Count-Free Weisfeiler-Leman and Group Isomorphism

WACT 2023

1,2. **Nathaniel A. Collins** 2. Michael Levet

March 27, 2023

1. Department of Mathematics
2. Department of Computer Science

Acknowledgments: Josh Grochow

Motivation

Key motivation for Group Isomorphism

Problem: Graph Isomorphism (GI)

Given two finite graphs G, H, does there exist an isomorphism $\varphi : V(G) \to V(H)$?

A brief tour of the complexity of GI

- Upper bound is $\mathsf{NP} \cap \mathsf{coAM}$
- Lower bound is DET ($\mathsf{NL} \subseteq \mathsf{DET} \subseteq \mathsf{TC}^1$)
- Possible candidate to be NP-intermediate (In NP but not P nor NP-complete).
- Most efficient general algorithm is $n^{\Theta(\log^2 n)}$ (Babai 2016)

A brief tour of the complexity of GI

- Upper bound is $\mathsf{NP} \cap \mathsf{coAM}$
- Lower bound is DET ($\mathsf{NL} \subseteq \mathsf{DET} \subseteq \mathsf{TC}^1$)
- Possible candidate to be NP-intermediate (In NP but not P nor NP-complete).
- Most efficient general algorithm is $n^{\Theta(\log^2 n)}$ (Babai 2016)
- How to improve?

A brief tour of the complexity of GI

- Upper bound is $\mathsf{NP} \cap \mathsf{coAM}$
- Lower bound is DET ($\mathsf{NL} \subseteq \mathsf{DET} \subseteq \mathsf{TC}^1$)
- Possible candidate to be NP-intermediate (In NP but not P nor NP-complete).
- Most efficient general algorithm is $n^{\Theta(\log^2 n)}$ (Babai 2016)
- How to improve?
  - Good special cases?

Background: Group Isomorphism

Problem: Group Isomorphism (GpI)

Given two finite groups G, H by their Cayley (multiplication) tables, does there exist an isomorphism $\varphi : G \to H$?

A brief tour of the complexity of GpI

- GpI is strictly easier than GI under AC^0 reductions (CTW 2013)
- Best general upper bound is $n^{\Theta(\log n)}$ ["1970s"]
    - Compare with $n^{\Theta(\log^2 n)}$ upper bound on GI
- Lots of work in the last 15 years on polynomial time algorithms for special cases

A brief tour of the complexity of GpI

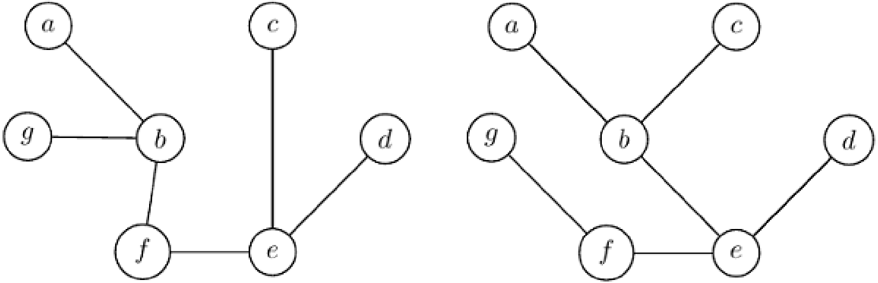
- GpI is strictly easier than GI under AC^0 reductions (CTW 2013)
- Best general upper bound is $n^{\Theta(\log n)}$ ["1970s"]
    - Compare with $n^{\Theta(\log^2 n)}$ upper bound on GI
- Lots of work in the last 15 years on poly-time algorithms for special cases
- These algorithms primarily leverage algebraic techniques.

A brief tour of the complexity of GpI

- GPI is strictly easier than GI under AC^0 reductions (CTW 2013)
- Best general upper bound is $n^{\Theta(\log n)}$ ["1970s"]
  - Compare with $n^{\Theta(\log^2 n)}$ upper bound on GI
- Lots of work in the last 15 years on poly-time algorithms for special cases
- These algorithms primarily leverage algebraic techniques.
- Since GPI is strictly easier than GI, can we fruitfully adapt successful combinatorial techniques from GI?

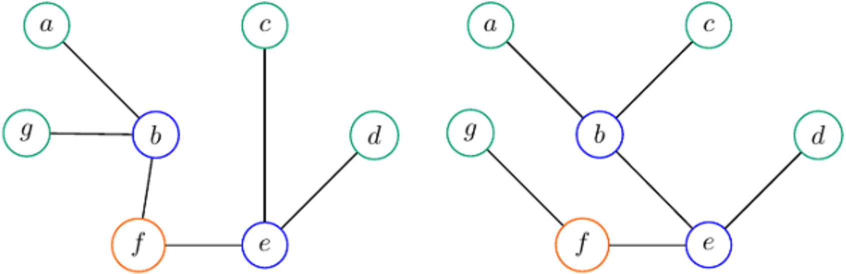
Weisfeiler-Leman (WL)

Example: 1-dim WL



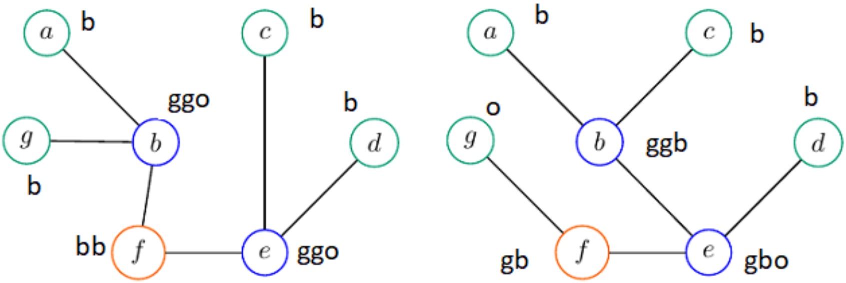
Example: 1-dim WL

- Vertices are initially colored by their degree.



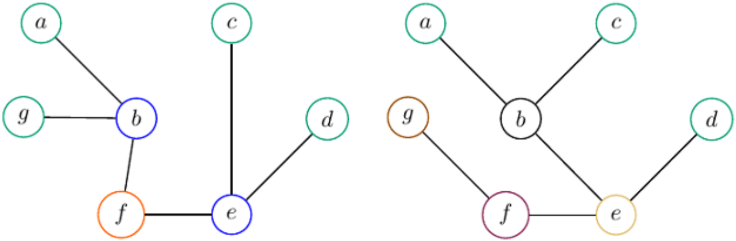
Example: 1-dim WL

- Vertices are then colored by the multiset of neighbors colors



Example: 1-dim WL

- Vertices are colored by the multiset of neighbors colors



1-dimensional Weisfeiler-Leman

- Colors vertices in an isomorphism-invariant manner.
- The r-round, 1-dimensional Weisfeiler-Leman algorithm consists of two parts:

1-dimensional Weisfeiler-Leman

- Colors vertices in an isomorphism-invariant manner.
- The r-round, 1-dimensional Weisfeiler-Leman algorithm consists of two parts:
    - An initial coloring where vertices are colored by their degree.

1-dimensional Weisfeiler-Leman

- Colors vertices in an isomorphism-invariant manner.
- The r-round, 1-dimensional Weisfeiler-Leman algorithm consists of two parts:
    - An initial coloring where vertices are colored by their degree.
    - An iterated color refinement step where each vertex is colored by the multiset of itself and it's neighbors colors.

1-dimensional Weisfeiler-Leman

- Colors vertices in an isomorphism-invariant manner.
- The r-round, 1-dimensional Weisfeiler-Leman algorithm consists of two parts:
    - An initial coloring where vertices are colored by their degree.
    - An iterated color refinement step where each vertex is colored by the multiset of its own and its neighbor's colors.
- Terminate after r rounds or after the multisets differ.

k-dimensional Weisfeiler-Leman

- For $k \geq 2$, k-WL colors k-tuples of vertices instead of single vertices.

k-dimensional Weisfeiler-Leman

- For $k \geq 2$, k-WL colors k-tuples of vertices instead of single vertices.
- For a fixed k, WL runs in polynomial time.
- For a fixed dimension k, each round of k-WL is TC^0-computable.

Counterexample

- 1-WL fails on regular graphs

Counterexample

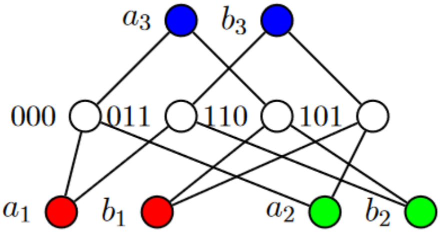
- 1-WL fails on regular graphs
- 2-WL fails on strongly regular graphs

Counterexample

- 1-WL fails on regular graphs
- 2-WL fails on strongly regular graphs
- Counterexample to higher WL?

General counterexample

- Replace each vertex in G and H with a gadget like the following and connect the gadgets in a specific way.
- Then WL fails to distinguish these graphs in polynomial time.



CFI Gadget *(Cai, Fürer, Immerman) 1992.*

WL for Group Isomorphism

Background: WL for Group Isomorphism

- Brachter and Schweitzer (2020) introduced three versions of WL

Background: WL for Group Isomorphism

- Brachter and Schweitzer (2020) introduced three versions of WL
- Versions I and II color k-tuples of group elements.

Background: WL for Group Isomorphism

- Brachter and Schweitzer introduced three versions of WL (BS 2020)
- Versions I and II color k-tuples of group elements.
  - Initial Coloring: $(g_1, \ldots, g_k)$ and $(h_1, \ldots, h_k)$ receive the same initial color iff:
    - Version I: Whenever $g_i = g_j$ then $h_i = h_j$ and whenever $g_i g_j = g_m$ then $h_i h_j = h_m$
    - Version II: The map $g_i \mapsto h_i, \forall i$ extends to an isomorphism $\langle g_1, \cdots, g_k \rangle \cong \langle h_1, \cdots, h_k \rangle$.

Background: WL for Group Isomorphism

- Brachter and Schweitzer introduced three versions of WL (BS 2020)
- Versions I and II color k-tuples of group elements.
  - Initial Coloring: $(g_1, \ldots, g_k)$ and $(h_1, \ldots, h_k)$ receive the same initial color iff:
    - Version I: Whenever $g_i = g_j$ then $h_i = h_j$ and whenever $g_i g_j = g_m$ then $h_i h_j = h_m$
    - Version II: The map $g_i \mapsto h_i, \forall i$ extends to an isomorphism $\langle g_1, \cdots, g_k \rangle \cong \langle h_1, \cdots, h_k \rangle$.
  - The refinement step is performed in the same manner as for graphs.

Background: WL for Group Isomorphism

- Brachter and Schweitzer introduced three versions of WL (BS 2020)
- Versions I and II color k-tuples of group elements.
  - Initial Coloring: $(g_1, \ldots, g_k)$ and $(h_1, \ldots, h_k)$ receive the same initial color iff:
    - Version I: Whenever $g_i = g_j$ then $h_i = h_j$ and whenever $g_i g_j = g_m$ then $h_i h_j = h_m$
    - Version II: The map $g_i \mapsto h_i, \forall i$ extends to an isomorphism $\langle g_1, \cdots, g_k \rangle \cong \langle h_1, \cdots, h_k \rangle$.
  - The refinement step is performed in the same manner as for graphs.
- The three versions are equivalent up to a factor of 2 in the dimension.

Mekler's Construction

Brachter & Schweitzer considered class 2 p-groups arising from Mekler's construction

Mekler's Construction

Brachter & Schweitzer considered class 2 p-groups arising from Mekler's construction

The setup

Graphs vs Groups

Theorem: Brachter & Schweitzer 2020

Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then the 3-dimensional WL Version II algorithm for groups distinguishes G from H.

Corollary

Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then we can distinguish G from H using a TC^1 circuit.

Proof.

- Their proof technique uses $O(\log n)$ rounds.
- The initial coloring of WL is L computable.
- Each refinement step is TC^0 computable.

$\square$

Our Results

Our results

Theorem: C.–Levet 2022

Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then the 3-dimensional WL Version I algorithm for groups distinguishes G from H in $O(\log \log n)$ rounds.

Our results

Theorem: C.–Levet 2022

Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then the 3-dimensional WL Version I algorithm for groups distinguishes G from H in $O(\log \log n)$ rounds.

- Previous best is $O(\log n)$ rounds with Version II

Theorem: C.–Levet 2022

Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then the 3-dimensional WL Version I algorithm for groups distinguishes G from H in $O(\log \log n)$ rounds.

Proof.

- Show WL Version I suffices, we only need edge relation.
- Initial coloring is TC^0 computable
- WL requires $O(\log \log n)$ rounds.

$\square$

Our results

Corollary - Parallel Complexity:

Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then we can distinguish G from H using a TC circuit of depth $O(\log \log n)$.

Corollary - Descriptive Complexity:

Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then we obtain a more succinct formula in a weaker logic.

Count-Free Weisfeiler-Leman

Background: Count-Free WL

1. WL: Examine multiset of colors at each round
2. Each round is computable with a TC^0 circuit.

Background: Count-Free WL

1. Count-Free WL: Examine ~~multiset~~ set of colors at each round

Background: Count-Free WL

1. Count-Free WL: Examine ~~multiset~~ set of colors at each round
2. Each round is computable with an $\cancel{\mathsf{TC}^0}$ AC^0 circuit.

Background: Count-Free WL

1. Count-Free WL: Examine ~~multiset~~ set of colors at each round
2. Each round is computable with an $\cancel{\mathsf{TC}^0}$ AC^0 circuit.
3. Can we replicate our results in Count-Free WL?

Our results

Theorem: C.–Levet 2022

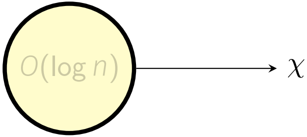
Let Γ_1, Γ_2 be CFI graphs. Let G, H be their corresponding CFI groups. Then the multiset of colors produced by the constant-dimensional Count-Free WL Version I algorithm after $O(\log \log n)$ rounds differ whenever $G \ncong H$.

Postprocessing

- Count-Free Weisfeiler-Leman can output the same multiset of colors but it lacks the power to compare the multisets.

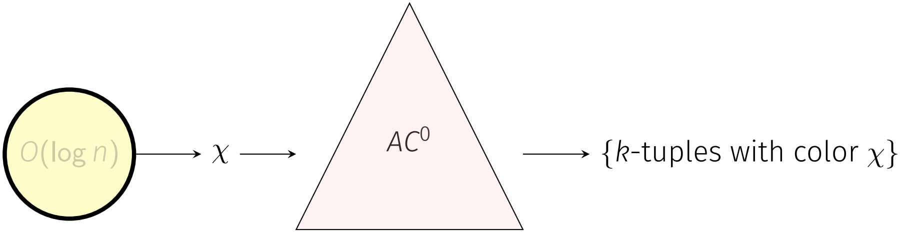
Postprocessing

1. Using $O(\log n)$ nondeterministic bits, guess a color χ where G has a higher multiplicity than H.



Postprocessing

2. Use an AC^0 circuit to find all the k-tuples with color C



Postprocessing

3. Feed the k-tuples into a single majority gate to compare the multiplicities

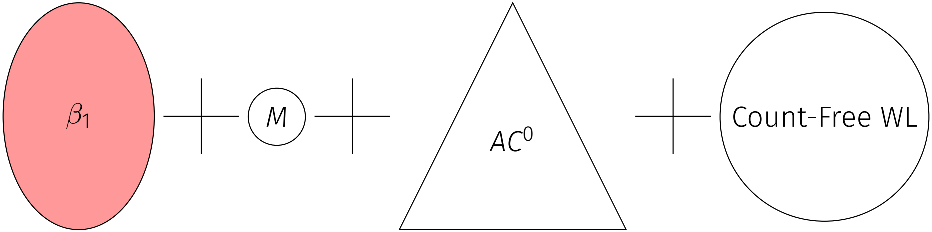
For every tuple in G with color χ	Feed 1 to the majority gate
For every tuple in H with color χ	Feed 0 to the majority gate

Postprocessing

Corollary: CFI Groups can be distinguished in $\beta_1 MAC^0(FOLL)$

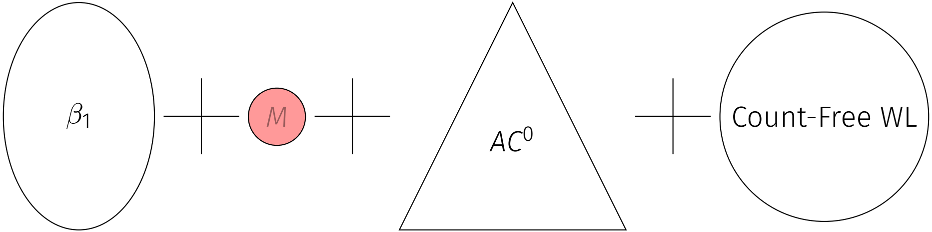
CFI Groups can be distinguished in $\beta_1 MAC^0(FOLL)$

Postprocessing



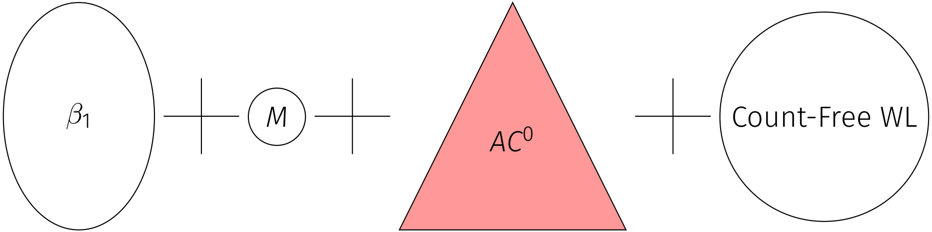
$\beta_1 MAC^0(FOLL)$

Postprocessing



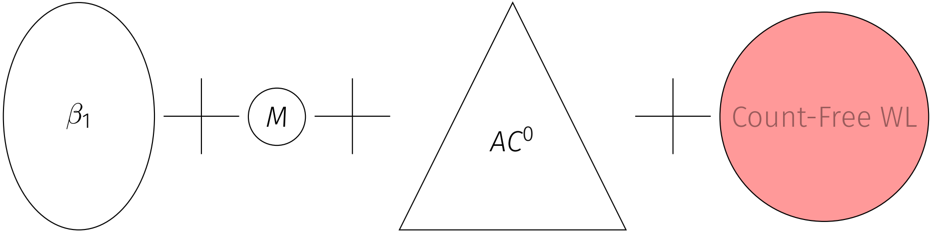
$\beta_1 MAC^0(FOLL)$

Postprocessing



$\beta_1 M AC^0 (FOLL)$

Postprocessing



$\beta_1 MAC^0(FOLL)$

What does this mean?

- Determining isomorphism of CFI groups is in $\beta_1 MAC^0(FOLL) \subseteq TC^{o(1)} \subseteq TC^1$

Open questions

- Can Count-Free WL distinguish CFI groups in $O(\log \log n)$ rounds without this postprocessing?

Open questions

- Can Count-Free WL distinguish CFI groups in $O(\log \log n)$ rounds without this postprocessing?
    - Such a result would imply that CFI groups can be distinguished in FOLL.

Open questions

- Can Count-Free WL distinguish CFI groups in $O(\log \log n)$ rounds without this postprocessing?
    - Such a result would imply that CFI groups can be distinguished in FOLL.
- Can 2-WL distinguish CFI groups?

Open questions

- Can Count-Free WL distinguish CFI groups in $O(\log \log n)$ rounds without this postprocessing?
  - Such a result would imply that CFI groups can be distinguished in FOLL.
- Can 2-WL distinguish CFI groups?
- What is the power of k-WL as a proof system for Group Isomorphism?
  - This is known for k-WL over Graph Isomorphism (BG 2015).

Thank You

Questions?