Combinatorial Properties of the Weisfeiler-Leman Algorithm

Sandra Kiefer

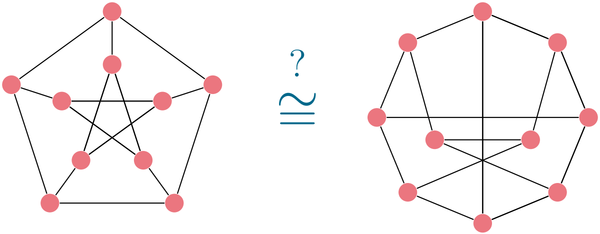
RWTH AACHEN UNIVERSITY

Workshop on {Symmetry, Logic, Computation}
Simons Institute for the Theory of Computation
November 9, 2016

Graph Isomorphism

The Graph Isomorphism Problem treats the following question:

Given two graphs G, H, decide whether $G \cong H$ or $G \ncong H$.



It can be solved in quasipolynomial time [Babai 2015], but it is not known to be in P.

One indispensable subroutine in isomorphism solvers is the Weisfeiler-Leman algorithm.

Color Refinement

k-dimensional WL algorithm iteratively computes coloring of V^k

CR = the 1-dimensional WL algorithm

1-dimensional WL algorithm

- *Initialisation*: All vertices have their initial color.
- *Refinement*: v and w get different colors $\iff$ there is a color c such that v and w have different numbers of c-colored neighbors.
- Stop when coloring is stable.

If two graphs have non-isomorphic stable colorings, then the graphs are non-isomorphic.

Outline

WL-Power
(C^k-Definability)

WL-Complexity
(C^k-Quantifier Depth)

Color Refinement

k-dimensional WL algorithm iteratively computes coloring of V^k

CR = the 1-dimensional WL algorithm

1-dimensional WL algorithm

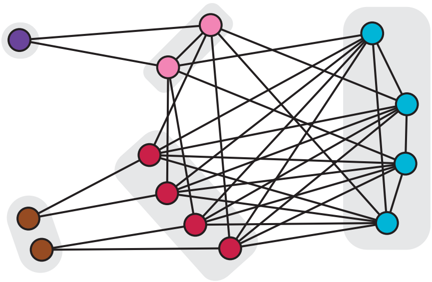
- *Initialisation*: All vertices have their initial color.
- *Refinement*: v and w get different colors $\iff$ there is a color c such that v and w have different numbers of c-colored neighbors.
- Stop when coloring is stable.

If two graphs have non-isomorphic stable colorings, then the graphs are non-isomorphic.

CR-partition = coarsest equitable partition

A partition is equitable if it is

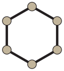
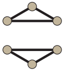
- regular in each class,
- biregular between every two classes.



Identification

If two graphs have isomorphic stable colorings w.r.t. CR, they are called CR-equivalent.

A graph G is identified by CR iff every CR-equivalent graph is isomorphic to G.

For example, the graphs  and  are CR-equivalent.

Hence, the 6-cycle is **not** identified by CR.

(But it is identified by the 2-dimensional WL algorithm.)

Characterization Theorem

Which graphs are identified by CR?

Theorem (K., Schweitzer, Selman)

*G is identified by CR $\iff$ G **flips** to a **bouquet forest**.*

In the context of logics this yields:

Corollary

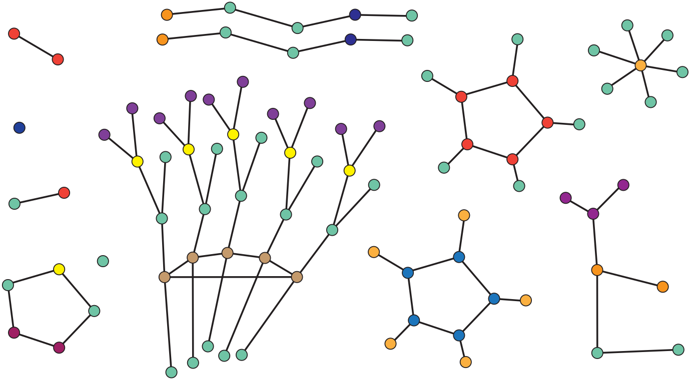
G is definable in C^2 $\iff$ G flips to a bouquet forest.

A similar result was obtained by Arvind, Köbler, Rattan, Verbitsky [AKRV '15].

Bouquet forests

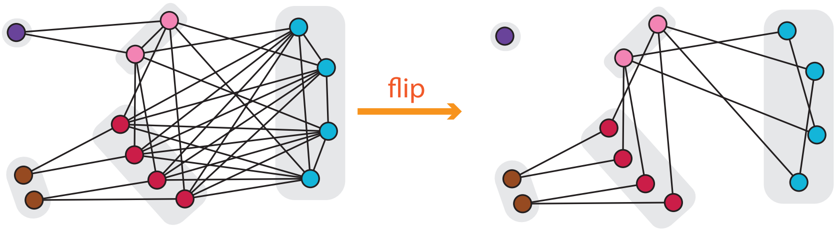
Bouquet: five copies $(T_1, v_1), \ldots, (T_5, v_5)$ of a tree (T, v), connected via a 5-cycle on $v_1, \ldots, v_5$

Bouquet forest: disjoint union of vertex-colored trees and **non-isomorphic** vertex-colored bouquets.



THE FLIP OF A GRAPH

In the CR-partition, flip between every pair of color classes.



Lemma (K., Schweitzer, Selman)

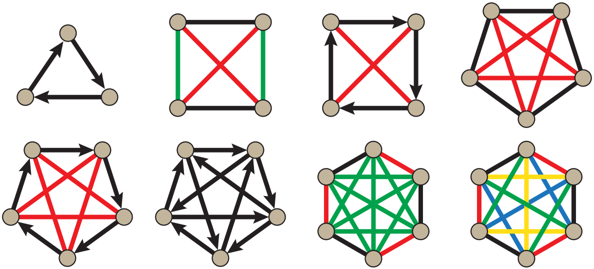
G is identified by CR $\iff$ the flip of G is identified by CR.

Characterization Theorem

Theorem (K., Schweitzer, Selman)

G is identified by CR $\iff$ G flips to a bouquet forest.

"Exceptions" in the generalization to finite relational structures:



The exceptions in higher dimensions would not look that nice – and there would be infinitely many of them. ☹

Outline

WL-Power
(C^k-Definability)

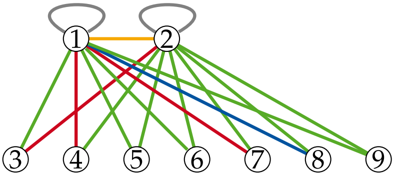
WL-Complexity
(C^k-Quantifier Depth)

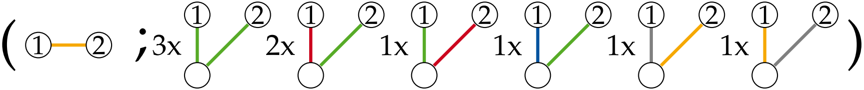
2-dimensional WL algorithm

- *Initialisation*: Arcs are colored according to edge, non-edge, loop.
- *Refinement*: (v, w) and (v', w') get different colors $\iff$ $\chi(v, w) \neq \chi(v', w')$ or

$$\{\{(\chi(u, w), \chi(v, u)) \mid u \in V\}\} \neq \{\{(\chi(u, w'), \chi(v', u)) \mid u \in V\}\}$$

- Stop when coloring is stable.





Iteration number

The stable partition can be computed in time $n^{k+1}\log(n)$.
[Cardon & Crochemore 1982; Berkholz et al. 2013]

How many iterations does the WL algorithm

	$k=1$	general k
lower bound	$n-\mathcal{O}(\sqrt{n})$ [Krebs & Verbitsky 2014]	$\Omega(n)$ [Fürer 2001]
upper bound	$n-1$	$\boldsymbol{n^k-1}$

For finite structures: $n^{\Omega(k/\log k)}$ [Berkholz & Nordström 2016]

Upper bounds for $k=2$?

Upper bounds on the iteration number

We show the following.

Theorem (K., Schweitzer)

The number of iterations of the 2-dimensional WL algorithm on graphs of size n is at most $O(n^2/\log(n))$.

In the context of logics this yields:

Corollary

In $\mathcal{C}^3$, formulas for graphs on n vertices require quantifier depth at most $O(n^2/\log(n))$.

Upper bounds on the iteration number

For graphs of bounded color class size, our proof yields linear bounds on the quantifier depth of distinguishing formulas.

Lemma

The number of iterations of the 2-dimensional WL algorithm on graphs with n vertices of color class size at most t is $O(2^t n)$.

Note:
For graphs of bounded color class size, upper and lower bound match.

Proof overview

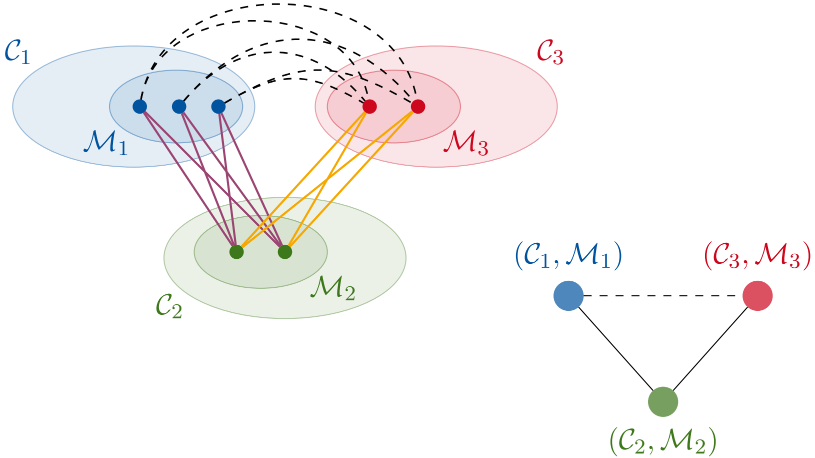
- We define a game where Maxi and Mini alternate in their turns on a graph.
- Each turn consists of choosing a refinement of the current edge coloring.
- Costs C form an upper bound on the WL iteration number.
- Both players contribute to C. Maxi aims at maximizing C, Mini aims at minimizing C.
- **Challenge**: Provide a good strategy for Mini.

Methods in the proof:

1. **Large** vertex color classes $\rightarrow$ potential function
2. **Small** vertex color classes $\rightarrow$ auxiliary graphs

Small vertex color classes

To analyze the costs related to small vertex color classes, we let Mini derive her strategy from auxiliary graphs.



Algorithm Strategy for Mini in the 2-player game on input G.

Input: A colored graph G.

Output: A graph G' satisfying $G \succeq G' \succeq \widetilde{G}$.

1: $G \leftarrow \text{cclean-up}(G)$
2: **while** $\triangle(\text{Aux}(G)) \neq \text{Aux}(G)$ **do**
3: $\quad G \leftarrow G^{(1)}$
4: $\quad G \leftarrow \text{cclean-up}(G)$
5: **end while**
6: **return** G

Lemma

The number of iterations in which color classes that are incident only to small vertex color classes are refined is $O(2^{t(n)}n)$.

Proof overview

1. **Large** vertex color classes $\rightarrow$ potential function
2. **Small** vertex color classes $\rightarrow$ auxiliary graphs

$\rightsquigarrow$ **total cost of** $O\left(2^{t(n)} \cdot n + n^2/t(n)\right)$

Theorem (K., Schweitzer)

The number of iterations of the 2-dimensional Weisfeiler-Leman algorithm on graphs of size n is at most $O(n^2/\log(n))$.

Corollary

In $\mathcal{L}^3$, formulas for graphs on n vertices require quantifier depth at most $O(n^2/\log(n))$.

Open questions

WL-Power

- Identification in dimension 1
  $\rightarrow$ both for graphs and relational structures
- Identification in higher dimensions?

WL-Complexity

- Iteration number in dimension 2
  $\rightarrow$ for graphs new lower bounds
- Iteration number in higher dimensions?
- Iteration number for finite relational structures?
  , probably