AN INTRODUCTION TO THE WEISFEILER–LEMAN ALGORITHM AND RECENT DEVELOPMENTS

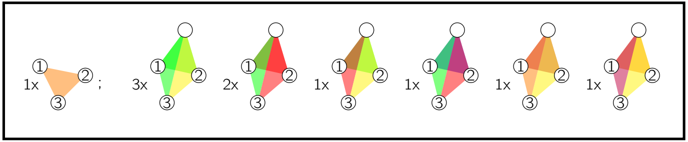
RHEIN-MAIN
UNIVERSITÄTEN
Goethe-Universität Frankfurt am Main
Johannes Gutenberg-Universität Mainz
Technische Universität Darmstadt

21.11.2025 Vienna

LOGALG 2025

Pascal Schweitzer

TECHNISCHE
UNIVERSITÄT
DARMSTADT



EngageSYMMETRY

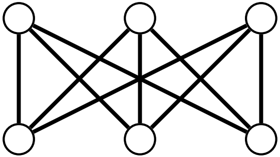
erc

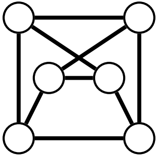
The Graph Isomorphism Problem

isomorphic graphs

Two graphs are isomorphic if there is a bijection of vertices that preserves adjacency and non-adjacency.

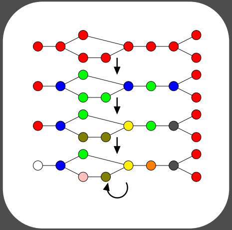
The Graph Isomorphism Problem (GI): Algorithmic task to decide whether two given graphs are isomorphic.





- One of the most important open problems in theoretical computer science
- GI is equivalent to computing graph automorphisms

1. WL Algorithm



Degrees and degrees in the neighborhood

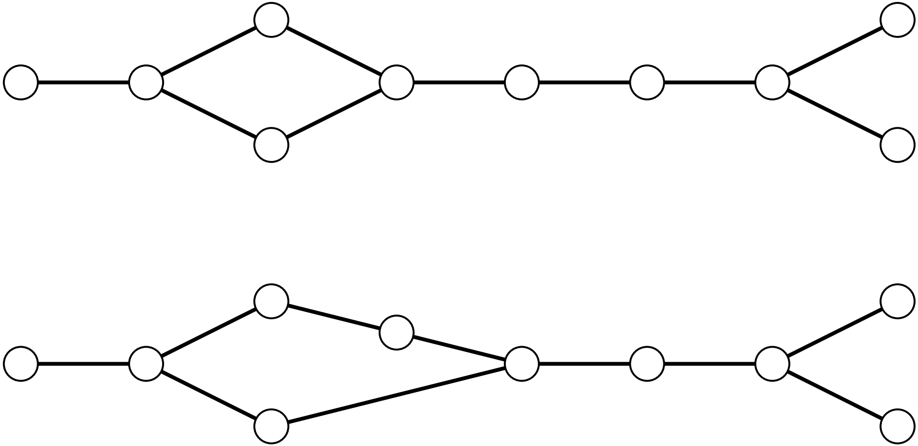
We want to explore the automorphisms of a graph combinatorially.

The 1-dimensional Weisfeiler Leman algorithm is a procedure to distinguish graphs according to combinatorial properties.

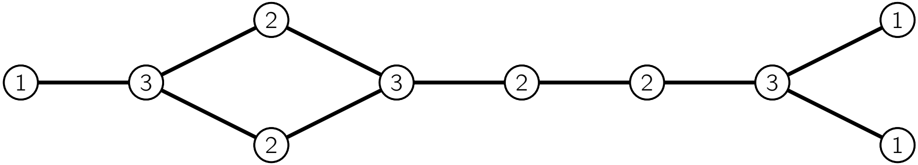
Observations:

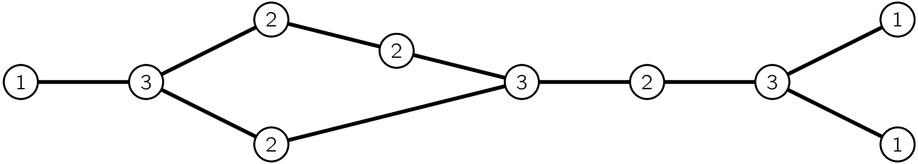
- Vertices must map to vertices of the same degree.
- The neighborhood of a vertex must map the neighborhood of its image.
- Thus, degrees that appear in the neighborhood must appear in the neighborhood of the image.
- This argument can be iterated.

Color refinement (aka. 1-WL)

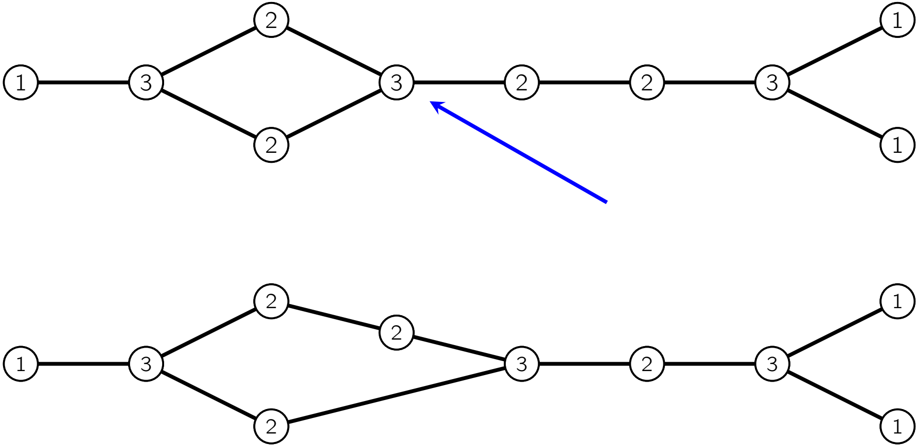


Color refinement (aka. 1-WL)

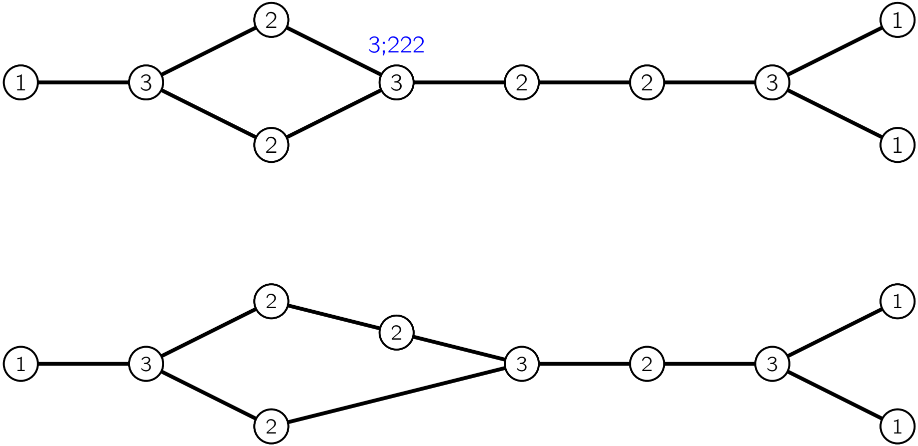




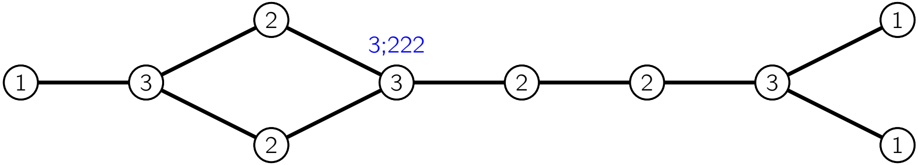
Color refinement (aka. 1-WL)

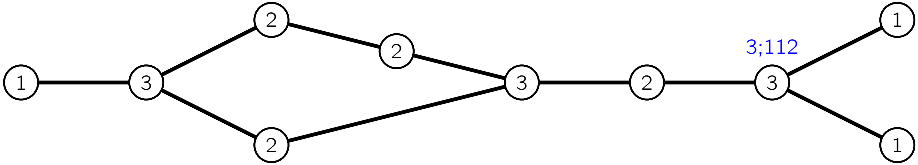


Color refinement (aka. 1-WL)

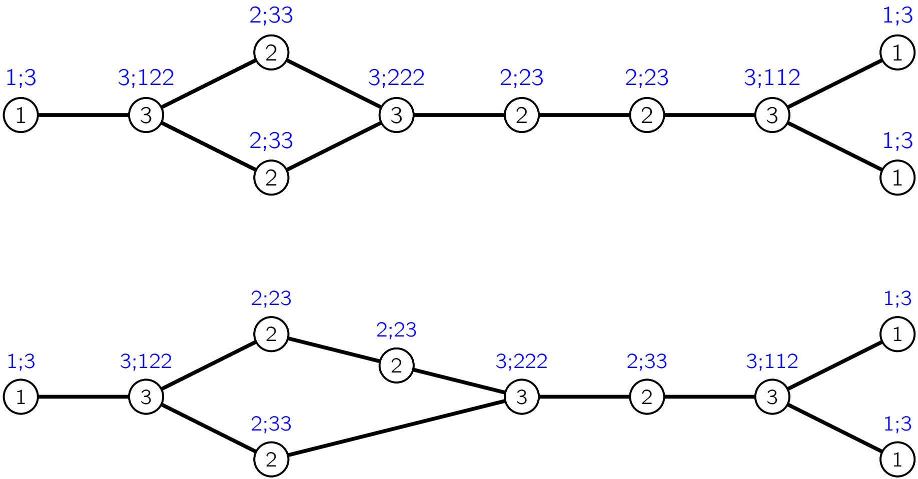


Color refinement (aka. 1-WL)

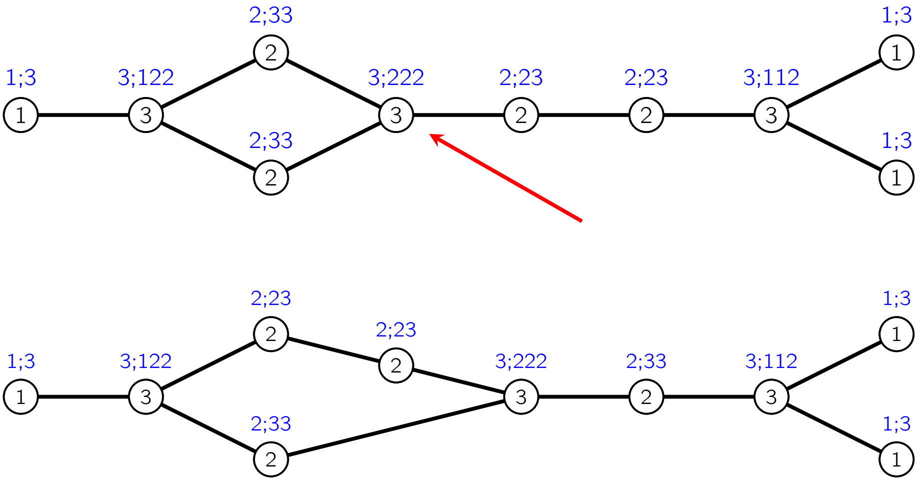




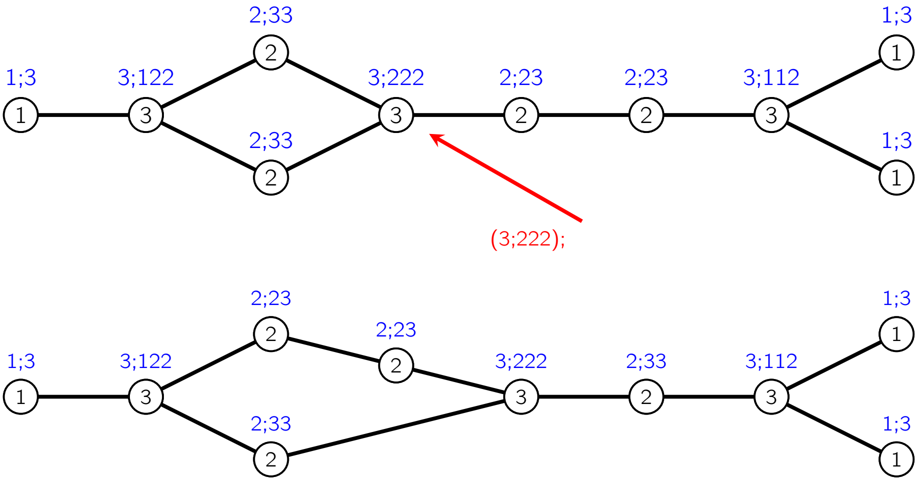
Color refinement (aka. 1-WL)



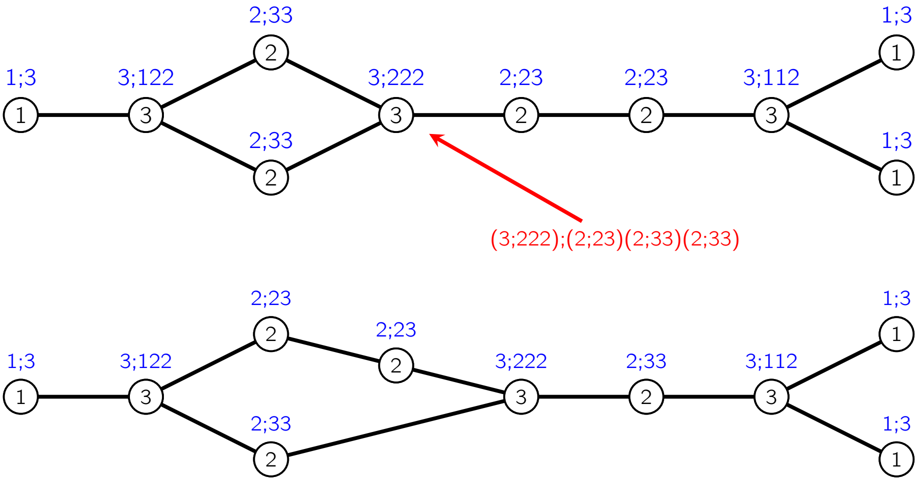
Color refinement (aka. 1-WL)



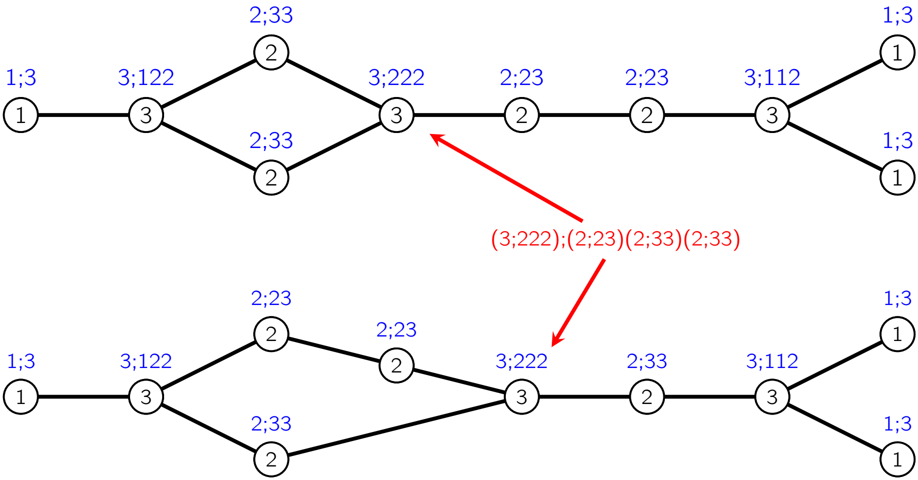
Color refinement (aka. 1-WL)



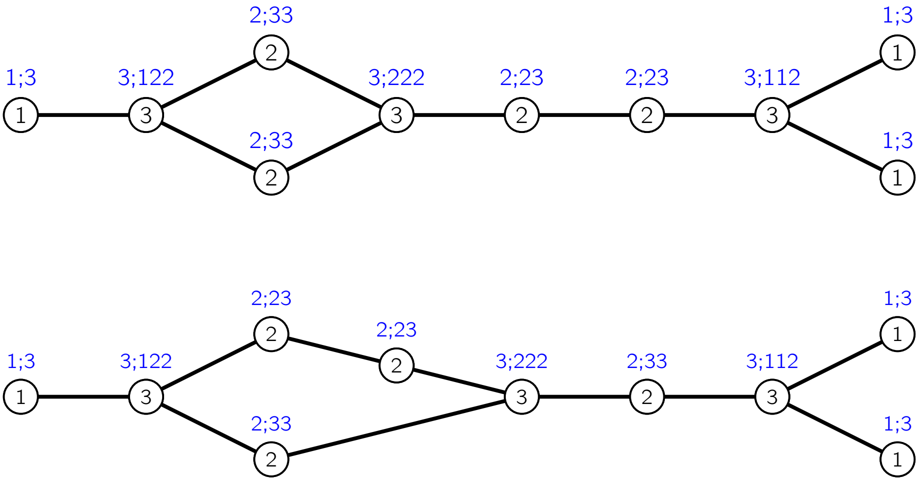
Color refinement (aka. 1-WL)



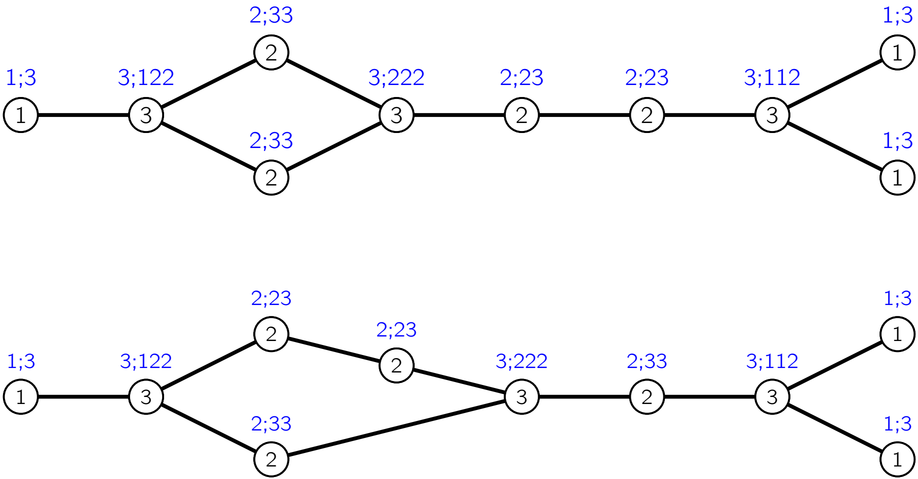
Color refinement (aka. 1-WL)



Color refinement (aka. 1-WL)

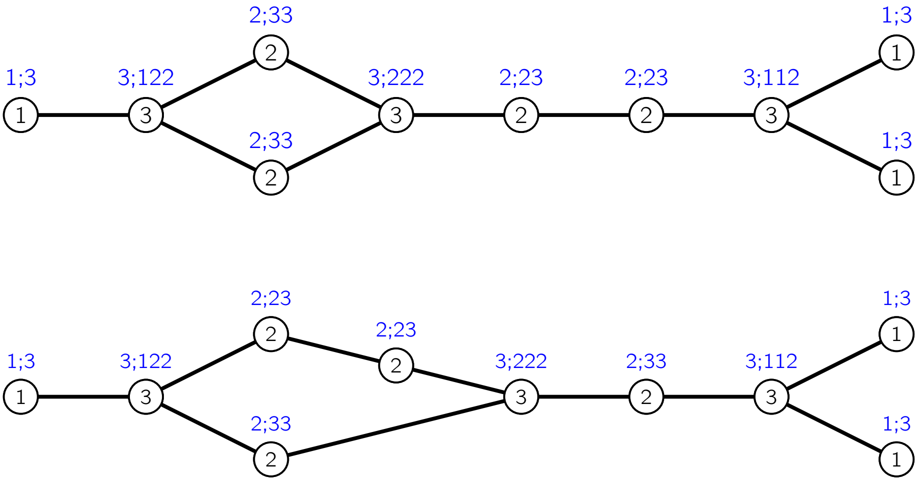


Color refinement (aka. 1-WL)



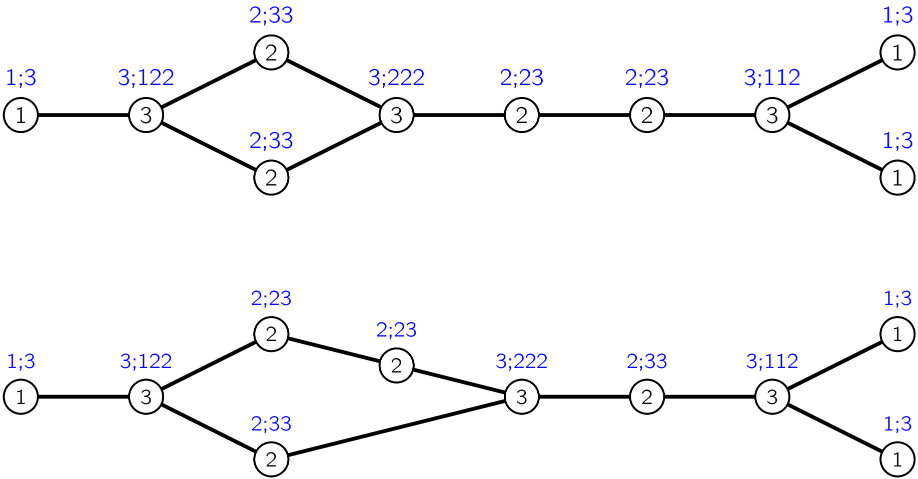
1;3
2;23
2;33
3;112
3;122
3;222

Color refinement (aka. 1-WL)



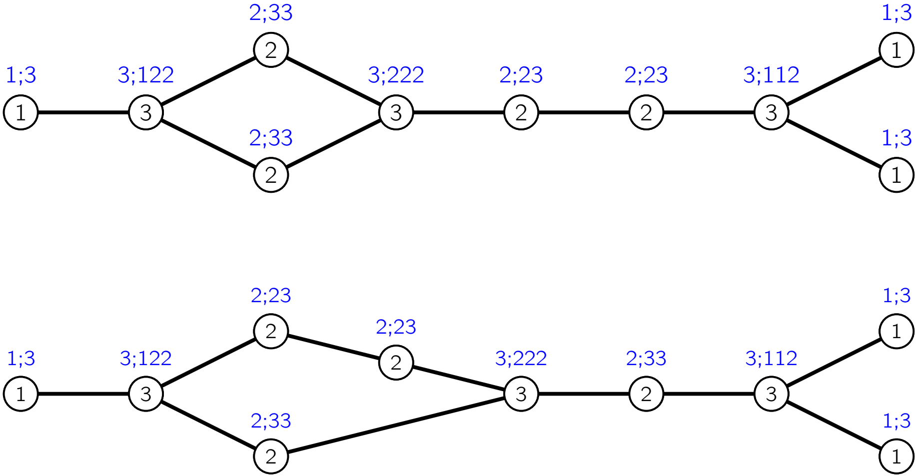
4 :=1;3
2;23
2;33
3;112
3;122
3;222

Color refinement (aka. 1-WL)



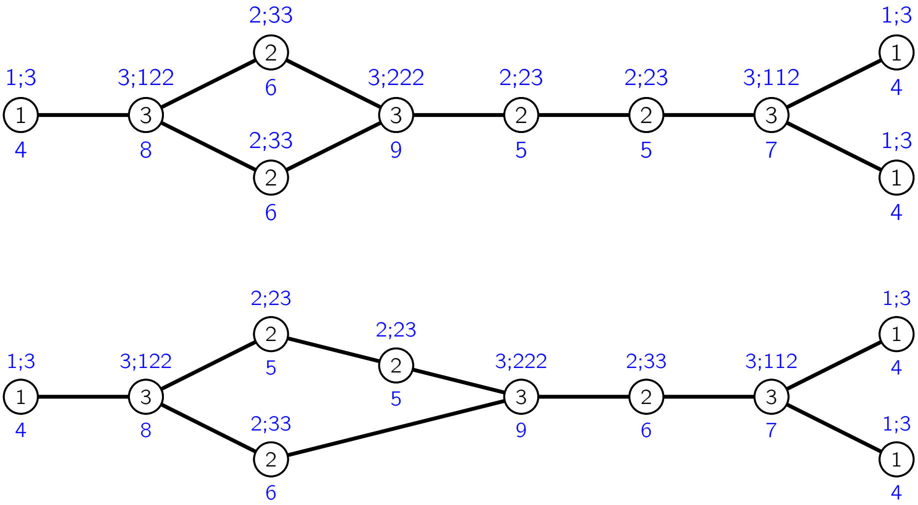
4 :=1;3
5 :=2;23
2;33
3;112
3;122
3;222

Color refinement (aka. 1-WL)



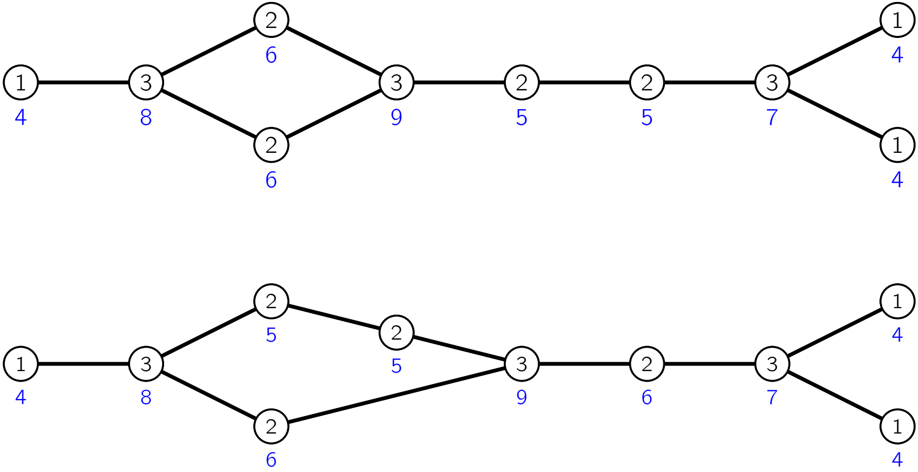
4 :=1;3
5 :=2;23
6 :=2;33
7 :=3;112
8 :=3;122
9 :=3;222

Color refinement (aka. 1-WL)

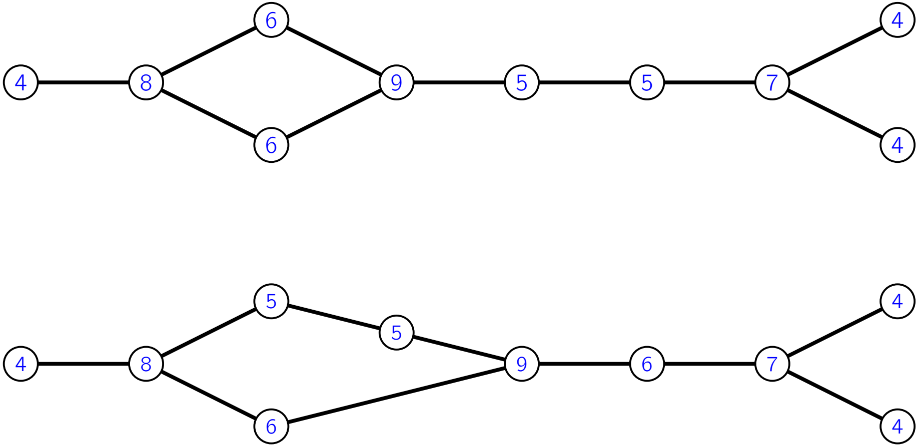


4 :=1;3
5 :=2;23
6 :=2;33
7 :=3;112
8 :=3;122
9 :=3;222

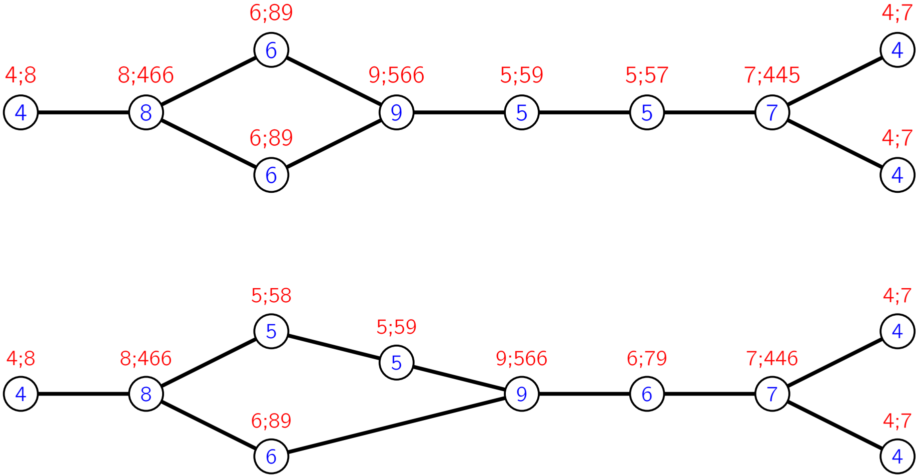
Color refinement (aka. 1-WL)



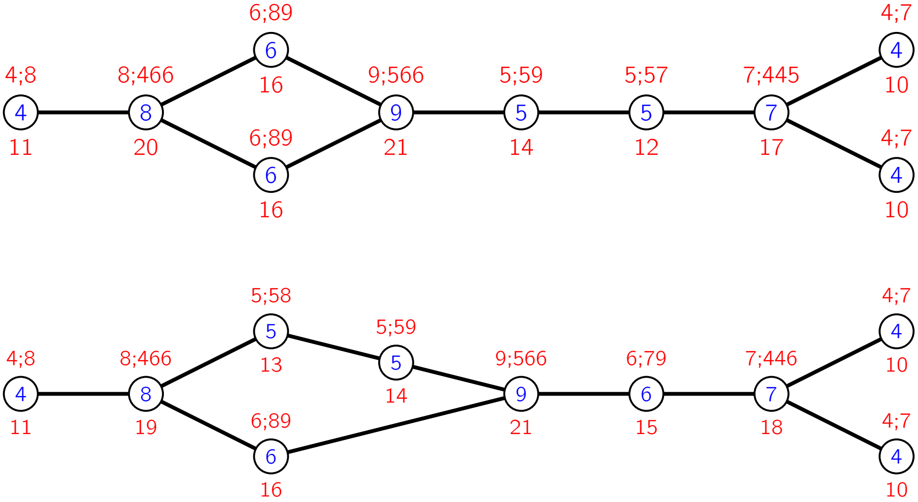
Color refinement (aka. 1-WL)



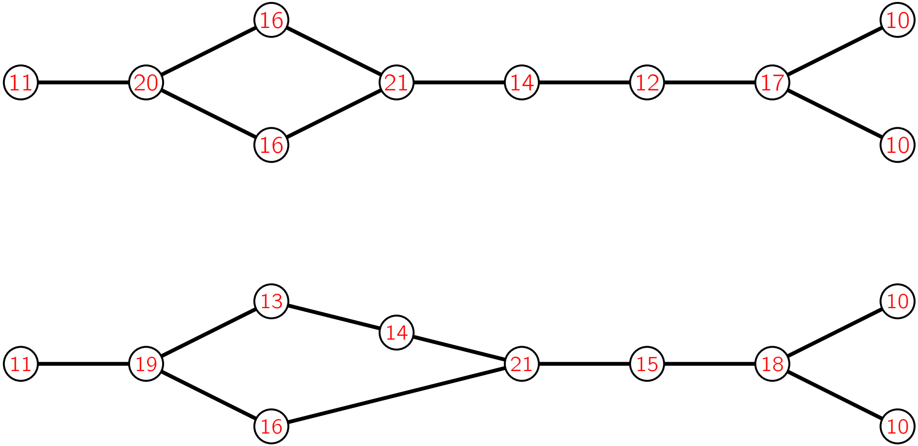
Color refinement (aka. 1-WL)



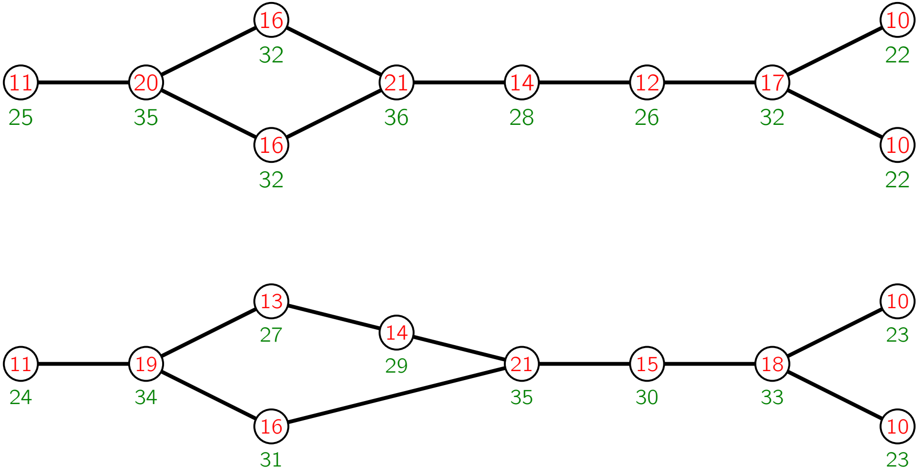
Color refinement (aka. 1-WL)



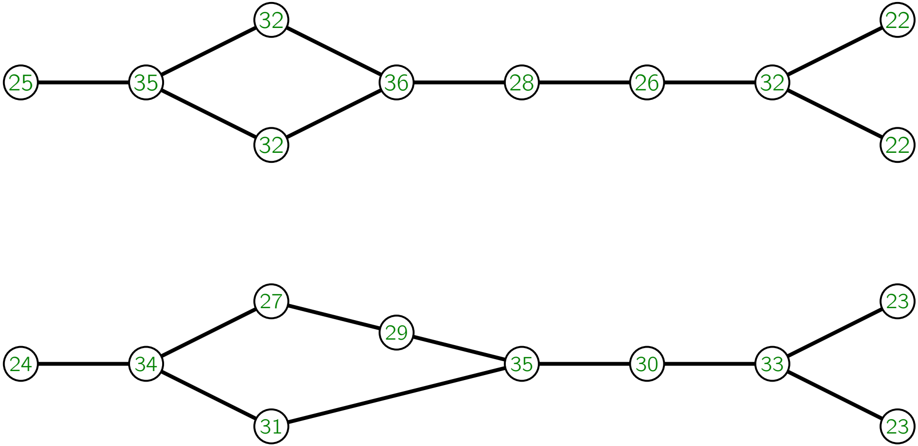
Color refinement (aka. 1-WL)



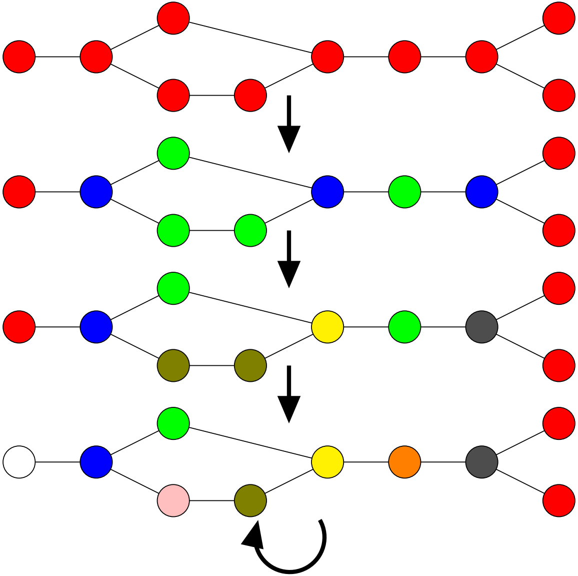
Color refinement (aka. 1-WL)



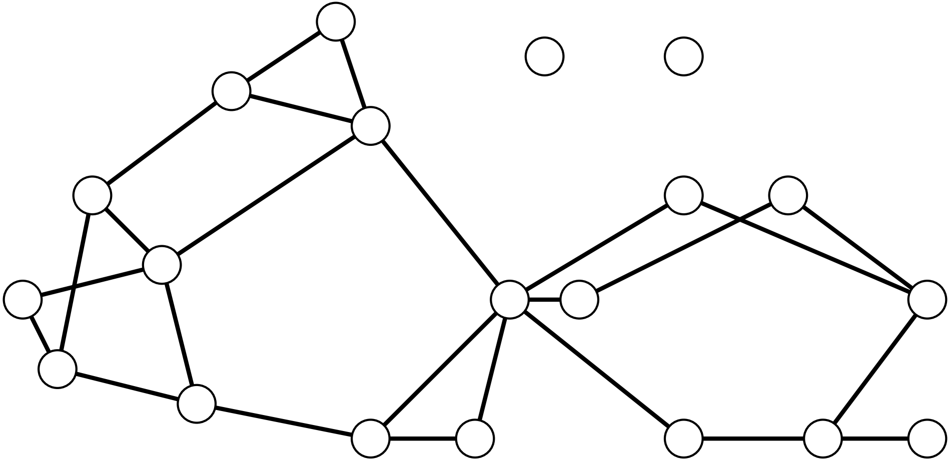
Color refinement (aka. 1-WL)



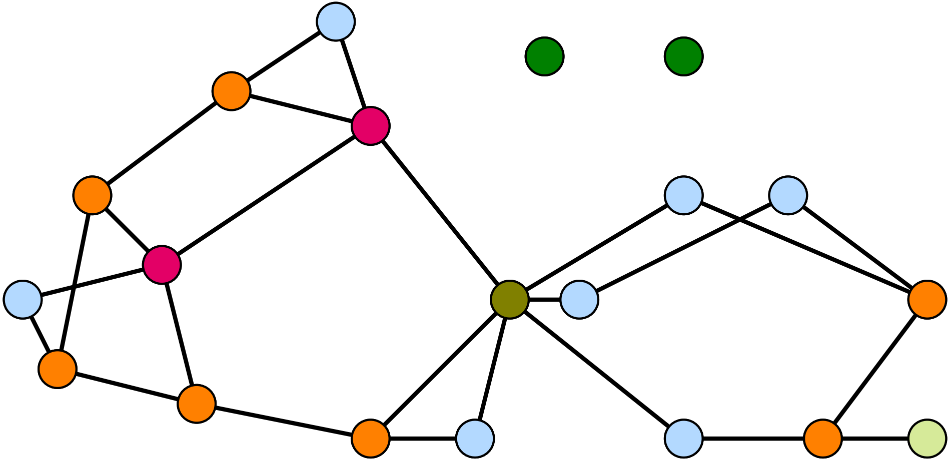
Color refinement (illustration with colors)



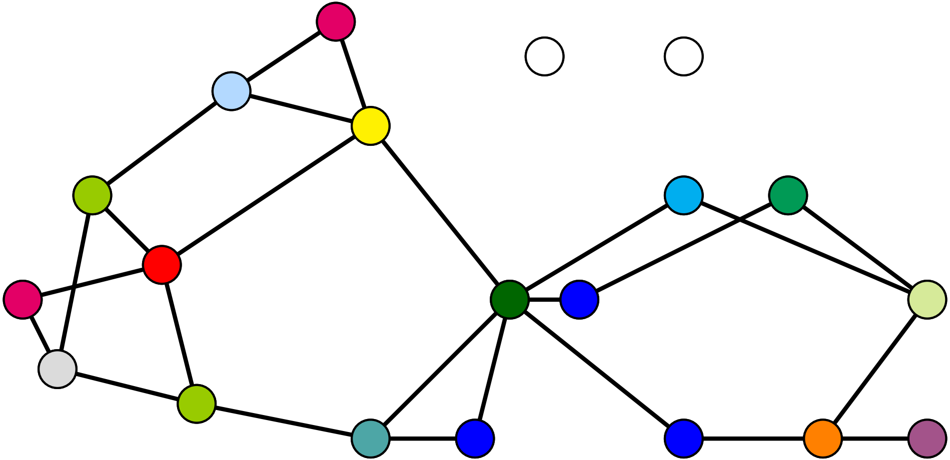
Color refinement on a random graph



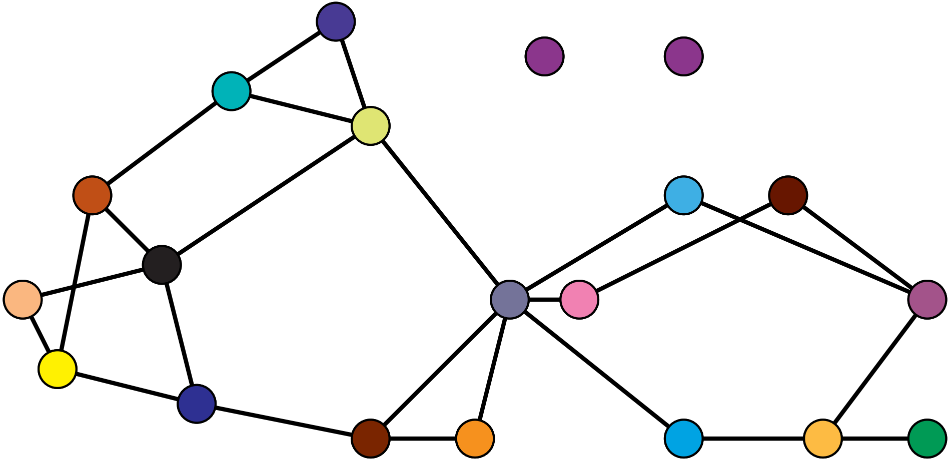
Color refinement on a random graph



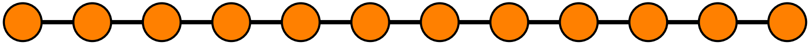
Color refinement on a random graph



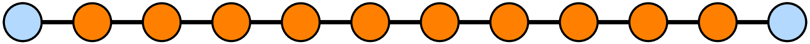
Color refinement on a random graph



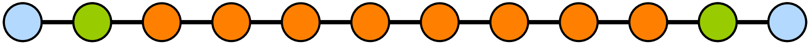
Color refinement on a path



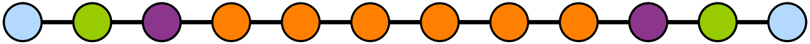
Color refinement on a path



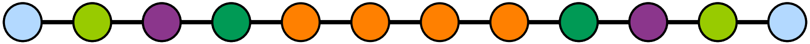
Color refinement on a path



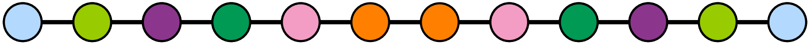
Color refinement on a path



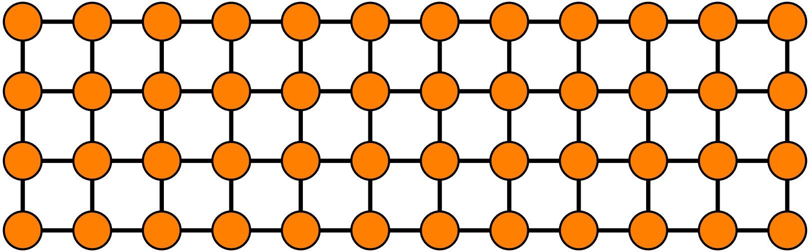
Color refinement on a path



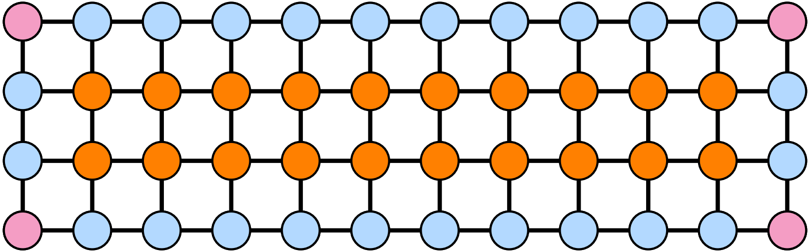
Color refinement on a path



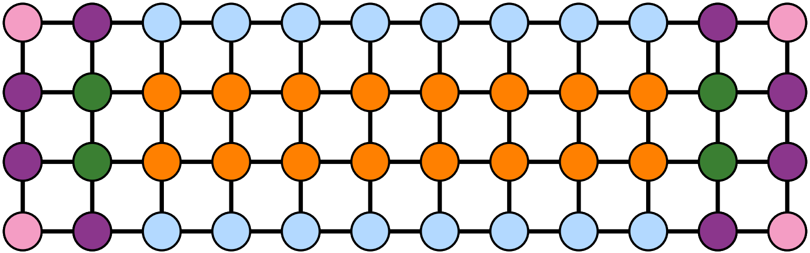
Color refinement on a grid



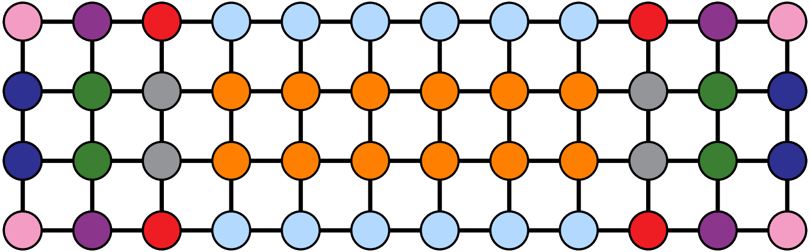
Color refinement on a grid



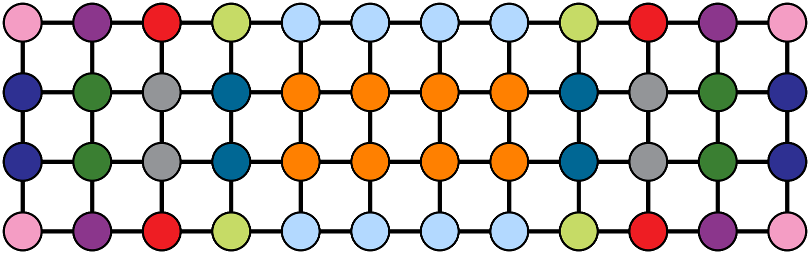
Color refinement on a grid



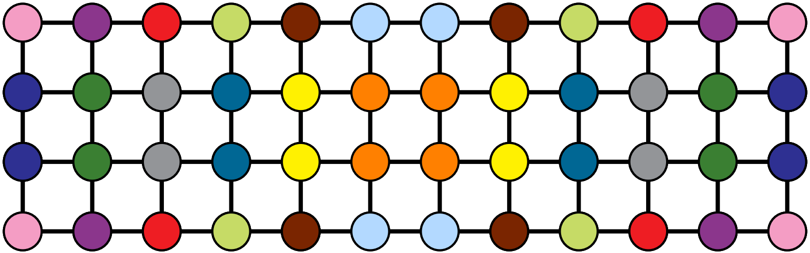
Color refinement on a grid



Color refinement on a grid



Color refinement on a grid



Some facts about 1-WL

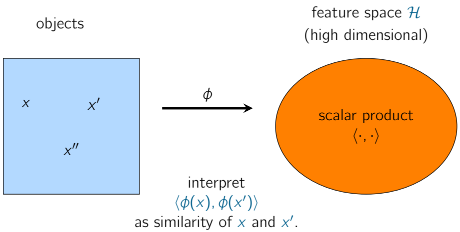
Facts about 1-WL:

- distinguishes almost all pairs of non-isomorphic graphs [Babai, Erdős, Selkow] (1980)
- cannot distinguish non-isomorphic regular graphs of same degree
- a precise characterization for which graphs it always works is known [Arvind, Köbler, Rattan, Verbitsky] [Kiefer, S., Selman] (2015)
- result computable in $O((m+n)\log n)$ [Cardon and Crochemore] (1982)
- used heavily in practice

Applications of 1-WL in machine learning on graphs

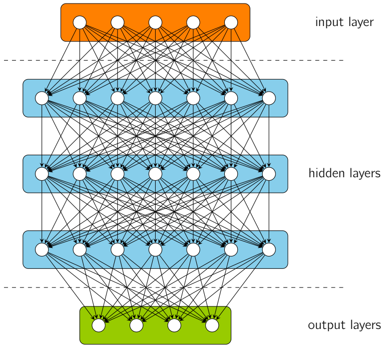
Two general graph isomorphism techniques for machine learning on graphs:

WL-kernels



[Shervashidze,S.,van Leeuwen,Mehlhorn,Borgwardt] (2011)

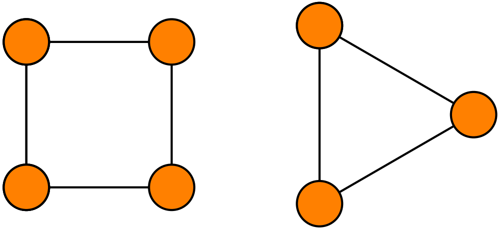
Universal graph neural networks



message passing neural networks $\Leftrightarrow$ 1-WL
[XHLJ] (2019) [MRFHLRG] (2019)
see also [Franks,Anders,Kloft,S.] (2023)

Limits of color refinement

Example

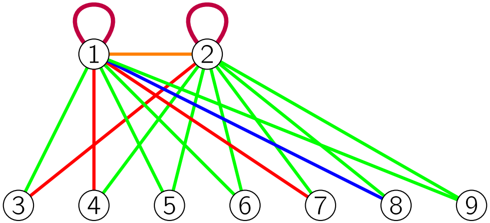


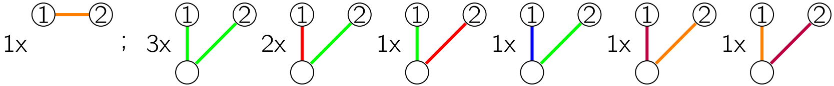
$C_4 \,\dot\cup\, C_3$

Consequence: To explore automorphisms we need to consider more than one vertex at a time.

The classical (2-dim) WL-algorithm

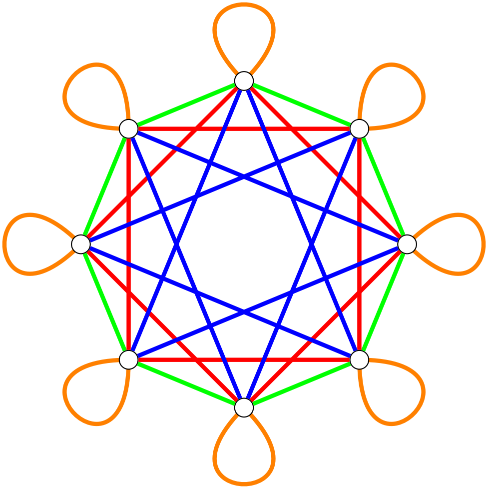
- The 2-dimensional WL-algorithm colors pairs of vertices.
- The initial coloring is into edges, non-edges and loops.
- Each iteration recolors each pair (u, v) according to colored triangles containing (u, v).



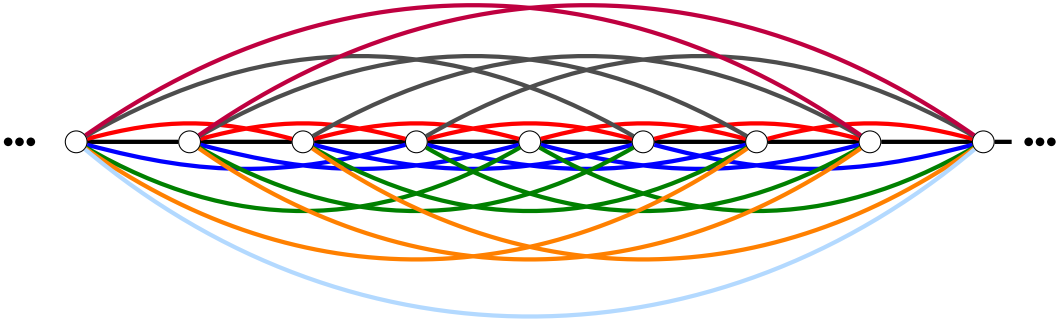


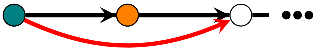
2-dim WL and distances (Illustration)

The 2-dim WL can "measure" distances in a graph.

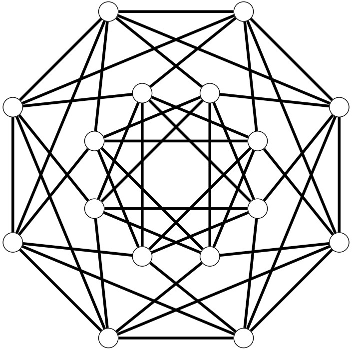


2-dim WL on paths

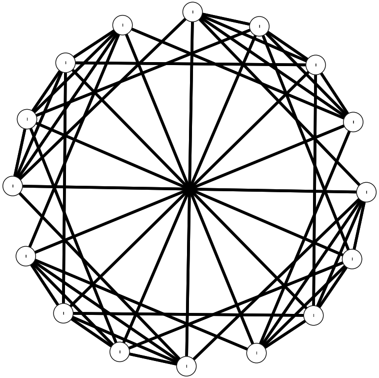




Non-isomorphic strongly regular graphs



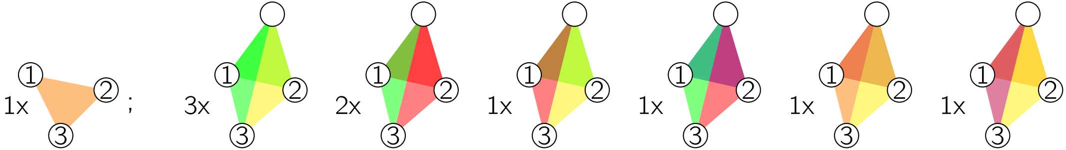
Shrikhande graph



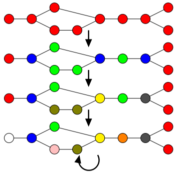
Line graph $L(K_{4,4})$

The smallest pair of non-isomorphic graphs not distinguished by 2-WL (16 vertices).

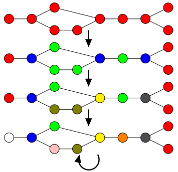
3-dim WL (Illustration)



1. WL Algorithm



1. WL Algorithm



2. Logics & Games

$\varphi \models G_1$ and $\varphi \not\models G_2$

$\Downarrow$

$G_1 \not\cong G_2$

Correspondence

There are three views with a well established **close correspondence** between parameters.

[Cai,Fürer,Immerman] (1992)

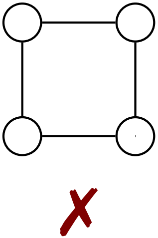
WL-refinement	dimension	iterations
$\updownarrow$	$\updownarrow$	$\updownarrow$
logic	variables	quantifier depth
$\updownarrow$	$\updownarrow$	$\updownarrow$
pebble game	no. of pebbles	length of optimal plays

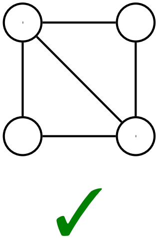
Logics and Sentences

Example

$$\exists x \qquad (\forall y \quad x = y \lor x \sim y) \lor (\forall y \quad x = y \lor x \not\sim y)$$

In words:
There is a vertex x that is adjacent to every other vertex or that is adjacent to no other vertex.





Observation
It is “somewhat easy” to decide whether a given graph satisfies the sentence.

Logic for isomorphism

Question: How can we use logic formulas for isomorphism tests?

Answer:

Fact

- If G_1 and G_2 are non-isomorphic graphs, there is a logical formula φ which is true for G_1 but false for G_2.
- We can use φ to observe G_1 and G_2 are not isomorphic.

$$\varphi \models G_1 \text{ and } \varphi \not\models G_2$$

$$\Downarrow$$

$$G_1 \not\cong G_2$$

Weisfeiler-Leman in terms of logic

The k-dimensional Weisfeiler-Leman algorithm is an algorithm that

- given graphs G_1 and G_2
- checks simultaneously for certain formulas φ whether
- φ distinguishes G_1 and G_2.

Running time:

- It runs in polynomial time $O(n^k \log(n))$.

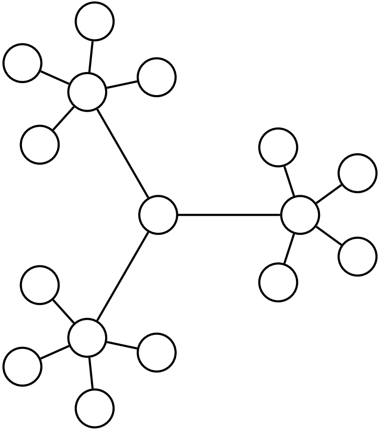
The counting logic

k-variable **counting logic** for graphs uses

- k variables
- counting quantifiers

Example: $\exists^{\geq 1} x. \quad \exists^{\geq 3} y. \quad x \sim y \wedge \exists^{\geq 5} x. \quad x \sim y.$

Read: there is a vertex with at least 3 neighbors of degree at least 5.



Fact:

k-WL checks all invariants expressible in the logic using only $k+1$-variables simultaneously.

Conclusion:

k-WL is thus a complete check for invariants in a natural, large class

Homomorphism counts

graph homomorphism: map between graphs preserving adjacency

$\hom(G, H)$: the set of homomorphisms from G to H.

Theorem ([Lovász] (1967))

Two finite graphs G_1, G_2 are isomorphic if and only if their homomorphism counts from all finite graphs agree, i.e., $|\hom(H, G_1)| = |\hom(H, G_2)|$ for all finite graphs H.

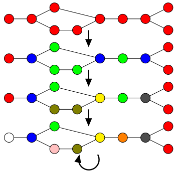
Similarity measures through homomorphism counts

Question: What happens if H comes from a restricted graph class?

count homs from	equivalence measure	
all graphs	isomorphism	[Lovász] (1967)
trees	fractional isom./1-WL equiv.	
tree width $\leq k$	k-WL equivalence	[Dvořák] (2010)
		[Dell,Grohe,Rattan] (2018)
cycles	cospectrality	
planar graphs	quantum isomorphism	[Manǒinska,Roberson] (2020)
tree depth	FO+C bounded qr	[Grohe] (2020)
width/depth restr. tree dec.	res. requant./conj.	[Schindling] (2025)

Many related recent results: [Seppelt] [Neuen] [Dawar,Jakl,Reggio] [Böker] [Atserias] . . .

1. WL Algorithm



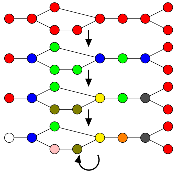
2. Logics & Games

$\varphi \models G_1$ and $\varphi \not\models G_2$

$\Downarrow$

$G_1 \not\cong G_2$

1. WL Algorithm



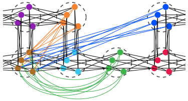
2. Logics & Games

$\varphi \models G_1$ and $\varphi \not\models G_2$

$\Downarrow$

$G_1 \not\cong G_2$

3. Recent developments



Research questions

General question:

- How complicated does the logical formula have to be to distinguish graphs?

Specifications:

- How many variables do we need?
- How many iterations (requantifications) do we need? (quantifier depth)
- Can we restrict requantification?
- What are the computational complexities that arise?
- What if we consider structures other than graphs?

The Weisfeiler-Leman Dimension

Definition

The Weisfeiler-Leman dimension of a graph G is the smallest k such that k-dimensional WL "always works" for G.

Isomorphism for graphs of bounded WL-dimension can be checked in polynomial time.

- Graphs with an excluded minor have bounded WL-dimension. [Grohe] (2012)
- For all orders n, there are graphs of WL-dimension $\Omega(n)$. [Cai,Fürer,Immerman] (1992)

Side remark: $O(\log(n))$-dimensional WL used in Babai's quasipolynomial time algorithm.

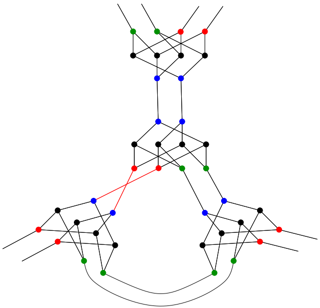
Upper bounds on the WL dimension for graph classes

graph class	dimension bound	reference	date
K_t-minor-free	$f(t)$	[Grohe]	(2012)
planar	3	[Kiefer, Ponomarekno, S.]	(2017)
genus g	$4g + 3$	[Kiefer, Grohe]	(2019)
tree width k	k	[Kiefer, Neuen]	(2019)
rank width k	3k+4	[Grohe, Neuen]	(2019)
permutation graphs	18	[Guo, Gavrilyuk, Ponomarenko]	(2023)
circulant graph	$\#\text{prime-div}(\|G\|) + 3$	[Wu, Ponomarenko]	(2024)

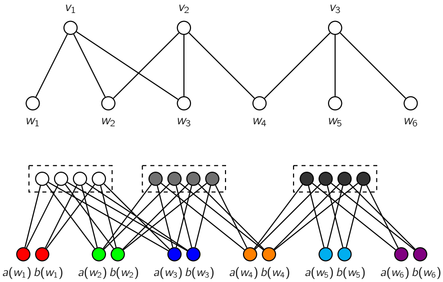
Unbounded WL-dimension

Theorem ([Cai, Fürer, Immerman] (1992))

There are graphs of arbitrarily large Weisfeiler-Leman dimension.



CFI Graphs



Multipedes

Bounds for WL iteration Number (i.e., quantifier depth)

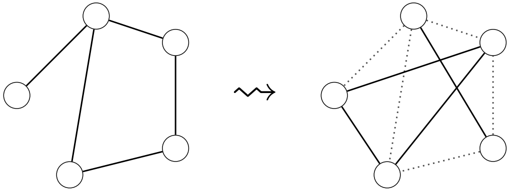
dimension		**bound**	**reference**	**year**
1-dim	upper bound	$n-1$	(by def.)	
	lower bound	$n-o(n)$	[VK]	(2014)
		$n-1$	[KM]	(2020)
2-dim	upper bound	$O(n^2/\log(n))$	[KS]	(2016)
		$O(n\log n)$	[LPS]	(2019)
	lower bound	$\Omega(n)$	[F]	(2001)
k-dim	upper bound	$O(n^{k-1}\log n)$	[GLN]	(2023)
	lower bound	$\Omega(n)$	[F]	(2001)
		$\Omega(n^{k/2})$	[GLNS]	(2023)

Iterated neighborhood graphs

Definition

The neighborhood graph $N(G)$ of G

- has the same vertex set as G and
- two vertices are adjacent if they have a common neighbor if G.



Observations:

- Bipartite graphs become disconnected.
- Odd cycles become an isomorphic non-identical graph.

Iterated neighborhood graphs: $N^i(G) = N(N^{i-1}(G))$ [Sonntag, Teichert] (KolKom09)

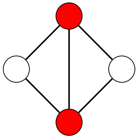
Theorem ([S.] (2013))

For every finite connected non-bipartite graph G that is not an odd cycle we have tight bounds

$$\lceil \log 2(\mathrm{Diam}(G)) \rceil \leq \mathrm{Stab}(G) \leq \lceil \log 2(\mathrm{Diam}(G)) \rceil + 2.$$

Generalized neighborhood graph

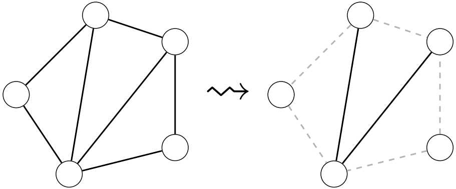
H a graph with two distinguished vertices.

H


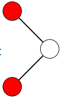
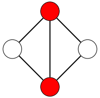
Definition

The generalized neighborhood graph $N_H(G)$

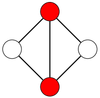
- has the same vertex set as G and
- two vertices are adjacent if they are the distinguished vertices in a copy of H in G.

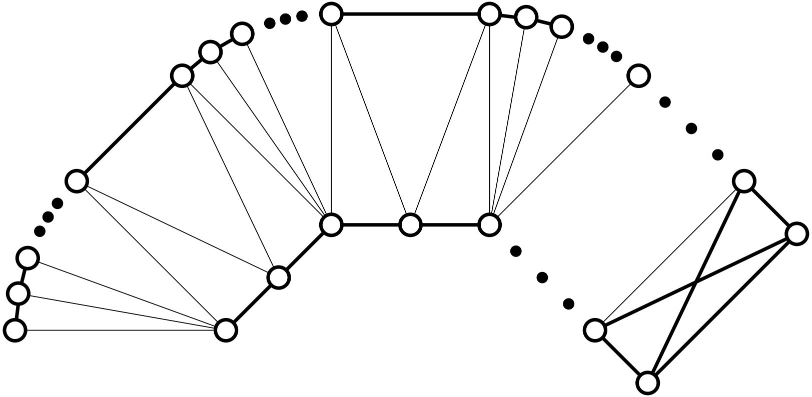


Observation and Results: (see [S.] (2013))

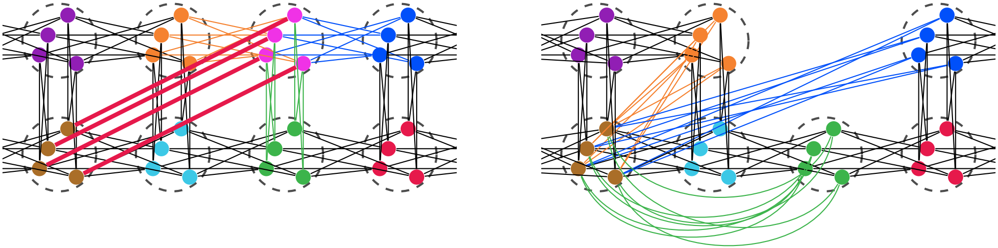
- $H =$  recovers neighborhood graphs.
- For $H =$  (diamond) the process is monotone but stabilization can take $\Omega(n^2)$.
- We can construct H to simulate the game of life (and get almost anything we like).

The lower bound construction

A figure explaining the $\Omega(n^2)$ lower bound for $H =$  (diamond).



The compressed CFI-construction



- The lower bound for k-WL is obtained using a compressed CFI-construction.
  [Grohe, Lichter, Neuen, S.]
- used for supercritical, robust trade-offs (e.g., depth vs width) for proof systems
  [de Rezende, Fleming, Janett, Nordström, Pang]
  [Berkholz, Lichter, Vinall-Smeeth]

Complexity

Theorem (Lichter, Raßmann, S. (paraphrased) (2025))

For determining the WL-dimension of a graph we have the following results:

- *It is NP-hard even on graphs of color class size 4.*
- *For fixed k and color class size at most 5 it is solvable in polynomial time.*
- *It is P-hard for fixed k.*
- *For Abelian color classes and fixed k, we can solve the problem in polynomial time.*

Open Problem:

Can we decide for each fixed k in polynomial time whether the WL-dimension is at most k?

Requantification

Question

What happens if we disallow certain variables to be nested within themselves (i.e., requantified)?

This question is related to **individualization-refinement** algorithms used in practice to compute isomorphism and automorphism.

Theorem (Raßmann, Schindling, S. (paraphrased) (2025))

- *We know how the expressiveness changes if we restrict requantification.*
- *We have a corresponding restricted logic and bijective pebble game.*
- *There are no succinct normal forms for restricted requantification.*
- *We know how the space consumptions changes when requantification is restricted.*

WL for groups

- There are several natural ways to define WL for groups, they all agree up to constants.
- There are "very similar" groups distinguished by 3-WL.

[Brachter, S.] (2020)

Beyond that there are

- other ways of incorporating WL into group isomorphism algorithms and
- various complexity results

[Brooksbank, Grochow, Li, Qiao, Wilson]

Open Problem: It is unknown whether the WL-dimension of groups is unbounded.

WL and isomorphism invariants of groups

Theorem

If groups are not distinguished by k-WL then k-WL does not distinguish their

- *centers* ($k \geq 2$),
- *inner automorphism groups* ($k \geq 4$)
- *derived series* ($k \geq 3$),
- *Abelian radicals* ($k \geq 3$),
- *solvable radicals* ($k \geq 2$),
- *Fitting groups* ($k \geq 3$), *and*
- π*-radicals* ($k \geq 3$)

They also have isomorphic socles ($k \geq 5$), *stepwise isomorphic factors in the derived series* ($k \geq 4$), *upper central series* ($k \geq 4$), *and lower central series* ($k \geq 4$).

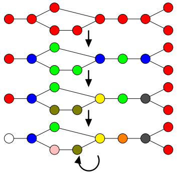
Theorem

If two groups are not distinguished by k-WL then

- *they have the same composition factors (as a multiset,* $k \geq 5$*) and*
- *their indecomposable direct factors are not distinguished by* $(k-1)$*-WL* ($k \geq 5$).

[Brachter, S.] (2022)

1. WL Algorithm



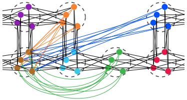
2. Logics & Games

$\varphi \models G_1$ and $\varphi \not\models G_2$

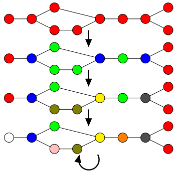
$\Downarrow$

$G_1 \ncong G_2$

3. Recent developments



1. WL Algorithm



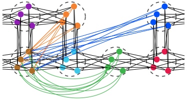
2. Logics & Games

$\varphi \models G_1$ and $\varphi \not\models G_2$

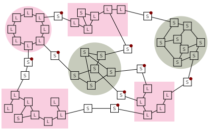
$\Downarrow$

$G_1 \not\cong G_2$

3. Recent developments



4. Dimension bound



Bounds for the dimension

Theorem (Schneider, S. (2025))

The maximum Weisfeiler-Leman dimension among all graphs on n nodes is

- *at most* $\frac{3}{20}n + o(n)$,
- *at least* $0.0105n$.

Related work

- $\leq \frac{n+3}{2}$ [Pikhurko, Veith, Verbitsky] (2006)
- $\leq \frac{n}{3}$ [Lutz] (2020)
- $\leq \frac{n}{4} + o(n)$ and $\geq \frac{1}{96}n - o(n)$. independently [Kiefer & Neuen] (2024)

Proof outline:

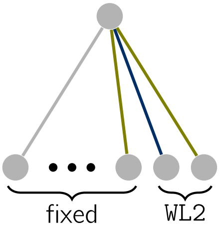
In a nutshell:

1. reduction to coherent configurations (i.e., objects stable under 2WL)
2. bounds for configurations with small fibers (size at most 7)
3. classification of small interspaces
4. new concept of restorability
5. local reductions
6. a global structure analysis

Theorem (Schneider, S. (2025))

The maximum Weisfeiler-Leman dimension among all graphs on n nodes and color class size at most 7 is at most $\frac{1}{20}n + o(n)$.

Local Reductions



Goal: avoid certain substructures

Approach:

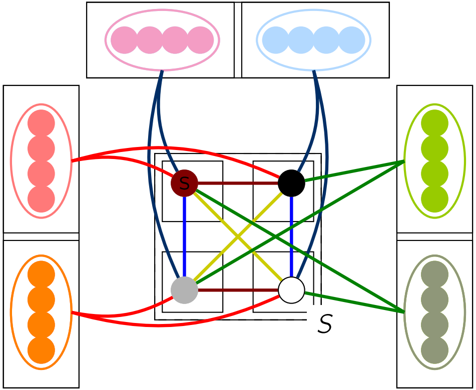
- individualize x vertices and apply 2-WL
- charge cost of individualization to "progress" measured as splits of vertex color classes

Potential function: $\tau(G) := \frac{3n_L - 8k_L + n_S}{20}$ where

- n_L number of vertices in large color classes
- k_L number of large color classes
- n_S number of vertices in small color classes

Local reductions blueprint:
If G has [certain substructure], then $\dim_{\mathrm{WL}} G \leq x + \tilde{f}(\tau(G) - \hat{x})$ where $\hat{x} \geq x$.

Local Reductions: Example

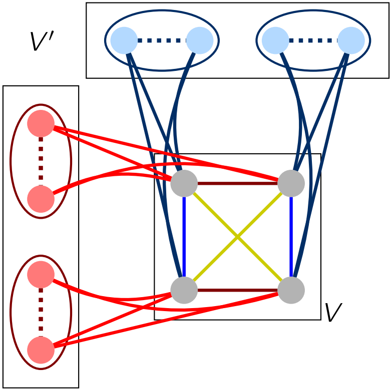


For arbitrary $s \in S$, progress in G_s

$$\begin{aligned}
&\tau(G_s) - \tau(G) \\
&\leq \frac{3\Delta n_L - 8\Delta k_L + \Delta n_S}{20} \\
&\leq \frac{(3 \cdot -24) - (8 \cdot -3) + (24 - 4)}{20} \\
&\leq -1.4
\end{aligned}$$

$$\dim_{\mathrm{WL}} G \leq 1 + \widetilde{f}(\tau(G) - 1.4).$$

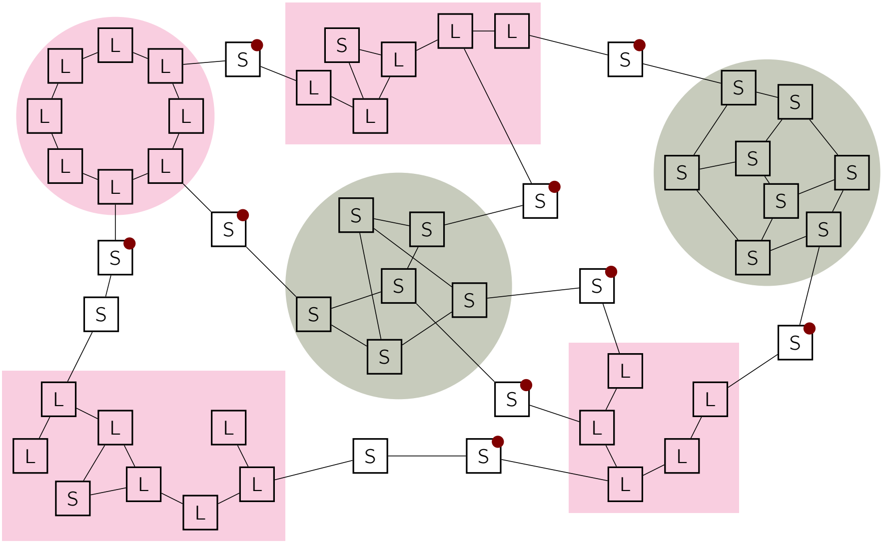
Restorability



We call a color class C restorable if every automorphism of $G[N(C)]$ that extends to an automorphism of $G[V - C]$ also extends to an automorphism of $G[C \cup N(C)]$.

Insight: Removing a restorable class does not change the WL-dimension.

Global Argument

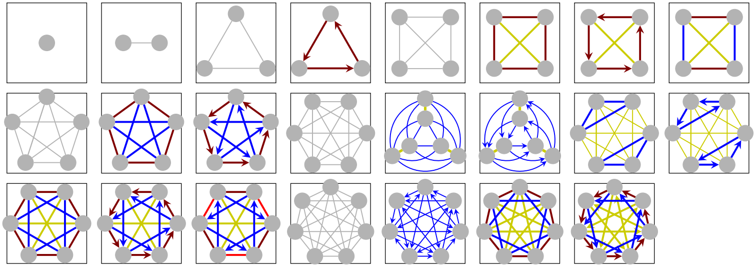


Small Color Classes

Color class R is **large** if $|R| \geq 8$, **small** if $7 \geq |R| \geq 4$, *and* **tiny** if $3 \geq |R|$.

Color class R is	**large**	if	$|R| \geq 8$,
	small	if	$7 \geq |R| \geq 4$, *and*
	tiny	if	$3 \geq |R|$.

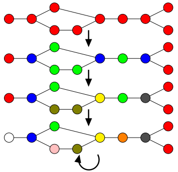
Enumeration of all homogeneous coherent configurations of order at most 34 (up to isomorphism) [Miyamoto, Hanaki] (00), [Hanaki] (03)



Interspace between small fibers

$\|R\|$	$\|B\|$	$\mathsf{T}(\mathfrak{X}[R,B])$
4	4	(C_8, C_8), $(2K_{2,2}, 2K_{2,2})$
4	6	$(Sp_{4,6}, Sp_{4,6})$, $(2K_{2,3}, 2K_{2,3})$
6	6	$(C_{12}, C_{12}, 3K_{2,2})$, $(2K_{3,3}, 2K_{3,3})$, $(3K_{2,2}, 3K_{2,2}, 3K_{2,2})$, $(3K_{2,2}, R \times B - 3K_{2,2})$
7	7	$(\mathrm{L}(\mathrm{FP}), R \times B - \mathrm{L}(\mathrm{FP}))$

1. WL Algorithm



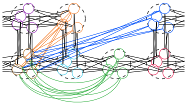
2. Logics & Games

$\varphi \models G_1$ and $\varphi \not\models G_2$

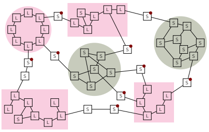
$\Downarrow$

$G_1 \not\cong G_2$

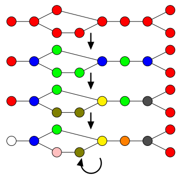
3. Recent developments



4. Dimension bound



1. WL Algorithm



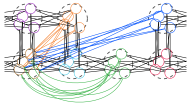
2. Logics & Games

$\varphi \models G_1$ and $\varphi \not\models G_2$

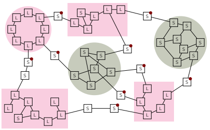
$\Downarrow$

$G_1 \not\cong G_2$

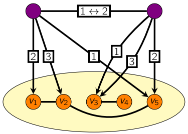
3. Recent developments



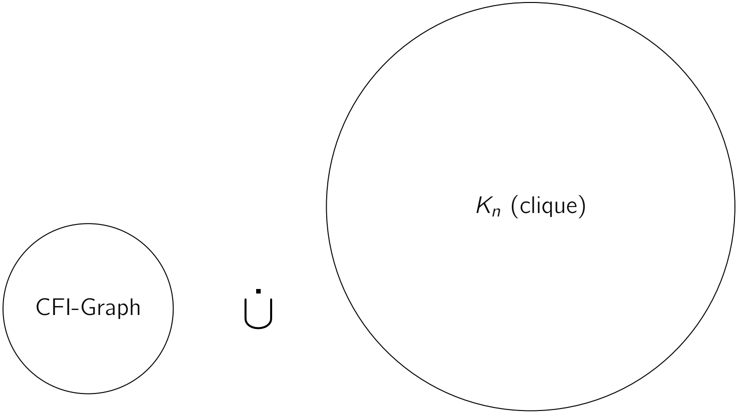
4. Dimension bound



5. Deep WL



Locally difficult graphs

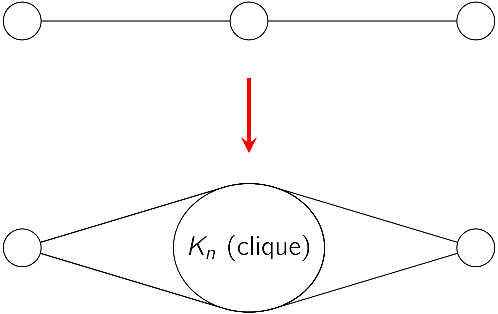


order $\log(\log(n))$ (tiny) order n (huge)

Idea: We should run k-WL for high k only on the small CFI-Graph.

Globally difficult graphs

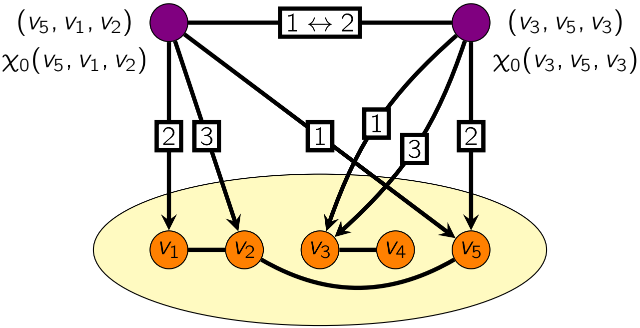
Consider CFI-Graph in which every vertex is blown up:



Idea: Treat the clique vertices like one object

k-dim WL (alternative View)

Alternative view for k-WL: We extend the structure by points representing k-tuples.



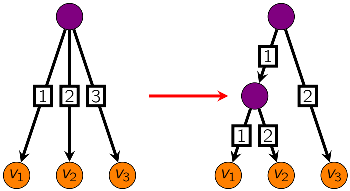
Facts:

$(k-1)$-WL $\preceq$ (1-WL on k-tuple extended structure) $\preceq$ k-WL (see [Otto] (1997))

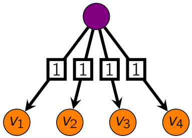
We can omit vertical connections and vertex colors by always using 2-dim WL.

Essential operations

2-tuples sufficient to simulate k-tuples:
$(v_1, v_2, v_3) \rightsquigarrow (v_1, (v_2, v_3))$



To handle blow ups, we can contract several vertices to one.

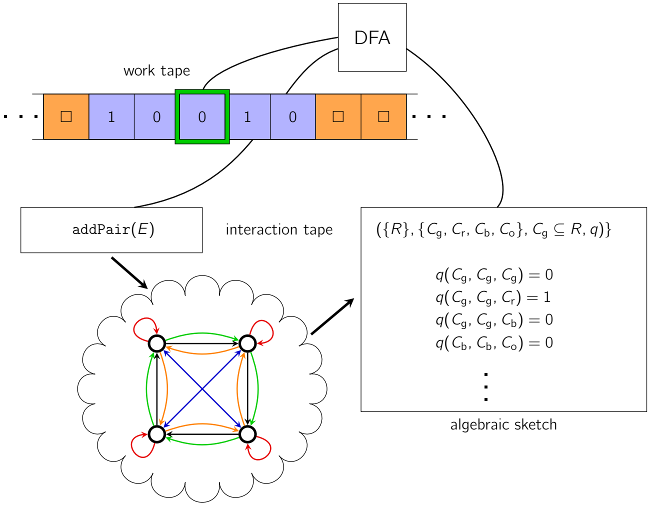


We need to make sure everything is **isomorphism invariant**.

- 2-tuples are inserted for entire color classes at a time.
- Contractions are only applied to strongly connected components (of constituent graphs).

Deep Weisfeiler Leman

- `addPair`(X): add vertices for all pairs (u, v) of color X.
- `contract`(X): add a vertex for each strongly connected component of X.
- `create`(π): create new relation that is union of all colors in π.
- `forget`(X): delete relation X.



Deep-WL and canonization

Theorem ([Grohe, S., Wiebking] (2020))

Let $\mathcal{G}$ be a graph class with polynomial-time Deep-WL-algorithm isomorphism algorithm. Then there is a polynomial-time Deep-WL-algorithm that computes a complete invariant for $\mathcal{G}$.

Corollary

Let $\mathcal{G}$ be a class of colored graphs closed under recoloring and with polynomial time Deep-WL isomorphism test. Then there is a polynomial time canonization algorithm for $\mathcal{G}$.

The power of Deep-WL

Theorem ([Grohe, S., Wiebking] (2020))

A property of graphs is decidable by a polynomial time Deep-WL-algorithm if and only if it is expressible in Choiceless Polynomial Time.

Fact: Isomorphism of CFI-graphs is expressible in Choiceless Polynomial Time.

[Dawar, Richerby, Rossman] (2008)

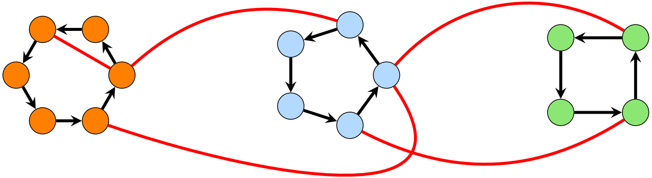
Corollary

There is a polynomial time Deep-WL-algorithm that decides isomorphism of the CFI graphs.

Abelian and Dihedral color classes

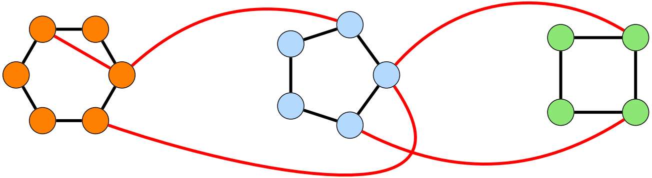
Theorem ([Abu Zaid, Grädel, Grohe, Pakusa] (2014))

Deep-WL decides isomorphism for structures with abelian color classes of bounded size.

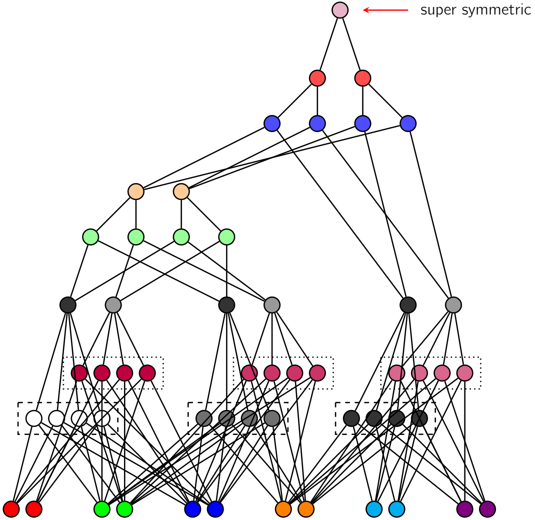


Theorem ([Lichter, S.] (2021))

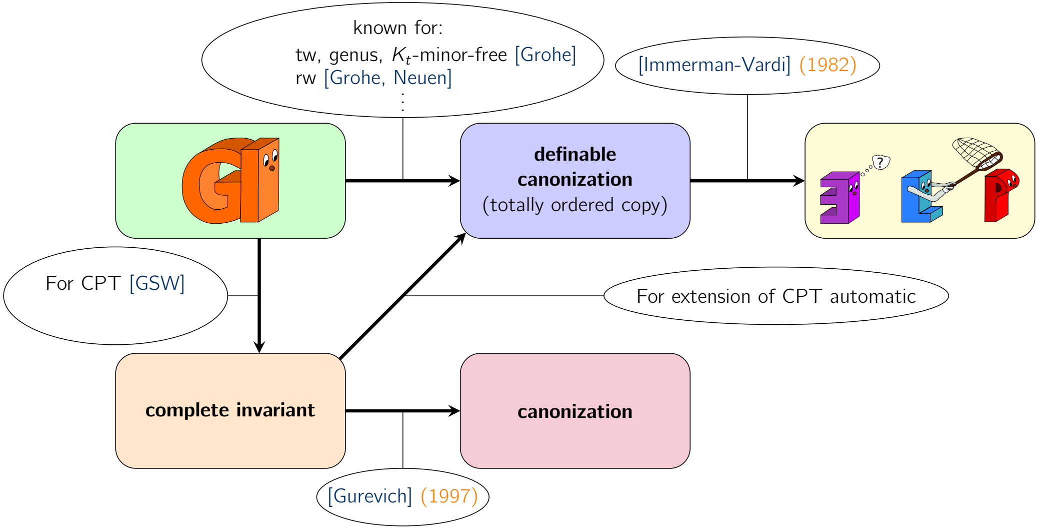
Deep-WL decides isomorphism for graphs with dihedral color classes of bounded size.



CFI with DeepWL (an intuition)



Capturing by canonization



Isomorphism to P-time logic

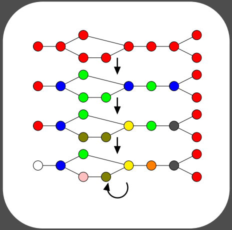
Theorem (Lichter, S. (2022))

Let $\mathcal{C}$ be a class of structures (closed under individualization).

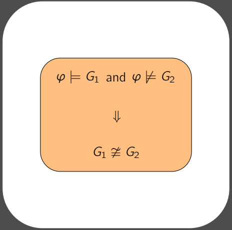
If isomorphism in $\mathcal{C}$ is definable in CPT+WSC then CPT+WSC captures P-time on $\mathcal{C}$.

(We also have some CFI-query decidable in CPT+WSC not known to be decidable in CPT).

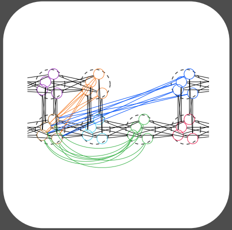
1. WL Algorithm



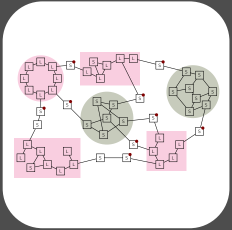
2. Logics & Games



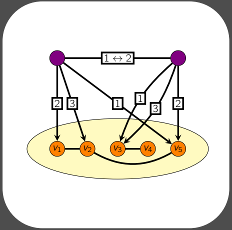
3. Recent developments



4. Dimension bound



5. Deep WL



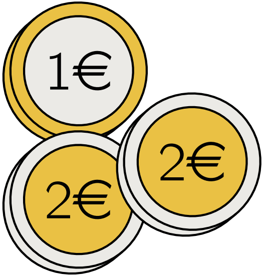
Cumulative prize money

Prize for a proof that **GI** $\in$ P or that **GI** $\notin$ P!

805Euro

Cumulative prize money

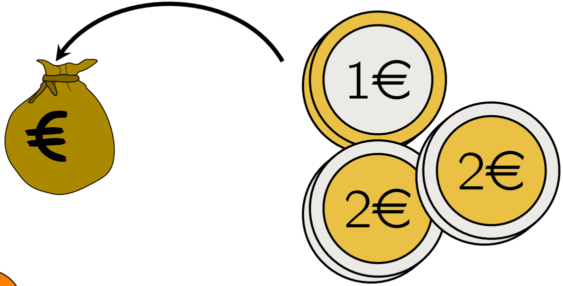
Prize for a proof that **GI** $\in$ P or that **GI** $\notin$ P!



805Euro $+\ 5$

Cumulative prize money

Prize for a proof that **GI** $\in$ P or that **GI** $\notin$ P!



805Euro $+ 5$

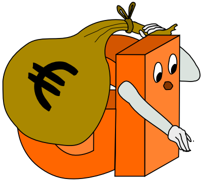
Cumulative prize money

Prize for a proof that **GI** $\in$ P or that **GI** $\notin$ P!

810Euro

Cumulative prize money

Prize for a proof that **GI** $\in$ P or that **GI** $\notin$ P!



810Euro

Questions?