Recent Advances in Homomorphism Indistinguishability

Tim Seppelt

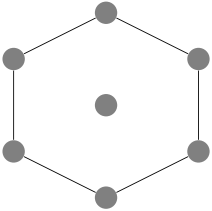
Joint work with Gaurav Rattan

Structure Meets Power 2022

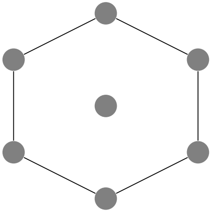
Research Training Group – Uncertainty and Randomness in Algorithms, Verification, and Logic
UnRAVeL
RWTH AACHEN UNIVERSITY

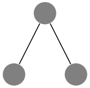
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Homomorphism Indistinguishability

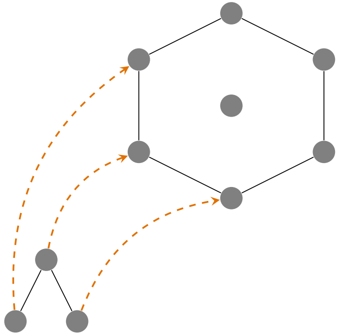


Homomorphism Indistinguishability

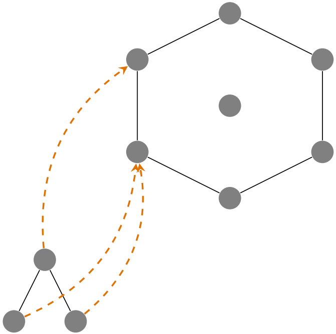




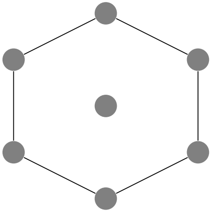
Homomorphism Indistinguishability

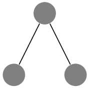


Homomorphism Indistinguishability



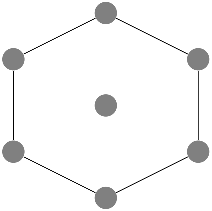
Homomorphism Indistinguishability

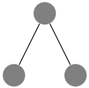




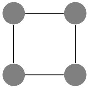
24

Homomorphism Indistinguishability



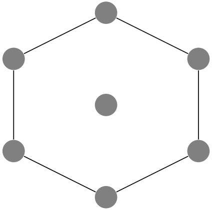
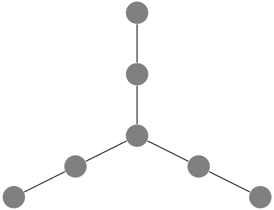
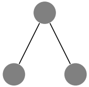
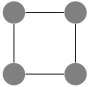


24

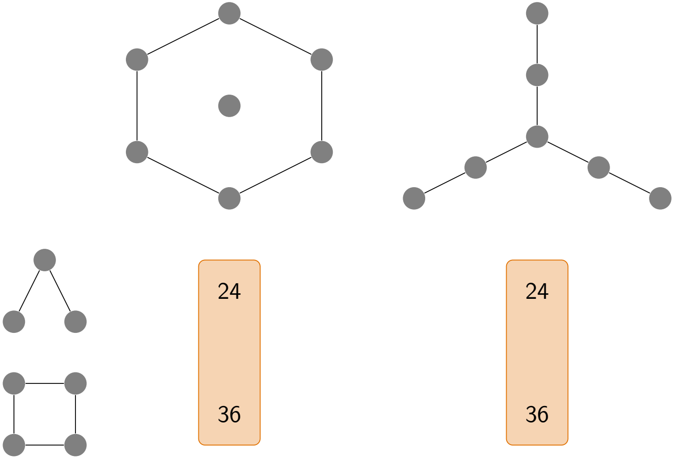


36

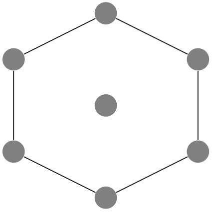
Homomorphism Indistinguishability

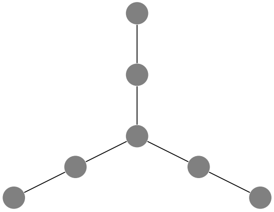
		
	24	24
	36	36

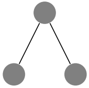
Homomorphism Indistinguishability

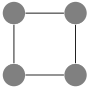


Homomorphism Indistinguishability

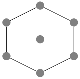
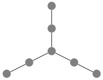








(path)	24	24
(square)	36	36

The graphs  and  are **homomorphism indistinguishable** over $\{$  ,  $\}$.

Why?

Logical Equivalences ⟵ **Homomorphism Indistinguishability** ⟶ **Matrix Equations** X s.t. $X\boldsymbol{A}_G = \boldsymbol{A}_H X$

Why?

Logical Equivalences	←	**Homomorphism Indistinguishability**	→	**Matrix Equations** X s.t. $X\boldsymbol{A}_G = \boldsymbol{A}_H X$
FO with counting C_2	←	Trees	→	Fractional Isomorphism

Why?

Logical Equivalences	⟵	**Homomorphism Indistinguishability**	⟶	**Matrix Equations** X s.t. $X\boldsymbol{A}_G = \boldsymbol{A}_H X$
FO with counting C_2	⟵	Trees	⟶	Fractional Isomorphism
		Planar graphs	⟶	Quantum Isomorphism

Why?

Logical Equivalences		Homomorphism Indistinguishability		Matrix Equations
Logical Equivalences	⟵	**Homomorphism Indistinguishability**	⟶	**Matrix Equations** X s.t. $X\boldsymbol{A}_G = \boldsymbol{A}_H X$
FO with counting C_2	⟵	Trees	⟶	Fractional Isomorphism
		Planar graphs	⟶	Quantum Isomorphism
FO with counting $\mathsf{C}_k \cap \mathsf{C}^q$	⟵	k-pebble forest covers of depth q	⟶	Novel System

Techniques

Techniques

Logical Equivalences ⟵ **Homomorphism Indistinguishability** ⟶ Matrix Equations X s.t. $X\boldsymbol{A}_G = \boldsymbol{A}_H X$

Techniques

Logical Equivalences $\longleftarrow$ **Homomorphism Indistinguishability** $\longrightarrow$ Matrix Equations X s.t. $X\boldsymbol{A}_G = \boldsymbol{A}_H X$

Category Theory

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & \mathfrak{C}A \\ \Big\downarrow{\scriptstyle \alpha} & & \Big\downarrow{\scriptstyle \mathfrak{C}\alpha} \\ \mathfrak{C}A & \xrightarrow{\delta_A} & \mathfrak{C}\mathfrak{C}A \end{array}$$

Dawar et al. (2021); Montacute and Shah (2021); Abramsky et al. (2022); Reggio (2021); etc.

Techniques

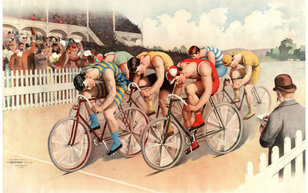
Logical Equivalences ⟵ **Homomorphism Indistinguishability** ⟶ Matrix Equations X s.t. $X\boldsymbol{A}_G = \boldsymbol{A}_H X$

Category Theory

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & \mathfrak{C}A \\ \downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle \mathfrak{C}\alpha} \\ \mathfrak{C}A & \xrightarrow{\delta_A} & \mathfrak{C}\mathfrak{C}A \end{array}$$

Dawar et al. (2021); Montacute and Shah (2021); Abramsky et al. (2022); Reggio (2021); etc.

Algebra and Representation Theory

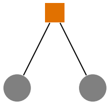


Mančinska and Roberson (2020); Grohe et al. (2022); Rattan and Seppelt (2022); etc.

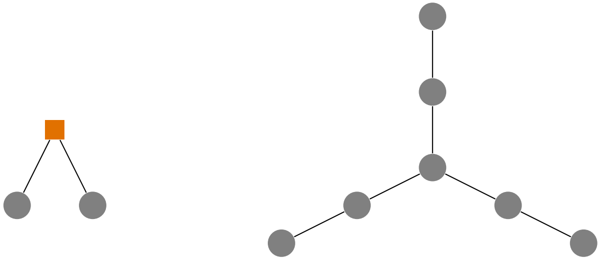
The Recipe for Matrix Equations

1. Construct family $\mathcal{F}$ of labelled graphs
2. Define suitable operations
3. Prove that $\mathcal{F}$ is finitely generated under operations
4. Define representation and recover system of equations

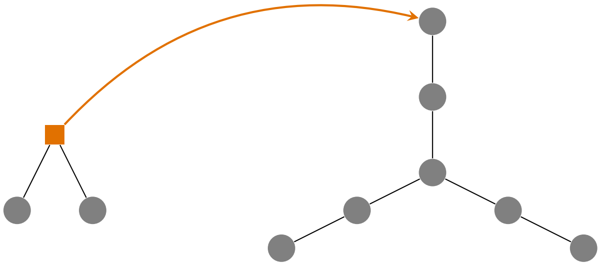
Labelled Graphs and Homomorphism Tensors



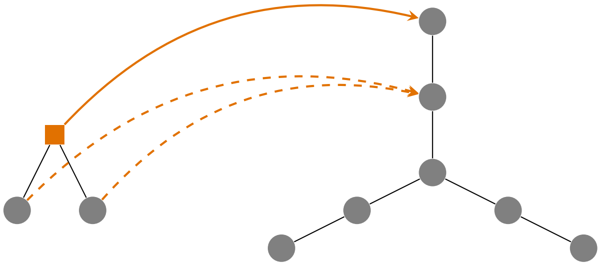
Labelled Graphs and Homomorphism Tensors



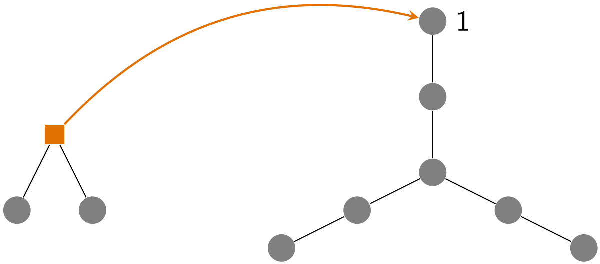
Labelled Graphs and Homomorphism Tensors



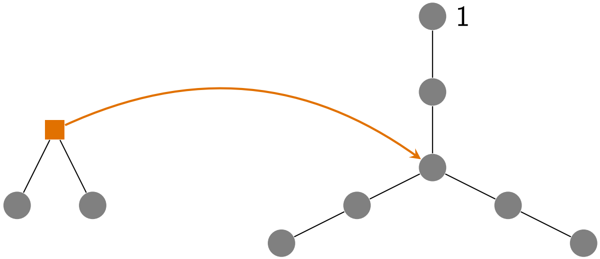
Labelled Graphs and Homomorphism Tensors



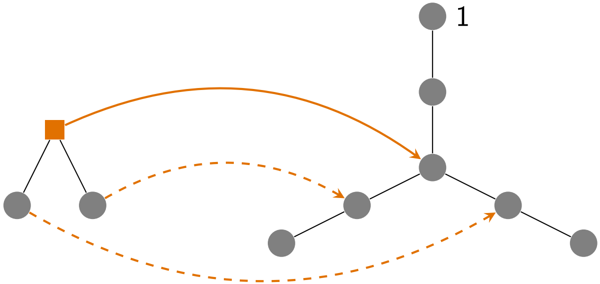
Labelled Graphs and Homomorphism Tensors



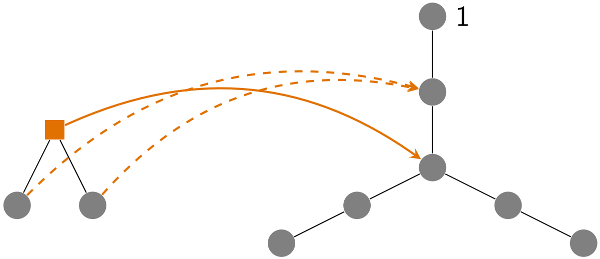
Labelled Graphs and Homomorphism Tensors



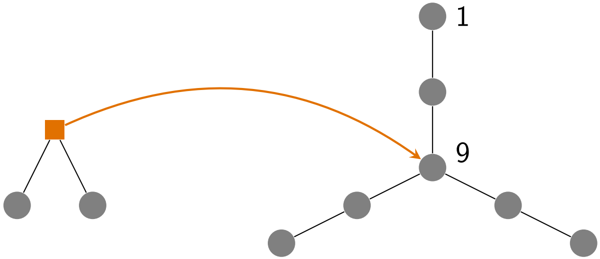
Labelled Graphs and Homomorphism Tensors



Labelled Graphs and Homomorphism Tensors

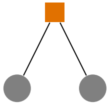


Labelled Graphs and Homomorphism Tensors

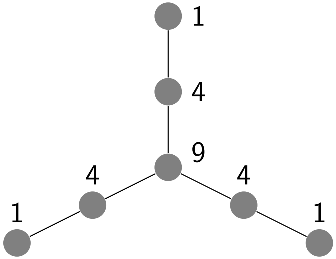


Labelled Graphs and Homomorphism Tensors

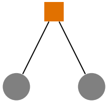
$$\mathcal{F} \longrightarrow \mathbb{C}^{V(G)}$$



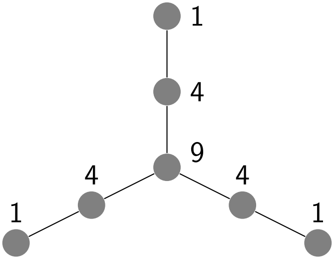
$\mapsto$



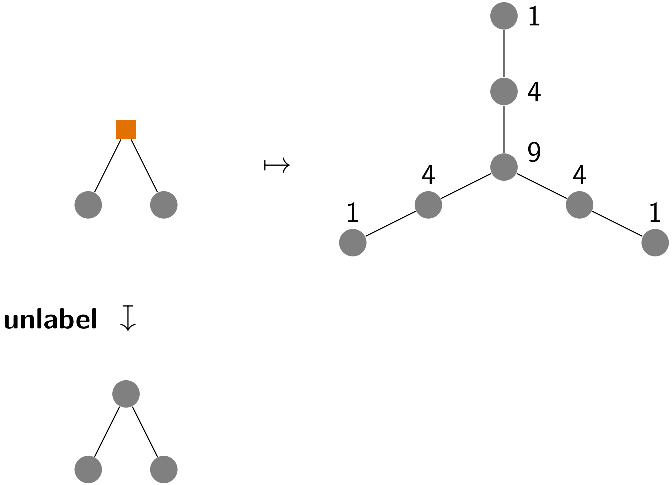
Combinatorial and Algebraic Operations: Unlabelling and Sum-of-Entries



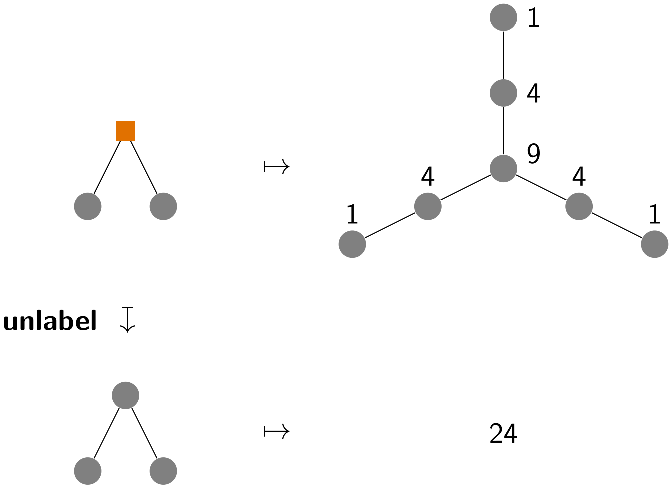
$\mapsto$



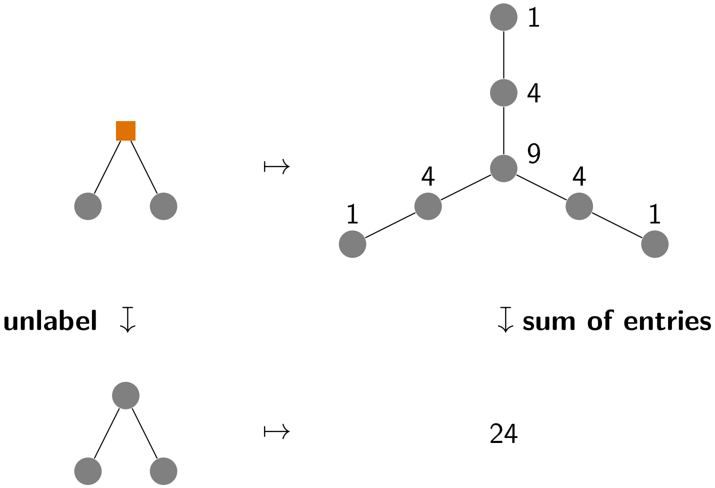
Combinatorial and Algebraic Operations: Unlabelling and Sum-of-Entries



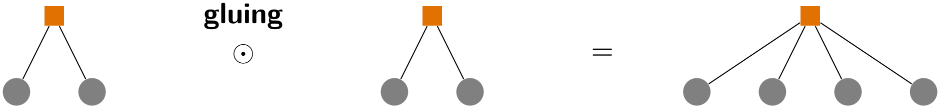
Combinatorial and Algebraic Operations: Unlabelling and Sum-of-Entries



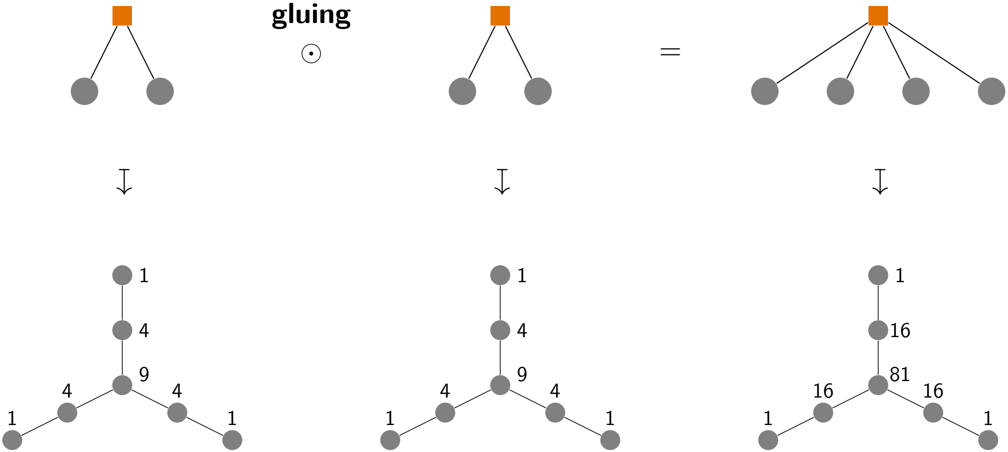
Combinatorial and Algebraic Operations: Unlabelling and Sum-of-Entries



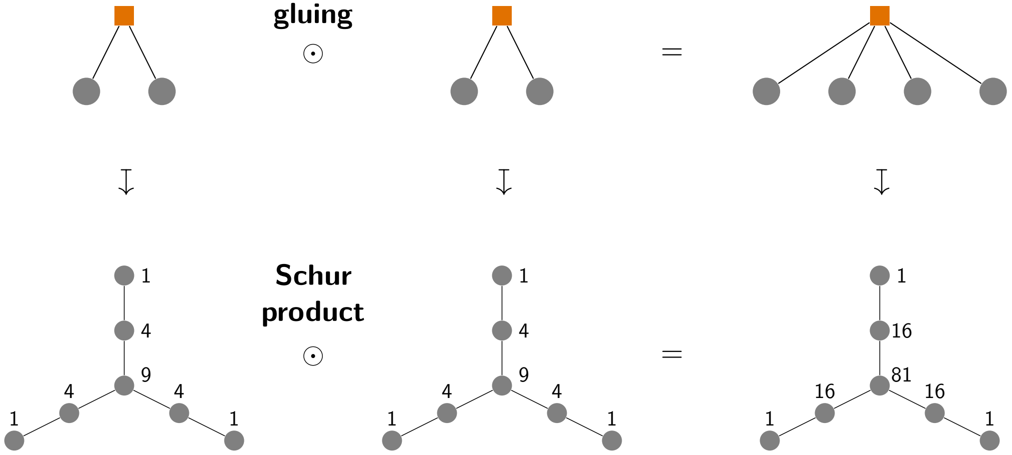
Combinatorial and Algebraic Operations: Gluing and Schur Product



Combinatorial and Algebraic Operations: Gluing and Schur Product



Combinatorial and Algebraic Operations: Gluing and Schur Product



Example: Trees

1. Construct family $\mathcal{F}$ of labelled graphs
2. Define suitable operations
3. Prove that $\mathcal{F}$ is finitely generated under operations
4. Define representation and recover system of equations

Example: Trees

1. Construct family $\mathcal{F}$ of labelled graphs
   - trees with one label at arbitrary vertex
2. Define suitable operations

3. Prove that $\mathcal{F}$ is finitely generated under operations

4. Define representation and recover system of equations

Example: Trees

1. Construct family $\mathcal{F}$ of labelled graphs
   - trees with one label at arbitrary vertex
2. Define suitable operations
   - gluing, attaching edges
3. Prove that $\mathcal{F}$ is finitely generated under operations

4. Define representation and recover system of equations

Example: Trees

1. Construct family $\mathcal{F}$ of labelled graphs
   - trees with one label at arbitrary vertex
2. Define suitable operations
   - gluing, attaching edges
3. Prove that $\mathcal{F}$ is finitely generated under operations
   - all trees are generated by these operations starting from a single vertex
4. Define representation and recover system of equations

Example: Trees

1. Construct family $\mathcal{F}$ of labelled graphs
   - trees with one label at arbitrary vertex
2. Define suitable operations
   - gluing, attaching edges
3. Prove that $\mathcal{F}$ is finitely generated under operations
   - all trees are generated by these operations starting from a single vertex
4. Define representation and recover system of equations
   - homomorphism tensor representation
   - some linear algebra and representation theory developed in Grohe et al. (2022)
   - Fractional Isomorphism

Glue(ten) intolerance

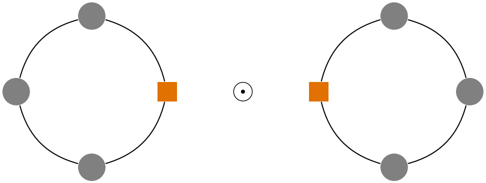
Can the recipe be used to cook up systems of equations for other graph classes in a generic fashion?

The ingredients labelling, operations, finite generation, and representation have to blend together well.

Glue(ten) intolerance

Can the recipe be used to cook up systems of equations for other graph classes in a generic fashion?

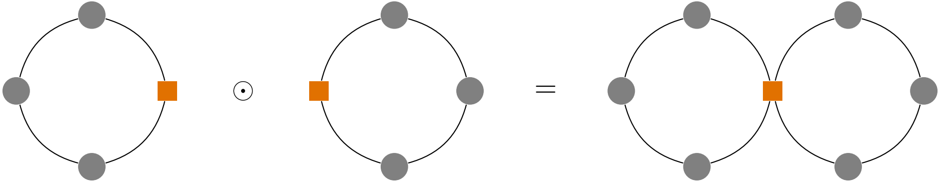
The ingredients labelling, operations, finite generation, and representation have to blend together well.



Glue(ten) intolerance

Can the recipe be used to cook up systems of equations for other graph classes in a generic fashion?

The ingredients labelling, operations, finite generation, and representation have to blend together well.



Modern Cuisine: New Recipes

Theorem (Abramsky et al. (2022))

For every "natural" graph class $\mathcal{F}$ there exists a comonad $\mathfrak{C}$ on Graph such that

$$F \in \mathcal{F} \text{ if and only if } F \text{ has a } \mathfrak{C}\text{-coalgebra.}$$

Modern Cuisine: New Recipes

Theorem (Abramsky et al. (2022))

For every "natural" graph class $\mathcal{F}$ there exists a comonad $\mathfrak{C}$ on Graph such that

$$F \in \mathcal{F} \text{ if and only if } F \text{ has a } \mathfrak{C}\text{-coalgebra.}$$

1. Construct family $\mathcal{F}$ of labelled graphs
2. Define suitable operations
3. Prove that $\mathcal{F}$ is finitely generated under operations
4. Define representation and recover system of equations

Modern Cuisine: New Recipes

Theorem (Abramsky et al. (2022))

For every "natural" graph class $\mathcal{F}$ there exists a comonad $\mathfrak{C}$ on Graph such that

$$F \in \mathcal{F} \text{ if and only if } F \text{ has a } \mathfrak{C}\text{-coalgebra.}$$

1. Construct family $\mathcal{F}$ of labelled graphs
   - labelled $\mathfrak{C}$-coalgebras, i.e. morphisms $(L, \lambda) \to (A, \alpha)$.
2. Define suitable operations
3. Prove that $\mathcal{F}$ is finitely generated under operations
4. Define representation and recover system of equations

Modern Cuisine: New Recipes

Theorem (Abramsky et al. (2022))

For every "natural" graph class $\mathcal{F}$ there exists a comonad $\mathfrak{C}$ on Graph such that

$$F \in \mathcal{F} \text{ if and only if } F \text{ has a } \mathfrak{C}\text{-coalgebra.}$$

1. Construct family $\mathcal{F}$ of labelled graphs
   - labelled $\mathfrak{C}$-coalgebras, i.e. morphisms $(L, \lambda) \to (A, \alpha)$.
2. Define suitable operations
   - finite colimits in the Eilenberg–Moore category of $\mathfrak{C}$, e.g. coproducts, pushouts
3. Prove that $\mathcal{F}$ is finitely generated under operations

4. Define representation and recover system of equations

Modern Cuisine: New Recipes

Theorem (Abramsky et al. (2022))

For every "natural" graph class $\mathcal{F}$ there exists a comonad $\mathfrak{C}$ on Graph such that

$$F \in \mathcal{F} \text{ if and only if } F \text{ has a } \mathfrak{C}\text{-coalgebra.}$$

1. Construct family $\mathcal{F}$ of labelled graphs
   - labelled $\mathfrak{C}$-coalgebras, i.e. morphisms $(L, \lambda) \to (A, \alpha)$.
2. Define suitable operations
   - finite colimits in the Eilenberg–Moore category of $\mathfrak{C}$, e.g. coproducts, pushouts
3. Prove that $\mathcal{F}$ is finitely generated under operations

4. Define representation and recover system of equations
   - homomorphism tensor representation $\otimes$ label representation
   - some linear algebra and representation theory...

Modern Cuisine: New Recipes

Theorem (Abramsky et al. (2022))

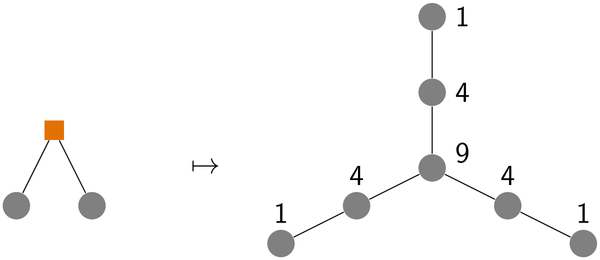
For every "natural" graph class $\mathcal{F}$ there exists a comonad $\mathfrak{C}$ on Graph such that

$$F \in \mathcal{F} \text{ if and only if } F \text{ has a } \mathfrak{C}\text{-coalgebra.}$$

1. Construct family $\mathcal{F}$ of labelled graphs
   - labelled $\mathfrak{C}$-coalgebras, i.e. morphisms $(L, \lambda) \to (A, \alpha)$.
2. Define suitable operations
   - finite colimits in the Eilenberg–Moore category of $\mathfrak{C}$, e.g. coproducts, pushouts
3. Prove that $\mathcal{F}$ is finitely generated under operations
   - **requires instance-specific arguments**
4. Define representation and recover system of equations
   - homomorphism tensor representation $\otimes$ label representation
   - some linear algebra and representation theory...

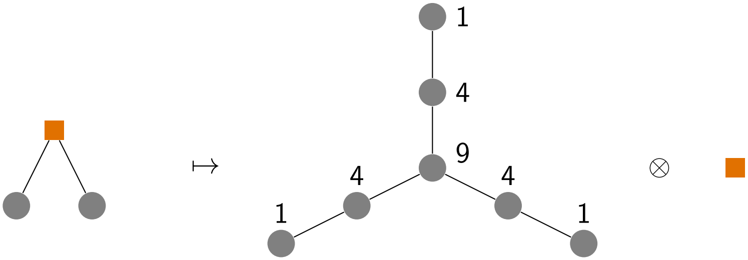
Label Representation

$$\mathcal{F} \longrightarrow \mathbb{C}^{V(G)}$$



Label Representation

$$\mathcal{F} \longrightarrow \mathbb{C}^{V(G)} \otimes \bigoplus_{(L,\lambda)} \mathbb{C}$$



Conclusion

- Comonads on Graph provide new recipes for systems of equations capturing homomorphism indistinguishability
  - labelling, operations, and representations are governed by comonad
  - finite generation yet requires instance-specific arguments

Conclusion

- Comonads on Graph provide new recipes for systems of equations capturing homomorphism indistinguishability
  - labelling, operations, and representations are governed by comonad
  - finite generation yet requires instance-specific arguments
- This yields novel systems of equations characterising $\mathsf{C}_k \cap \mathsf{C}^q$ equivalence
  - check out Rattan and Seppelt (2022) `arXiv:2103.02972`!

Bibliography I

Abramsky, S., Jakl, T., and Paine, T. (2022). Discrete density comonads and graph parameters. Number: arXiv:2205.06589 arXiv:2205.06589 [cs, math].

Dawar, A., Jakl, T., and Reggio, L. (2021). Lovász-type theorems and game comonads. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE.

Grohe, M., Rattan, G., and Seppelt, T. (2022). Homomorphism Tensors and Linear Equations. In Bojańczyk, M., Merelli, E., and Woodruff, D. P., editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Mančinska, L. and Roberson, D. E. (2020). Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 661–672. IEEE.

Montacute, Y. and Shah, N. (2021). The Pebble-Relation Comonad in Finite Model Theory. *arXiv:2110.08196 [cs, math]*. arXiv: 2110.08196.

Bibliography II

Rattan, G. and Seppelt, T. (2022). Weisfeiler–Leman and Graph Spectra. Number: arXiv:2103.02972 arXiv:2103.02972 [cs, math].

Reggio, L. (2021). Polyadic Sets and Homomorphism Counting. *arXiv:2110.11061 [cs, math]*. arXiv: 2110.11061.

Picture: "Bicycle race scene. A peloton of six cyclists crosses the finish line in front of a crowded grandstand, observed by a referee." (1895) by Calvert Lithographic Co., Detroit, Michigan, Public Domain, via Wikimedia Commons.

`https://commons.wikimedia.org/wiki/File:Bicycle_race_scene,_1895.jpg`