. In the simulating global game $\mathcal{G}_k$, Player I has only to be specific about the selection of meta-vertices, but not about his choice within any meta-vertex, while Player II still shows her complete selection. Thus Player I can follow his old winning strategy. When Player I wins the simulated game, some pair of pebble is adjacent in one copy, but not adjacent in the other one. These two pairs correspond to a trapped twist in G_n^h, indicating a win in the simulating game too.

(b) Player I wins the m-move $\mathcal{G}_k$ game on the pair $(X(G_n^h), \tilde{X}(G_n^h))$. In the simulating game $\mathcal{C}_k$, Player I has to make choices within meta-vertices. He always chooses the origin. A trapped twist in the global game $\mathcal{G}_k$ corresponds to an edge vs. non-edge pair in the simulating game implying a win too.

(c) Player II wins the m-move $\mathcal{C}_k$ game on the pair $(X(G_n^h), \tilde{X}(G_n^h))$. As Player II is restricted to proper plays, there is always a placement of twists onto the edges such that her moves are exactly matching the moves of Player I. The placements of twists on edges determine a unique parity of twists in each component, producing the simulating move of Player II. The simulated move produces no conflict if the simulated move did not.

(d) Player II wins the m-move $\mathcal{G}_k$ game on the pair $(X(G_n^h), \tilde{X}(G_n^h))$. The moves of Player II in the $\mathcal{G}_k$-game really describe her strategy to reply to any move of Player I on the same meta-vertex. Player II just follows this strategy. □

5 Upper and Lower Bounds

Theorem 5. *The number of moves sufficient for Player I to win the game* $\mathcal{C}_k$ *varies as follows depending on the number of pebbles.*

(a) *Player I has a winning strategy in the* $\mathcal{C}_{3h}$ *game on the pair* $(X(G_n^h), \tilde{X}(G_n^h))$ *in* $O(\log n)$ *moves.*
(b) *Player I has a winning strategy in the* $\mathcal{C}_{2h+1}$ *game on the pair* $(X(G_n^h), \tilde{X}(G_n^h))$ *in* $O(\sqrt{n})$ *moves.*
(c) *Player I has a winning strategy in the* $\mathcal{C}_{h+1}$ *game on the pair* $(X(G_n^h), \tilde{X}(G_n^h))$ *in* $O(n)$ *moves.*

Proof. It is sufficient to consider the corresponding $\mathcal{G}_k$ game. We say that Player I builds a *wall* if he places pebbles on all vertices of a cut disconnection the leftmost column form the rightmost (full) column. For example the vertices of one column of G_n^h form a wall.

(a) Having enough pebbles to produce 3 walls in G_n^h, Player I can employ a divide-and-conquer strategy. The pebbles of one wall have only to be removed when the twist is captured between the other 2 walls.
(b) Player I builds a new wall at distance $\sqrt{n}$ from the previous wall starting in the middle and moving towards the twist. As soon as a wall is built that keeps the twist away from the other one, the old wall is no longer needed and its pebbles can be reused. If the twist is located between two walls, then Player I moves one of them slowly inside using the additional pebble.
(c) Player I builds a wall anywhere (best in the middle). Then move it slowly towards the side containing the twist. □

Note that Player I can win the $\mathcal{G}_k$ game on G_n^h by a particularly simple winning strategy. He can build a wall on the left hand side and move it towards the right hand side, step by step decreasing the size of the component containing the twist. All moves of Player I are independent of the moves of Player II.

Theorem 6. *For* $k \leq h$*, Player II has a winning strategy in the* $\mathcal{C}_k$ *game on the pair of graphs* $(X(G_n^h), \tilde{X}(G_n^h))$*.*

Proof. We may look at the corresponding $\mathcal{G}_k$ game. Even for $k = h$, Player I has just enough pebbles to build a wall, but in the next move he has to break it down again. Player II easily maintains a single twist, always in the largest component. □

Corollary 1. $h - 1$*-dim W-L cannot detect a difference between the graphs* $X(G_n^h)$ *and* $\tilde{X}(G_n^h)$*.*

Corollary 2. $X(G_n^h)$ *and* $\tilde{X}(G_n^h)$ *agree on all formulas of* $\mathcal{C}_h$*.*

Definition 6. *The* size *of a component in* G_n^h *is the number of empty columns in it. A component is* good *if its size is positive.*

Theorem 7. *For $k \leq 2h$, every winning strategy of Player I in the $\mathcal{G}_k$ game on G_n^h requires at least $\Omega(n)$ moves.*

Proof. Let us start with the trivial observation that in G_n^h there are h vertex-disjoint paths between any pair of distinct good components. Thus there is a wall consisting of at least h pebbled vertices between these components. Thus with at most $2h$ pebbles, there are at any time at most 3 components.

We now want to describe a strategy for Player II, that sufficiently delays a win of Player I. In this strategy, Player II always maintains just a single twist. Assume that one good component C_1 of size s_1 exists, and another good component containing the twist is just split into into good component C_2, C_3 with sizes s_2 and s_3 respectively. Let C_2 be the component between C_1 and C_3. Then Player II puts the twist into C_3 if $s_1 + s_2 \leq s_3$, and otherwise into C_2. The following removal of any pebble by Player I breaks a wall, again producing 2 components with the twist being in a component of size at least s_3.

When two good components are formed after m' moves, the twist is in the larger component of size at least $(n - m')/2$. After m moves, the twist is usually in a component of size at least $(n-m')/2-(m-m') = (n+m')/2-m > n/2-m$. There is an exception for the isolated times, when 3 components exist, in which case $n/2 - m$ is a lower bound on the sum of the sizes of the middle and any outer component. Player II does not lose before the twist is in a bad component (of size 0). Thus the number of moves is at least $n/2 = \Omega(n)$. □

Corollary 3. *For $k \leq 2h$, every winning strategy of Player I in the $\mathcal{C}_k$ game on the pair $(X(G_n^h), \tilde{X}(G_n^h))$ requires at least $\Omega(n)$ moves.* □

Theorem 8. *For $k \leq 2h+1$, every winning strategy of Player I in the $\mathcal{G}_k$ game on G_n^h requires at least $\Omega(\sqrt{n})$ moves.*

Proof. As in the proof of Theorem 7, there are at most 3 good components at any time. When 2 good components are formed for the first time, a good strategy for Player II is to move the twist into the larger one. When 3 good component C_1, C_2, C_3 (with s_i = size of C_i) are formed, she has a choice between say C_2 and C_3 where C_2 is between C_1 and C_3. She chooses C_2 if $s_2 > \sqrt{n}$ and $s_3 < n/2 - k$. (This selection could be slightly better optimized without improving the Theorem.) Consider the integer r defined by

$$r = \min(s_1 + s_2, s_3 + s_2, s_2\sqrt{n})$$

if there are 3 good components, and the twist is in C_2. If there are less than 3 good components, then r is defined to be the size of the larger or only good component. When two components are formed form one, then r gets a value of at least $(n - k)/2$. Once the value of r is less than $n/2 - k$, it can never decrease by more than $\sqrt{n}$ in a single move. This can be shown by case analysis, where the only interesting case is going form 2 good components to 3. The $\Omega(n)$ lower bound follows immediately. □

Corollary 4. *For $k \leq 2h+1$, every winning strategy of Player I in the $\mathcal{C}_k$ game on the pair $(X(G_n^h), \tilde{X}(G_n^h))$ requires at least $\Omega(\sqrt{n})$ moves.* □

A recent result of Grohe [6] says that determining whether two graphs are $\mathcal{C}_{k+1}$ equivalent, and thus whether they can be distinguished by k-dimensional Weisfeiler-Lehman refinement, is P-complete. Grohe shows the same result for $\mathcal{L}_{k+1}$ equivalence too. This does not imply, but certainly strongly suggests that k-dimensional Weisfeiler-Lehman refinement is slow. Indeed the method of Grohe could also be used to prove Theorem 7. It seems that such a proof would be much more complicated than the proof given in this paper.

Acknowledgment. I want to thank Luitpold Babel for an email conversation in 1994 on some results that implicitly assumed associativity of the multiplication in coherent algebras. This has caused me to discover the main result of this paper.

References

1. L. Babai and L. Kučera, *Graph canonization in linear average time*, 20th Annual Symposium on Foundations of Computer Science (Long Beach, Ca., USA), IEEE Computer Society Press, October 1979, pp. 39–46.
2. Jin-Yi Cai, Martin Fürer, and Neil Immerman, *An optimal lower bound on the number of variables for graph identification*, Combinatorica **12** (1992), no. 4, 389–410.
3. A. Ehrenfeucht, *An application of games to the completeness problem for formalized theories*, Fund. Math. **49** (1960/1961), 129–141.
4. I. A. Faradžev, M. H. Klin, and M. E. Muzichuk, *Cellular rings and groups of automorphisms of graphs*, Investigations in algebraic theory of combinatorial objects, Kluwer Acad. Publ., Dordrecht, 1994, pp. 1–152.
5. Roland Fraïssé, *Sur quelques classifications des systèmes de relations*, Publ. Sci. Univ. Alger. Sér. A. **1** (1954), 35–182 (1955).
6. Martin Grohe, *Equivalence in finite-variable logics is complete for polynomial time*, Combinatorica **19** (1999), no. 4, 507–532.
7. D. G. Higman, *Coherent configurations. I. Ordinary representation theory*, Geometriae Dedicata **4** (1975), no. 1, 1–32.
8. N. Immerman and E. S. Lander, *Describing graphs: A first-order approach to graph canonization*, Alan L. Selman, Editor, Complexity Theory Retrospective, In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988, Springer-Verlag, 1990, pp. 59–81.
9. Neil Immerman, *Upper and lower bounds for first order expressibility*, Journal of Computer and System Sciences **25** (1982), no. 1, 76–98.
10. L. Kučera, *Canonical labeling of regular graphs in linear average time*, Proceedings of the 28th Annual Symposium on Foundations of Computer Science (Los Angeles, CA) (Ashok K. Chandra, ed.), IEEE Computer Society Press, October 1987, pp. 271–279.
11. Boris Weisfeiler (ed.), *On construction and identification of graphs*, Springer-Verlag, Berlin, 1976, With contributions by A. Lehman, G. M. Adelson-Velsky, V. Arlazarov, I. Faragev, A. Uskov, I. Zuev, M. Rosenfeld and B. Weisfeiler, Lecture Notes in Mathematics, Vol. 558.