On Numbers of Simplicial Walks and Equivalent Canonizations for Graph Recognition

Marek Černý [1]

[1] University of Antwerp, Antwerp, Belgium

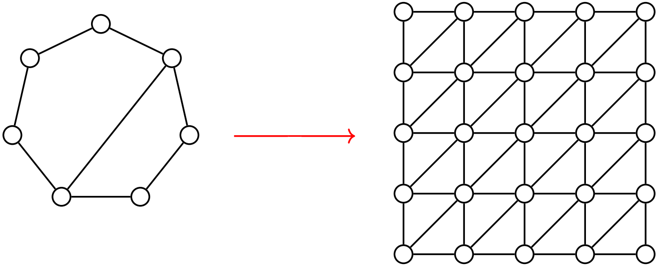
LATIN, April 2026

Graph Homomorphisms

Definition

A *(graph) homomorphism* $F \longrightarrow G$ is a function $\varphi : V(F) \to V(G)$ such that

$$uv \in E(F) \implies \varphi(u)\varphi(v) \in E(G).$$

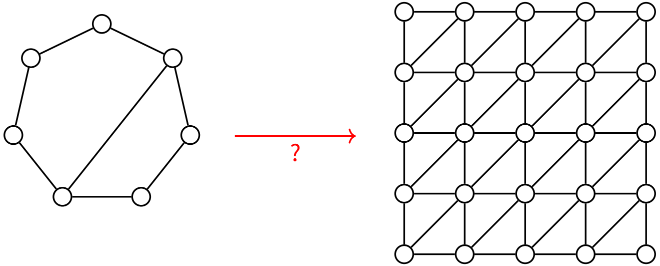


Graph Homomorphisms

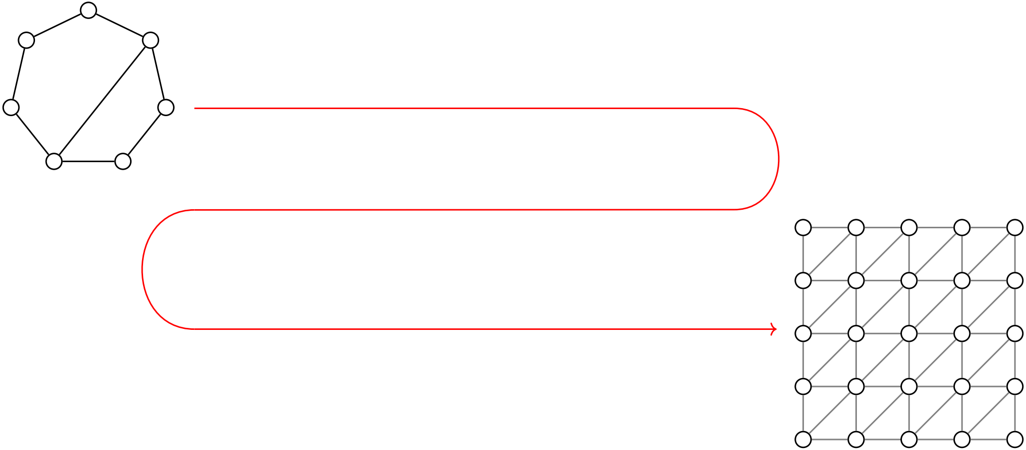
Definition

A *(graph) homomorphism* $F \longrightarrow G$ is a function $\varphi : V(F) \to V(G)$ such that

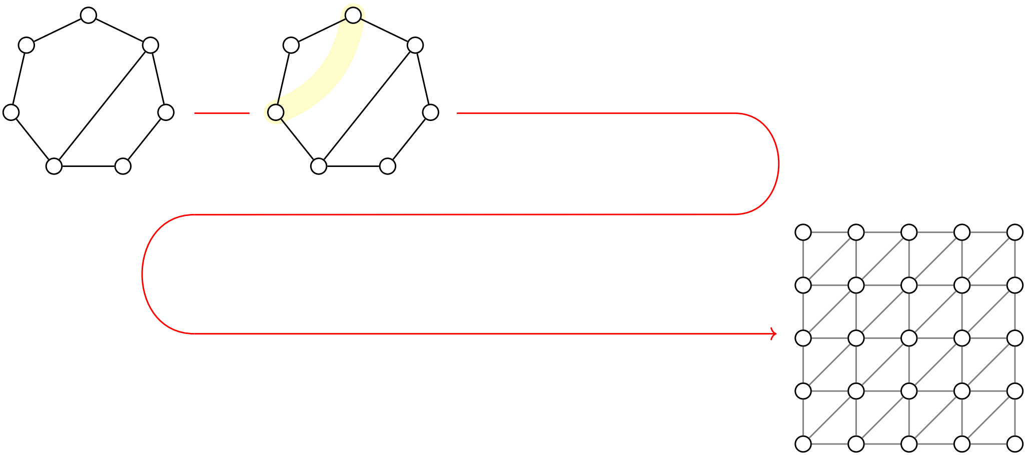
$$uv \in E(F) \implies \varphi(u)\varphi(v) \in E(G).$$



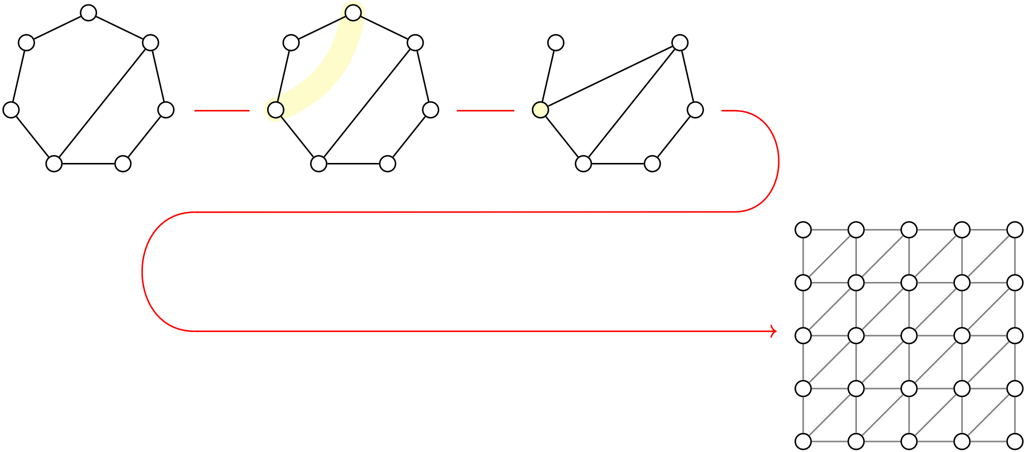
Example of a homomorphism $F \longrightarrow G$.



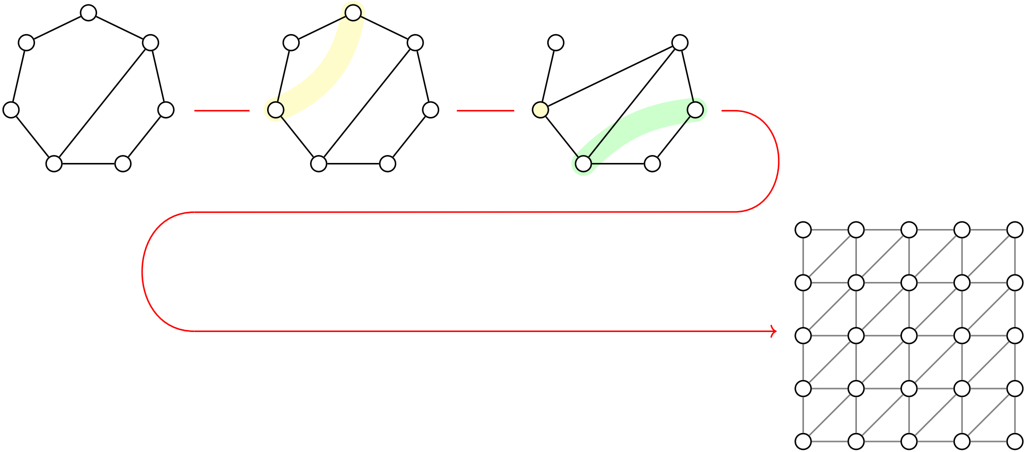
Example of a homomorphism $F \longrightarrow G$.



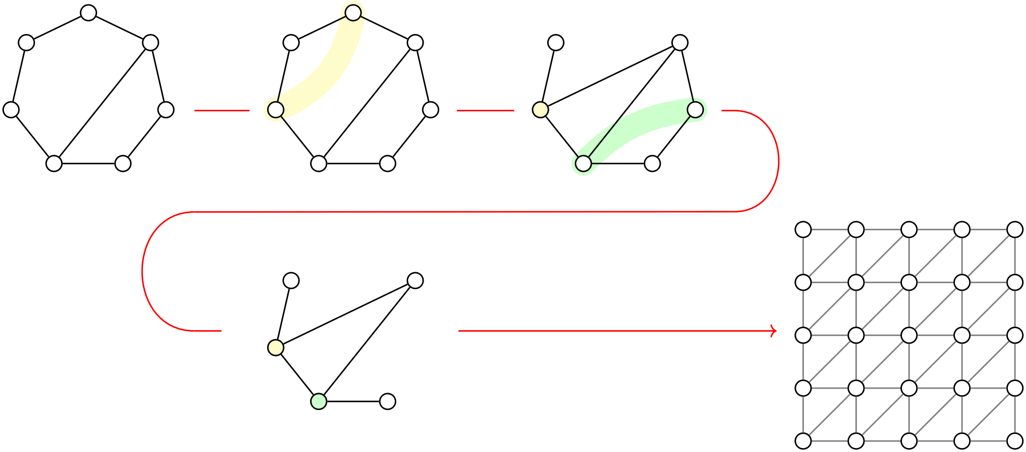
Example of a homomorphism $F \longrightarrow G$.



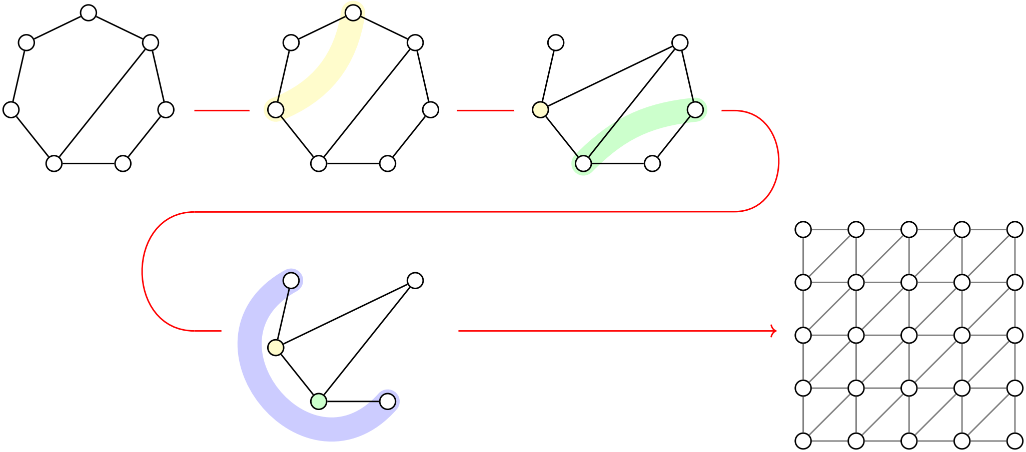
Example of a homomorphism $F \longrightarrow G$.



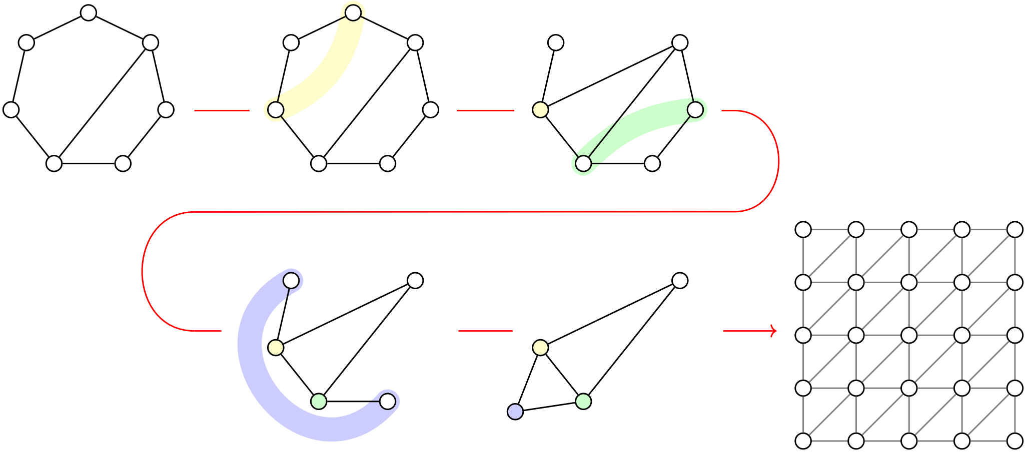
Example of a homomorphism $F \longrightarrow G$.



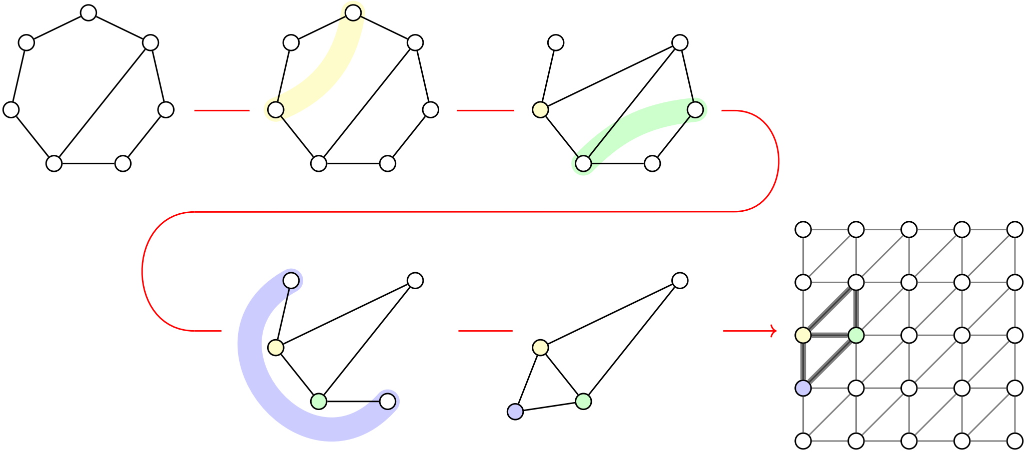
Example of a homomorphism $F \longrightarrow G$.



Example of a homomorphism $F \longrightarrow G$.



Example of a homomorphism $F \longrightarrow G$.



Homomorphism Indistinguishability over $\mathcal{F}$

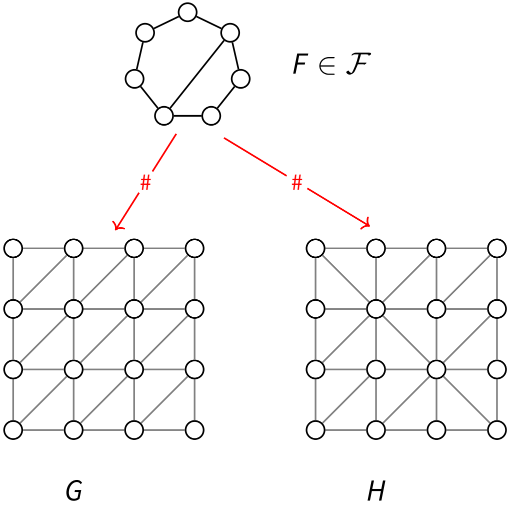
$\hom(F, G) \ldots$ the number of homomorphisms $F \longrightarrow G$.

Definition ($G \equiv_{\mathcal{F}} H$)

$\hom(F, G) = \hom(F, H) \quad \forall F \in \mathcal{F}$.

Problem (HomInd($\mathcal{F}$))

Input: *Graphs G and H.*
Decide: $G \equiv_{\mathcal{F}} H$.



Connections of HomInd($\mathcal{F}$) to Other Problems

Graph class $\mathcal{F}$ Problem $G \equiv_{\mathcal{F}} H$

See the HomInd Zoo: tseppelt.github.io/homind-database .

Connections of HomInd($\mathcal{F}$) to Other Problems

Graph class $\mathcal{F}$	Problem $G \equiv_{\mathcal{F}} H$	
All graphs	Graph Isomorphism	Lovász 1967

See the HomInd Zoo: tseppelt.github.io/homind-database .

Connections of HomInd($\mathcal{F}$) to Other Problems

Graph class $\mathcal{F}$	Problem $G \equiv_{\mathcal{F}} H$	
All graphs	Graph Isomorphism	Lovász 1967
Planar graphs	Quantum Isomorphism	Mančinska and Roberson 2020

See the HomInd Zoo: tseppelt.github.io/homind-database .

Connections of HomInd($\mathcal{F}$) to Other Problems

Graph class $\mathcal{F}$	Problem $G \equiv_{\mathcal{F}} H$	
All graphs	Graph Isomorphism	Lovász 1967
Planar graphs	Quantum Isomorphism	Mančinska and Roberson 2020
Treewidth $\leq k$	k-dim Weisfeiler–Leman Test	Dvořák 2010
	$(k+1)$-variable FOL with counting	Cai, Fürer, and Immerman 1992

See the HomInd Zoo: tseppelt.github.io/homind-database .

Connections of HomInd($\mathcal{F}$) to Other Problems

Graph class $\mathcal{F}$	Problem $G \equiv_{\mathcal{F}} H$	
All graphs	Graph Isomorphism	Lovász 1967
Planar graphs	Quantum Isomorphism	Mančinska and Roberson 2020
Treewidth $\leq k$	k-dim Weisfeiler–Leman Test	Dvořák 2010
	$(k+1)$-variable FOL with counting	Cai, Fürer, and Immerman 1992
Treedepth $\leq d$	d-quantifier FOL with counting	Grohe 2020

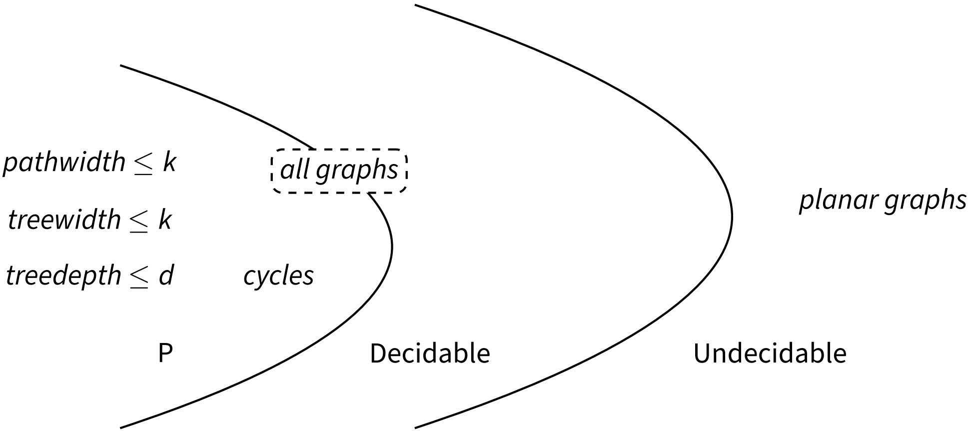
See the HomInd Zoo: tseppelt.github.io/homind-database .

Connections of HomInd($\mathcal{F}$) to Other Problems

Graph class $\mathcal{F}$	Problem $G \equiv_{\mathcal{F}} H$	
All graphs	Graph Isomorphism	Lovász 1967
Planar graphs	Quantum Isomorphism	Mančinska and Roberson 2020
Treewidth $\leq k$	k-dim Weisfeiler–Leman Test	Dvořák 2010
	$(k+1)$-variable FOL with counting	Cai, Fürer, and Immerman 1992
Treedepth $\leq d$	d-quantifier FOL with counting	Grohe 2020
Cycles	Cospectrality of adjacency matrices	folklore

See the HomInd Zoo: tseppelt.github.io/homind-database .

Complexity of $\textsc{HomInd}(\mathcal{F})$



Equivalent Views on Recognizing Graph Properties

$\textsc{HomInd}(\mathcal{T}_k)$

Equivalent Views on Recognizing Graph Properties

$\textsc{HomInd}(\mathcal{T}_k)$

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

$\textsc{HomInd}(\mathcal{T}_k)$

$\Vert$

Dvořák 2010

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

k-WL

k-dimensional
Weisfeiler–Leman
refinement

$\textsc{HomInd}(\mathcal{T}_k)$

$\|$

Dvořák 2010

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

k-WL

k-dimensional
Weisfeiler–Leman
refinement

Cai, Fürer, and Immerman 1992

$\textsc{HomInd}(\mathcal{T}_k)$

Dvořák 2010

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

k-WL

k-dimensional
Weisfeiler–Leman
refinement

Cai, Fürer, and Immerman 1992

$\textsc{HomInd}(\mathcal{T}_k)$

$\textsc{HomInd}(\mathcal{P}_k)$

Dvořák 2010

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

k-WL

k-dimensional
Weisfeiler–Leman
refinement

Cai, Fürer, and Immerman 1992

$\textsc{HomInd}(\mathcal{T}_k)$

Dvořák 2010

$\textsc{HomInd}(\mathcal{P}_k)$

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

$\mathsf{C}_{\mathsf{R}}^{k+1}$

restricted-conjunction
$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

k-WL

k-dimensional
Weisfeiler–Leman
refinement

Cai, Fürer, and Immerman 1992

$\textsc{HomInd}(\mathcal{T}_k)$

Dvořák 2010

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

$\textsc{HomInd}(\mathcal{P}_k)$

Montacute and Shah 2024

$\mathsf{C}^{k+1}_{\mathsf{R}}$

restricted-conjunction
$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

k-WL

k-dimensional
Weisfeiler–Leman
refinement

k-SW

k-dimensional
simplicial walk
refinement

$\textsc{HomInd}(\mathcal{T}_k)$

Cai, Fürer, and Immerman 1992

$\textsc{HomInd}(\mathcal{P}_k)$

Dvořák 2010

Montacute and Shah 2024

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

$\mathsf{C}_{\mathsf{R}}^{k+1}$

restricted-conjunction
$(k+1)$-variable **first-order logic**
with counting quantifiers

Equivalent Views on Recognizing Graph Properties

k-WL

k-dimensional
Weisfeiler–Leman
refinement

$\textsc{HomInd}(\mathcal{T}_k)$

Dvořák 2010

Cai, Fürer, and Immerman 1992

C^{k+1}

$(k+1)$-variable **first-order logic**
with counting quantifiers

k-SW

k-dimensional
simplicial walk
refinement

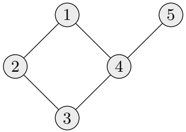
$\textsc{HomInd}(\mathcal{P}_k)$

Montacute and Shah 2024

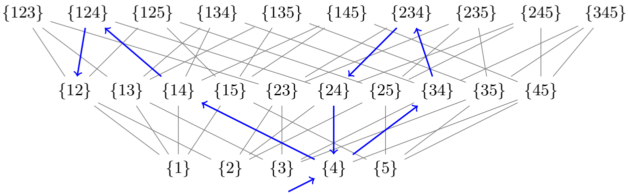
$\mathsf{C}^{k+1}_{\mathsf{R}}$

restricted-conjunction
$(k+1)$-variable **first-order logic**
with counting quantifiers

Simplicial Walks



Graph G

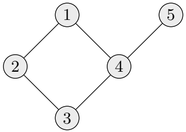


Hasse graph of 2-skeleton $\mathcal{H}(K(2))$ of $K = \mathcal{P}(V(G)) \setminus \{\emptyset\}$

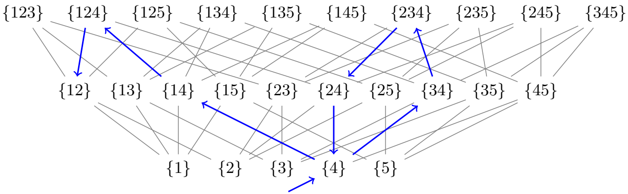
Notation:

- $\mathcal{H}(K(k))$ is the Hasse graph of the k-skeleton $K(k)$ of a simplicial complex K.
- $\dim \sigma = |\sigma| - 1$ for a simplex σ in $K(k)$.

Simplicial Walks



Graph G



Hasse graph of 2-skeleton $\mathcal{H}(K(2))$ of $K = \mathcal{P}(V(G)) \setminus \{\emptyset\}$

Notation:

- $\mathcal{H}(K(k))$ is the Hasse graph of the k-skeleton $K(k)$ of a simplicial complex K.
- $\dim \sigma = |\sigma| - 1$ for a simplex σ in $K(k)$.

Definition (Simplicial walk)

Let G be a graph, $k \geq 1$, and $K = \mathcal{P}(V(G)) \setminus \{\emptyset\}$.
Then a *k-simplicial walk in G of length t* is a walk $\boldsymbol{\sigma} = (\sigma_0, \sigma_1, \ldots, \sigma_t)$ in the graph $\mathcal{H}(K(k))$ such that $\dim \sigma_0 = 0$.

Colored Simplicial Walks

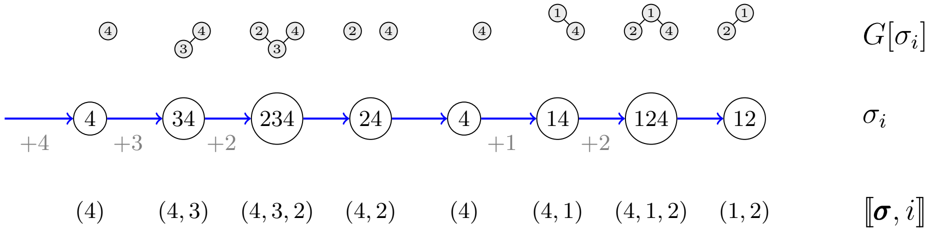


Figure: A simplicial walk $\boldsymbol{\sigma}$ in G with each σ_i ordered as $[\![\boldsymbol{\sigma}, i]\!]$.

Definition (Colored simplicial walk)

Let $k \geq 1$, and let $\boldsymbol{\sigma}$ be a k-simplicial walk of length t in a graph G, then a *color of* $\boldsymbol{\sigma}$ is $\boldsymbol{c} = (c_1, c_2, \ldots, c_t)$ with $c_i = \chi_k(G, [\![\boldsymbol{\sigma}, i]\!])$, where χ_k is given for each ℓ-tuple $\boldsymbol{u}$ in $V(G)^\ell$ by

$$\chi_k(G, \boldsymbol{u}) := \chi_{k\text{-wl}}^{(k+\ell-1)}(G, u_1, u_2, \ldots, u_\ell, \ldots, u_\ell), \qquad 1 \leq \ell \leq k+1. \tag{1}$$

From Simplicial Walks To Multiplicity Automata

Theorem

For two graphs G and H, the following conditions are equivalent:

1. *G and H have the same numbers of colored k-simplicial walks.*
2. *G and H have the same homomorphism counts over graphs in $\mathcal{P}_k$.*
3. *G and H satisfy the same formulas definable in logic $\mathsf{C}_{\mathsf{R}}^{k+1}$.*

From Simplicial Walks To Multiplicity Automata

Theorem

For two graphs G and H, the following conditions are equivalent:

1. *G and H have the same numbers of colored k-simplicial walks.*
2. *G and H have the same homomorphism counts over graphs in* $\mathcal{P}_k$.
3. *G and H satisfy the same formulas definable in logic* $\mathsf{C}_{\mathsf{R}}^{k+1}$.

- *Simplicial walk* refinement (SW) aggregates colored simplicial walks

From Simplicial Walks To Multiplicity Automata

Theorem

For two graphs G and H, the following conditions are equivalent:

1. *G and H have the same numbers of colored k-simplicial walks.*
2. *G and H have the same homomorphism counts over graphs in* $\mathcal{P}_k$.
3. *G and H satisfy the same formulas definable in logic* $\mathsf{C}_{\mathsf{R}}^{k+1}$.

- *Simplicial walk* refinement (SW) aggregates colored simplicial walks
  - similar to Weisfeiler–Leman refinement,
  - unlike WL multisets, SW multisets can be exponentially large

From Simplicial Walks To Multiplicity Automata

Theorem

For two graphs G and H, the following conditions are equivalent:

1. *G and H have the same numbers of colored k-simplicial walks.*
2. *G and H have the same homomorphism counts over graphs in $\mathcal{P}_k$.*
3. *G and H satisfy the same formulas definable in logic $\mathsf{C}_{\mathsf{R}}^{k+1}$.*

- *Simplicial walk* refinement (SW) aggregates colored simplicial walks
  - similar to Weisfeiler–Leman refinement,
  - unlike WL multisets, SW multisets can be exponentially large
- Solution: use *multiplicity automata* to represent SW.

Multiplicity Automata (MAs)

Definition

A multiplicity automaton is $\mathcal{A} = (S, \Sigma, M, \alpha, \eta)$

- states S, and alphabet Σ (both finite)
- $\alpha \in \mathbb{Q}^{1 \times S}$ (initial v.), $\eta \in \mathbb{Q}^{S}$ (final v.),
- $M\colon \Sigma \to \mathbb{Q}^{S \times S}$ (transition multiplicities).

Multiplicity Automata (MAs)

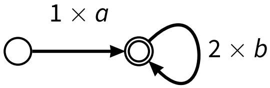


Figure: MA $\mathcal{A}_m$. E.g., $[\![\mathcal{A}_m]\!](abbb) = 2^3$.

Definition

A multiplicity automaton is $\mathcal{A} = (S, \Sigma, M, \alpha, \eta)$

- states S, and alphabet Σ (both finite)
- $\alpha \in \mathbb{Q}^{1\times S}$ (initial v.), $\eta \in \mathbb{Q}^S$ (final v.),
- $M : \Sigma \to \mathbb{Q}^{S\times S}$ (transition multiplicities).

Definition (Semantics)

$\mathcal{A}$ *recognises* the rational series $[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{Q}$ by

$$[\![\mathcal{A}]\!](\sigma_1\sigma_2\cdots\sigma_t) = \alpha M(\sigma_1)M(\sigma_2)\cdots M(\sigma_t)\eta.$$

for each word $\sigma_1\sigma_2\cdots\sigma_t \in \Sigma^*$.

Multiplicity Automata (MAs)

Definition

A multiplicity automaton is $\mathcal{A} = (S, \Sigma, M, \alpha, \eta)$

- states S, and alphabet Σ (both finite)
- $\alpha \in \mathbb{Q}^{1\times S}$ (initial v.), $\eta \in \mathbb{Q}^{S}$ (final v.),
- $M \colon \Sigma \to \mathbb{Q}^{S\times S}$ (transition multiplicities).

Definition (Semantics)

$\mathcal{A}$ *recognises* the rational series $[\![\mathcal{A}]\!] \colon \Sigma^* \to \mathbb{Q}$ by

$$[\![\mathcal{A}]\!](\sigma_1\sigma_2\cdots\sigma_t) = \alpha M(\sigma_1)M(\sigma_2)\cdots M(\sigma_t)\eta.$$

for each word $\sigma_1\sigma_2\cdots\sigma_t \in \Sigma^*$.

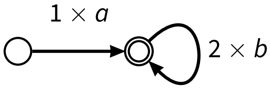


Figure: MA $\mathcal{A}_m$. E.g., $[\![\mathcal{A}_m]\!](abbb) = 2^3$.

Definition (MA Equivalence)

Two MAs $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if

$$[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!].$$

Definition (MA Minimality)

An MA $\mathcal{A}$ is *minimal* if there is no other equivalent MA with fewer states.

Multiplicity Involution Automata

- Minimization of MA $\mathcal{A}$ is computable in polynomial time,

Multiplicity Involution Automata

- Minimization of MA $\mathcal{A}$ is computable in polynomial time,
  - but a minimal $\widehat{\mathcal{A}}^{(B)}$ is unique up to *change of basis* B.

Multiplicity Involution Automata

- Minimization of MA $\mathcal{A}$ is computable in polynomial time,
  - but a minimal $\widehat{\mathcal{A}}^{(B)}$ is unique up to *change of basis* B.
- Solution: endow MA with involution.
  - Involution is a unary operation $^* : \Sigma \to \Sigma$ satisfying $(a^*)^* = a$ for all a in Σ.

Multiplicity Involution Automata

- Minimization of MA $\mathcal{A}$ is computable in polynomial time,
  - but a minimal $\widehat{\mathcal{A}}^{(B)}$ is unique up to *change of basis* B.
- Solution: endow MA with involution.
  - Involution is a unary operation $^* : \Sigma \to \Sigma$ satisfying $(a^*)^* = a$ for all a in Σ.

Definition

A *multiplicity involution automaton* $\mathcal{A} = (Q, \Sigma, M, \alpha, \eta)$ satisfying the following conditions:

- The initial and final vectors are adjoint, i.e. $\alpha^* = \eta$.
- There is an *involution* $^* : \Sigma \to \Sigma$, such that $M(a^*) = M(a)^*$ for all a in Σ.

Multiplicity Involution Automata

- Minimization of MA $\mathcal{A}$ is computable in polynomial time,
  - but a minimal $\widehat{\mathcal{A}}^{(B)}$ is unique up to *change of basis B*.
- Solution: endow MA with involution.
  - Involution is a unary operation $^* : \Sigma \to \Sigma$ satisfying $(a^*)^* = a$ for all a in Σ.

Definition

A *multiplicity involution automaton* $\mathcal{A} = (Q, \Sigma, M, \alpha, \eta)$ satisfying the following conditions:

- The initial and final vectors are adjoint, i.e. $\alpha^* = \eta$.
- There is an *involution* $^* : \Sigma \to \Sigma$, such that $M(a^*) = M(a)^*$ for all a in Σ.

Theorem

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two multiplicity involution automata over the alphabet Σ, then

$$[\![\mathcal{A}_1]\!] = [\![\mathcal{A}_2]\!] \quad \text{if and only if} \quad \widehat{\mathcal{A}}_1 = \widehat{\mathcal{A}}_2.$$

Canonizations Equivalent to Numbers of Simplicial Walks

Theorem

For $h, k \geq 1$, $n \geq 0$, there exists a canonical representation $\mathcal{I}_{\mathsf{sw}}$ of size $\mathrm{O}(kn^{2k})$, computable in time $\mathrm{O}(kn^{\mathbf{3}k})$, such that for every graph G on n vertices, $\mathcal{I}_{\mathsf{sw}}(G, k, h)$ recognizes exactly the same graph properties as numbers of h-colored k-simplicial walks in G.

Canonizations Equivalent to Numbers of Simplicial Walks

Theorem

For $h, k \geq 1$, $n \geq 0$, there exists a canonical representation $\mathcal{I}_{\mathsf{sw}}$ of size $\mathrm{O}(kn^{2k})$, computable in time $\mathrm{O}(kn^{\mathbf{3}k})$, such that for every graph G on n vertices, $\mathcal{I}_{\mathsf{sw}}(G, k, h)$ recognizes exactly the same graph properties as numbers of h-colored k-simplicial walks in G.

This improves upon the time $\mathrm{O}(k^c n^{7k+7})$ for $n^{k+1} \log n \geq e^{2000}$ shown by Seppelt 2024.

Corollary

The problem HOMIND$(\mathcal{P}_k)$ *for graphs on n vertices is decidable in time $\mathrm{O}(kn^{\mathbf{3}k})$.*

References I

- Cai, Jin-Yi, Martin Fürer, and Neil Immerman (1992). "An optimal lower bound on the number of variables for graph identification". In: *Combinatorica* 12.4, pp. 389–410. ISSN: 1439-6912. DOI: 10.1007/BF01305232. URL: https://doi.org/10.1007/BF01305232.
- Dvořák, Zdeněk (2010). "On recognizing graphs by numbers of homomorphisms". In: *Journal of Graph Theory* 64.4, pp. 330–342. DOI: 10.1002/jgt.20461. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.20461.
- Grohe, Martin (2020). "Counting Bounded Tree Depth Homomorphisms". In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '20. New York, NY, USA: Association for Computing Machinery, pp. 507–520. ISBN: 978-1-4503-7104-9. DOI: 10.1145/3373718.3394739.
- Lovász, L. (1967). "Operations with structures". In: *Acta Mathematica Academiae Scientiarum Hungarica* 18.3, pp. 321–328. ISSN: 1588-2632. DOI: 10.1007/BF02280291. URL: https://doi.org/10.1007/BF02280291.

References II

Mančinska, Laura and David E. Roberson (2020). “Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs”. In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 661–672. DOI: 10.1109/FOCS46700.2020.00067.

Montacute, Yoàv and Nihil Shah (May 2024). “The Pebble-Relation Comonad in Finite Model Theory”. In: *Logical Methods in Computer Science* Volume 20, Issue 2, 9. ISSN: 1860-5974. DOI: 10.46298/lmcs-20(2:9)2024. URL: https://lmcs.episciences.org/10884.

Seppelt, Tim (2024). “An Algorithmic Meta Theorem for Homomorphism Indistinguishability”. In: *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*. Vol. 306. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 82:1–82:19. ISBN: 978-3-95977-335-5. DOI: 10.4230/LIPIcs.MFCS.2024.82. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.MFCS.2024.82.