Homomorphism Counts, Logics & Query Algorithms

Balder ten Cate

ILLC, University of Amsterdam

TbiLLC 2023

Based on joint work with Victor Dalmau, Phokion Kolaitis, and Wei-Lin Wu

Many slides borrowed from a recent presentation by Phokion Kolaitis

What Mathematicians Do

Mathematicians study not objects, but relations between objects; the replacement of these objects by others is therefore indifferent to them, provided the relations do not change. The matter is for them unimportant, the form alone interests them.

What Mathematicians Do

Mathematicians study not objects, but relations between objects; the replacement of these objects by others is therefore indifferent to them, provided the relations do not change. The matter is for them unimportant, the form alone interests them.

Science and Hypothesis - 1902

Henri Poincaré

Isomorphism

- In mathematics, we study objects up to isomorphism.

Isomorphism

- In mathematics, we study objects up to isomorphism.

Definition:
Let $G = (V(G), E(G))$ and $H = (V(H), E(H))$ be two graphs.
An isomorphism from G to H is a function $h : V(G) \to V(H)$ such that

1. h is 1-1 and onto;
2. for all $u, v \in V(G)$,
$$(u, v) \in E(G) \text{ if and only if } (h(u), h(v)) \in E(H).$$

- Analogously for isomorphism between relational structures.

Algorithmic Aspects of Graph Isomorphism

The Graph Isomorphism Problem:
Given two finite graphs G and H, are they isomorphic?

Algorithmic Aspects of Graph Isomorphism

The Graph Isomorphism Problem:
Given two finite graphs G and H, are they isomorphic?

Open Problem:
What is the exact computational complexity of the Graph Isomorphism Problem?

Algorithmic Aspects of Graph Isomorphism

The Graph Isomorphism Problem

- is in NP;
- is unlikely to be NP-complete (else, PH collapses);
- is not known to be solvable in polynomial time;
- is solvable in quasi-polynomial time $2^{O((\log n)^c)}$, for some fixed $c > 0$ (Babai - 2017);
- is solvable in polynomial time on restricted classes of graphs:
  - planar graphs (Hopcroft and Wong - 1974);
  - graphs of bounded degree (Lucs - 1982);
  - ...

Algorithmic Aspects of Graph Isomorphism

The Graph Isomorphism Problem

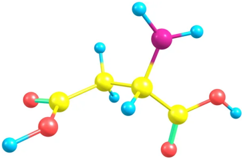
- is in NP;
- is unlikely to be NP-complete (else, PH collapses);
- is not known to be solvable in polynomial time;
- is solvable in quasi-polynomial time $2^{O((\log n)^c)}$, for some fixed $c > 0$ (Babai - 2017);
- is solvable in polynomial time on restricted classes of graphs:
  - planar graphs (Hopcroft and Wong - 1974);
  - graphs of bounded degree (Lucs - 1982);
  - ...

The Graph Isomorphism Disease

Brief digression: Graph Classification

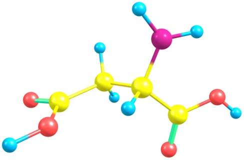
Brief digression: Graph Classification

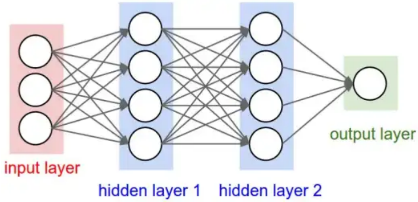
Suppose we wish to predict if a protein is an enzyme.



Brief digression: Graph Classification

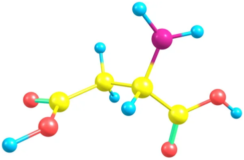
Suppose we wish to predict if a protein is an enzyme.

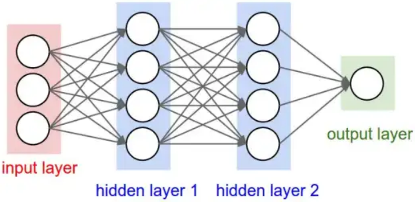




Brief digression: Graph Classification

Suppose we wish to predict if a protein is an enzyme.

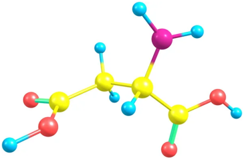


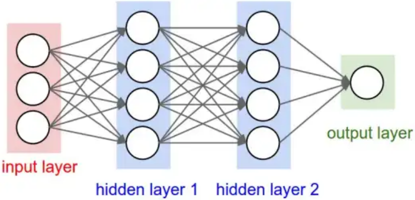


We need to encode a graph G as a "embedding vector" $v_G \in \mathbb{R}^k$.

Brief digression: Graph Classification

Suppose we wish to predict if a protein is an enzyme.





We need to encode a graph G as a "embedding vector" $v_G \in \mathbb{R}^k$.

We would like isomorphic graphs to be indistinguishable for the ML model.

Beyond Isomorphism

- In mathematics, we also study objects up to some other equivalence relation.

Examples:

1. Homeomorphism in Topology
2. Diffeomorphism in Differential Geometry
3. Logical Equivalence in First-Order Logic
4. ...

Beyond Isomorphism

- In mathematics, we also study objects up to some other equivalence relation.

Examples:

1. Homeomorphism in Topology
2. Diffeomorphism in Differential Geometry
3. Logical Equivalence in First-Order Logic
4. ...

- Here, we will focus on equivalence relations that arise from homomorphisms.

Homomorphism

Definition:

Let $G = (V(G), H(E))$ and $H = (V(H), E(H))$ be two graphs.
A homomorphism from G to H is a function $h : V(G) \to V(H)$ such that for all $u, v \in V(G)$,

$$\text{if } (u, v) \in E(G), \text{ then } (h(u), h(v)) \in E(H).$$

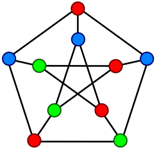
Homomorphism

Definition:
Let $G = (V(G), H(E))$ and $H = (V(H), E(H))$ be two graphs.
A homomorphism from G to H is a function $h : V(G) \to V(H)$ such that for all $u, v \in V(G)$,

$$\text{if } (u, v) \in E(G), \text{ then } (h(u), h(v)) \in E(H).$$

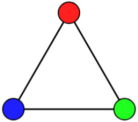
Example: Let G be a graph and let K_3 be the triangle graph.

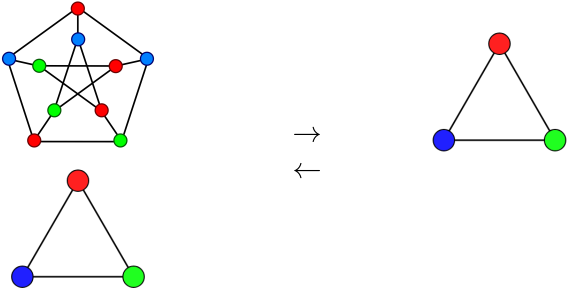
- There is a homomorphism from K_3 to G if and only if G contains a triangle.
- There is a homomorphism from G to K_3 if and only if G is 3-colorable.



$\rightarrow$

$\not\leftarrow$





Homomorphism Equivalence

Definition:

Two graphs G and H are homomorphically equivalent if there is a homomorphism from G and H, and a homomorphism from H and G.

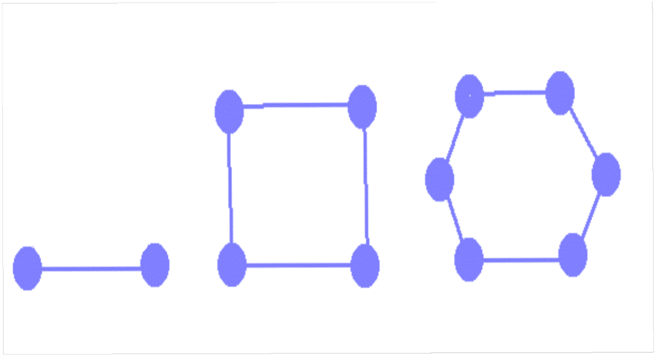
Homomorphism Equivalence

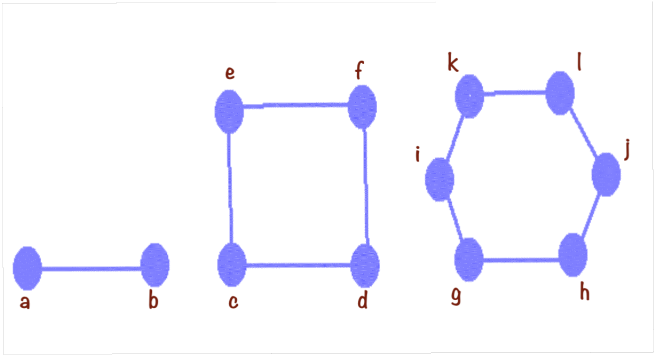
Definition:

Two graphs G and H are homomorphically equivalent if there is a homomorphism from G and H, and a homomorphism from H and G.

Example:

- If G and H are 2-colorable graphs with at least one edge each, then G and H are homomorphically equivalent.
- In particular, C_4 and C_6 are homomorphically equivalent (where C_n is the cycle with n nodes).




e
f
k
l
i
j
a
b
c
d
g
h

Complexity of Homomorphism Equivalence

Fact:

- Homomorphism Equivalence is an equivalence relation that is coarser than isomomorphism.

- Homomorphism Equivalence is NP-complete.

  Proof: Reduction from 3-Colorability:
  G is 3-colorable if and only if $G \oplus K_3$ is homomorphically equivalent to K_3.

Homomorphism Counts

Notation:

Let G and H be two graphs.

$\hom(G, H)$ = the number of homomorphisms from G to H.

Homomorphism Counts

Notation:

Let G and H be two graphs.

$\hom(G, H)$ = the number of homomorphisms from G to H.

Example:

Let G be a graph and let K_3 be the triangle graph.

- $\hom(K_3, G)$ = the number of triangles in G.
- $\hom(G, K_3)$ = the number of 3-colorings of G (times 6).

Two Interpretations of Homomorphism Counts

- Each H, gives rise to the constraint satisfaction problem

  $\mathrm{CSP}(H) = \{G : \text{there is a homomorphism from } G \text{ to } H\}$

  Thus,
  $\hom(G, H) = \#$ solutions of $\mathrm{CSP}(H)$ on input G.

Two Interpretations of Homomorphism Counts

- Each H, gives rise to the constraint satisfaction problem

  $\mathrm{CSP}(H) = \{G : \text{there is a homomorphism from } G \text{ to } H\}$

  Thus,
  $\hom(G, H) = \#$ solutions of $\mathrm{CSP}(H)$ on input G.

- Each G, gives rise to a conjunctive query Q^G

  Example: $Q^{K_3} : \exists x, y, z(E(x, y) \wedge E(y, z) \wedge E(z, x))$

  Thus,
  $\hom(G, H) = \#$ satisfying assignments from Q^G to input H.
  (this is the bag semantics of SQL)

Visualization of Homomorphism Counts

$\mathscr{G} = \{G_1, G_2, \dots\}$ is the class of all graphs (up to isomorphism).

$\hom(\cdot,\cdot)$	G_1	G_2	$\cdots$
G_1	$\hom(G_1, G_1)$	$\hom(G_1, G_2)$	$\cdots$
G_2	$\hom(G_2, G_1)$	$\hom(G_2, G_2)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

Left and Right Profiles

Definition: Let G be a graph.

- The left profile of G is the vector $\hom(\mathscr{G}, G) := (\hom(G_1, G), \hom(G_2, G), \dots)$.
- The right profile of G is the vector $\hom(G, \mathscr{G}) := (\hom(G, G_1), \hom(G, G_2), \dots)$.

$\hom(\cdot,\cdot)$	G_1	G_2	$\cdots$	G	$\cdots$
G_1	$\hom(G_1, G_1)$	$\hom(G_1, G_2)$	$\cdots$	$\hom(G_1, G)$	$\cdots$
G_2	$\hom(G_2, G_1)$	$\hom(G_2, G_2)$	$\cdots$	$\hom(G_2, G)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$
G	$\hom(G, G_1)$	$\hom(G, G_2)$	$\cdots$	$\hom(G, G)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$

Left/Right Profiles and Isomorphism

Lovász's Theorem (1967):

For all graphs G and H:

$$G \text{ and } H \text{ are isomorphic} \quad \text{iff} \quad \hom(\mathscr{G}, G) = \hom(\mathscr{G}, H).$$

- No two columns are equal.

Left/Right Profiles and Isomorphism

Lovász's Theorem (1967):

For all graphs G and H:

$$G \text{ and } H \text{ are isomorphic} \quad \text{iff} \quad \hom(\mathscr{G}, G) = \hom(\mathscr{G}, H).$$

- No two columns are equal.

Chaudhuri-Vardi Theorem (1993):

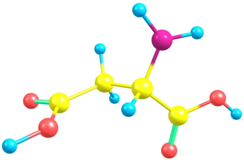
For all graphs G and H:

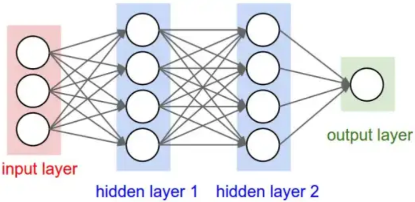
$$G \text{ and } H \text{ are isomorphic} \quad \text{iff} \quad \hom(G, \mathscr{G}) = \hom(H, \mathscr{G}).$$

- No two rows are equal.

Graph Classification Again

Suppose we wish to predict if a protein is an enzyme.





We need to encode a graph G as a embedding vector.

We would like isomorphic graphs to be indistinguishable for the ML model.

The left profile of G (i.e., $\mathrm{hom}(\mathscr{G}, G)$) provides an embedding vector which captures precisely the isomorphism type of G. The only problem is that it is infinite (and expensive to compute).

Restricted Profiles

Definition:

Let $\mathscr{F} = \{F_1, F_2, \ldots\}$ be a class of graphs and let G be a graph.

- The left profile of G restricted to $\mathscr{F}$ is the vector $\hom(\mathscr{F}, G) := (\hom(F_1, G), \hom(F_2, G), \ldots)$ (keep only the rows arising from graphs in $\mathscr{F}$).
- The right profile of G restricted to $\mathscr{F}$ is the vector $\hom(G, \mathscr{F}) := (\hom(G, F_1), \hom(G, F_2), \ldots)$ (keep only the columns arising from graphs in $\mathscr{F}$).

Equivalence Relations from Profiles

Each class $\mathscr{F}$ of graphs gives rise to two equivalence relations:

- $G \equiv^{L}_{\mathscr{F}} H$ if G and H have the same left profile restricted to $\mathscr{F}$.
- $G \equiv^{R}_{\mathscr{F}} H$ if G and H have the same right profile restricted to $\mathscr{F}$.

Note:
These equivalence relations are relaxations of isomorphism.

Equivalence Relations from Profiles

Each class $\mathscr{F}$ of graphs gives rise to two equivalence relations:

- $G \equiv^{L}_{\mathscr{F}} H$ if G and H have the same left profile restricted to $\mathscr{F}$.
- $G \equiv^{R}_{\mathscr{F}} H$ if G and H have the same right profile restricted to $\mathscr{F}$.

Note:

These equivalence relations are relaxations of isomorphism.

Question:

- Which equivalence relations $\equiv$ on graphs are of the form $\equiv^{L}_{\mathscr{F}}$ or of the form $\equiv^{R}_{\mathscr{F}}$?

Counting Logics with Finitely Many Variables

Definition: Let k be a positive integer.

- FO^k: First-order logic FO with at most k distinct variables.
- C^k: FO^k + Counting Quantifiers $(\exists i\ y)$, $i \geq 2$

$(\exists i\ y)\varphi(y)$: there are are at least i nodes y such that $\varphi(y)$ holds.

Counting Logics with Finitely Many Variables

Definition: Let k be a positive integer.

- FO^k: First-order logic FO with at most k distinct variables.
- C^k: FO^k + Counting Quantifiers $(\exists i\ y)$, $i \geq 2$

$(\exists i\ y)\varphi(y)$: there are are at least i nodes y such that $\varphi(y)$ holds.

Example: G is 7-regular is C^2-definable:

$$\forall x((\exists 7\ y)E(x,y) \land \neg(\exists 8\ y)E(x,y))$$

Counting Logics with Finitely Many Variables

Definition: Let k be a positive integer.

- FO^k: First-order logic FO with at most k distinct variables.
- C^k: FO^k + Counting Quantifiers $(\exists i\ y)$, $i \geq 2$

$(\exists i\ y)\varphi(y)$: there are are at least i nodes y such that $\varphi(y)$ holds.

Example: G is 7-regular is C^2-definable:

$$\forall x((\exists 7\ y)E(x,y) \wedge \neg(\exists 8\ y)E(x,y))$$

Theorem (Cai, Fürer, Immerman - 1992):
For every two graphs G and H, and for every $k \geq 2$, TFAE:

1. $G \equiv_{\mathrm{C}^k} H$ (i.e., G and H satisfy the same C^k-sentences).
2. G and H are indistinguishable by the $(k-1)$-dimensional Weisfeiler-Leman algorithm.

Restricted Left Profiles and Counting Logics

Theorem (Dvořák - 2010):

For every two graphs G and H, and for every $k \geq 2$, TFAE:

1. $G \equiv_{\mathrm{C}^k} H$ (i.e., G and H satisfy the same C^k-sentences).
2. $\hom(\mathscr{T}_k, G) = \hom(\mathscr{T}_k, H)$, where $\mathscr{T}_k$ is the class of all graphs of treewidth < k.

Restricted Left Profiles and Counting Logics

Theorem (Dvořák - 2010):
For every two graphs G and H, and for every $k \geq 2$, TFAE:

1. $G \equiv_{\mathrm{C}^k} H$ (i.e., G and H satisfy the same C^k-sentences).
2. $\hom(\mathscr{T}_k, G) = \hom(\mathscr{T}_k, H)$, where $\mathscr{T}_k$ is the class of all graphs of treewidth < k.

Note: The treewidth of a graph is a positive integer that measures how far from being a tree the graph is.

- Every tree has treewidth 1
- Every cycle has treewidth 2
- The clique K_n with n nodes has treewidth $n - 1$

In particular, $G \equiv_{\mathrm{C}^2} H$ iff $\hom(\textit{trees}, G) = \hom(\textit{trees}, H)$.

Restricted Left Profiles and Co-Spectrality

Definition:
Two graphs G, H are co-spectral if their adjacency matrices have the same spectrum, i.e., the same multiset of eigenvalues.

Example: $C_4 \oplus K_1$ and the star S_5 have spectrum $\{-2, 0^3, 2\}$.

Restricted Left Profiles and Co-Spectrality

Definition:
Two graphs G, H are co-spectral if their adjacency matrices have the same spectrum, i.e., the same multiset of eigenvalues.

Example: $C_4 \oplus K_1$ and the star S_5 have spectrum $\{-2, 0^3, 2\}$.

Theorem (Dell-Grohe-Rattan 2018):
For every two graphs G and H, the following are equivalent:

1. G and H are co-spectral.
2. $\hom(\mathscr{C}, G) = \hom(\mathscr{C}, H)$, where $\mathscr{C}$ is the class of all cycles.

Restricted Left Profiles vs. Restricted Right Profiles

- Restricted left profiles can capture interesting relaxations of isomorphism, such as C^k-equivalence and co-spectrality.
- Atserias-Kolaitis-Wu (2021): what about restricted right profiles?

Restricted Left Profiles vs Restricted Right Profiles

$\mathscr{G}$: all graphs $\mathscr{T}_k$: all graphs of treewidth $< k$ ($k \geq 2$)
$\mathscr{C}$: all cycles $\mathscr{K}$: all cliques

$\equiv$	$\hom(\mathscr{F}, \cdot)$	$\hom(\cdot, \mathscr{F})$
isomorphism	$\mathscr{G}$	$\mathscr{G}$
C^k-equivalence ($k \geq 2$)	$\mathscr{T}_k$	none
co-spectrality	$\mathscr{C}$	none
chromatic equivalence	none	$\mathscr{K}$
FO^k-equivalence ($k \geq 1$)	none	none
QD^k-equivalence ($k \geq 1$)	none	none

Note:

- FO^k: first-order sentences with at most k variables.
- QD^k: first-order sentences of quantifier depth at most k.

Modal Equivalence Relations

Jesse Comer (MSc Thesis 2023):

- Performed a similar analysis for modal equivalence relations between pointed labeled transition systems M_a
- Considers both $\hom_{\mathbb{B}}(\cdot,\cdot)$ and $\hom_{\mathbb{N}}(\cdot,\cdot)$ — following Atserias-Kolaitis-Wu (2021)

$$\hom_{\mathbb{B}}(F,G) = \begin{cases} 1, & \text{if } F \to G \\ 0, & \text{if } F \not\to G. \end{cases}$$

Language	Invariance Relation	Characterizing Vector
$\mathrm{ML}^{+}_{\Diamond}$	Simulation Equivalence	$\hom_{\mathbb{B}}(\mathcal{T}, M_a)$
ML^{+}	Directed Simulation Equivalence	None for $\mathbb{B}$ or $\mathbb{N}$
BML	Bisimulation	None for $\mathbb{B}$ or $\mathbb{N}$
$\mathrm{ML}_{\#}$	Graded Bisimulation	$\hom_{\mathbb{N}}(\mathcal{T}, M_a)$
$\mathrm{ML}^{+,B}_{\Diamond}$	Back-and-Forth Simulation Equivalence	$\hom_{\mathbb{B}}(\mathcal{A}, M_a)$
$\mathrm{ML}^{+,G}_{\Diamond}$	Global Simulation Equivalence	$\hom_{\mathbb{B}}(\mathcal{F}, M_a)$
$\mathrm{ML}^{B}_{\#}$	Back-and-Forth Graded Bisimulation	$\hom_{\mathbb{N}}(\mathcal{A}, M_a)$
$\mathrm{ML}^{G}_{\#}$	Global Graded Bisimulation	$\hom_{\mathbb{N}}(\mathcal{F}, M_a)$

See also: Barceló et al (2020). The logical expressiveness of graph neural networks.

Homomorphism Counts and Query Algorithms

Chen, Flum, Liu, and Xun - 2022

Introduced a framework for testing membership in a class of structures using finitely many homomorphism counts.

Homomorphism Counts and Query Algorithms

Chen, Flum, Liu, and Xun - 2022

Introduced a framework for testing membership in a class of structures using finitely many homomorphism counts.

Definition: A class $\mathcal{C}$ of structures admits a left query algorithm over $\mathbb{N}$, if for some $k \geq 1$, there are structures $F_1, F_2, \ldots, F_k$ and a set $X \subseteq \mathbb{N}^k$ such that for every structure G,

$$G \in \mathcal{C} \iff (\hom(F_1, G), \hom(F_2, G), \ldots, \hom(F_k, G)) \in X.$$

Homomorphism Counts and Query Algorithms

Chen, Flum, Liu, and Xun - 2022

Introduced a framework for testing membership in a class of structures using finitely many homomorphism counts.

Definition: A class $\mathcal{C}$ of structures admits a left query algorithm over $\mathbb{N}$, if for some $k \geq 1$, there are structures $F_1, F_2, \ldots, F_k$ and a set $X \subseteq \mathbb{N}^k$ such that for every structure G,

$$G \in \mathcal{C} \iff (\hom(F_1, G), \hom(F_2, G), \ldots, \hom(F_k, G)) \in X.$$

Fact: The following are equivalent:

1. $\mathcal{C}$ admits a left query algorithm over $\mathbb{N}$.
2. There is a finite class $\mathcal{F} = \{F_1, \ldots, F_k\}$ such that for all structures G and H, if $\hom(\mathcal{F}, G) = \hom(\mathcal{F}, H)$, then
$$G \in \mathcal{C} \iff H \in \mathcal{C}.$$

Homomorphism Counts and Query Algorithms

Definition: A class $\mathcal{C}$ of structures admits a left query algorithm over $\mathbb{N}$, if for some $k \geq 1$, there are structures $F_1, F_2, \ldots, F_k$ and a set $X \subseteq \mathbb{N}^k$ such that for every structure G,

$$G \in \mathcal{C} \iff (\hom(F_1, G), \hom(F_2, G), \ldots, \hom(F_k, G)) \in X.$$

Theorem: (Chen, Flum, Liu, and Xun - 2022)

- Every class of graphs definable by a Boolean combination of universal FO-sentences admits a left query algorithm over $\mathbb{N}$.
- The class of all K_3-free graphs does not admit a right query algorithm over $\mathbb{N}$.

Homomorphism Counts and Query Algorithms

tC-Dalmau-Kolaitis-Wu (to appear in ICDT 2024):

- studied query algorithms over the Boolean semiring $\mathbb{B}$;
- compared query algorithms over $\mathbb{B}$ to those over $\mathbb{N}$.

Homomorphism Counts and Query Algorithms

tC-Dalmau-Kolaitis-Wu (to appear in ICDT 2024):

- studied query algorithms over the Boolean semiring $\mathbb{B}$;
- compared query algorithms over $\mathbb{B}$ to those over $\mathbb{N}$.

$$\hom_{\mathbb{B}}(F, G) = \begin{cases} 1, & \text{if } F \to G \\ 0, & \text{if } F \not\to G. \end{cases}$$

Definition: A class $\mathcal{C}$ of structures admits a left query algorithm over $\mathbb{B}$, if for some $k \geq 1$, there are structures $F_1, F_2, \ldots, F_k$ and a set $X \subseteq \{0, 1\}^k$ such that for every structure G,

$$G \in \mathcal{C} \iff (\hom_{\mathbb{B}}(F_1, G), \hom_{\mathbb{B}}(F_2, G), \ldots, \hom_{\mathbb{B}}(F_k, G)) \in X.$$

Left Query Algorithms over $\mathbb{B}$

Theorem (tCDKW - 2023) Let $\mathcal{C}$ be a class of structures. TFAE:

1. $\mathcal{C}$ admits a left query algorithm over $\mathbb{B}$.
2. $\mathcal{C}$ is definable by a Boolean combination of conjunctive queries.
3. $\mathcal{C}$ is FO-definable and closed under homomorphic equivalence.

Proof Hint: (3) $\Longrightarrow$ (1) use tools by Rossman to prove the Homomorphism Preservation Theorem in the finite.

Left Query Algorithms over $\mathbb{B}$

Theorem (tCDKW - 2023) Let $\mathcal{C}$ be a class of structures. TFAE:

1. $\mathcal{C}$ admits a left query algorithm over $\mathbb{B}$.
2. $\mathcal{C}$ is definable by a Boolean combination of conjunctive queries.
3. $\mathcal{C}$ is FO-definable and closed under homomorphic equivalence.

Proof Hint: (3) $\Longrightarrow$ (1) use tools by Rossman to prove the Homomorphism Preservation Theorem in the finite.

Corollary: If $\mathcal{C}$ is closed under homomorphism equivalence, then TFAE:

1. $\mathcal{C}$ admits a left query algorithm over $\mathbb{B}$.
2. $\mathcal{C}$ is FO-definable.

Special Cases: $\mathrm{CSP}(H)$ and $[H]_{\leftrightarrow}$, for every structure H.

Existence vs. Counting ($\mathbb{B}$ vs. $\mathbb{N}$)

Fact: Let $\mathcal{C}$ be a class of structures.

- If $\mathcal{C}$ admits a left query algorithm over $\mathbb{B}$, then $\mathcal{C}$ admits a left query algorithm over $\mathbb{N}$.
- $\mathcal{C}$ may admit a left query algorithm over $\mathbb{N}$, but not over $\mathbb{B}$. For example, take $\mathcal{C}$ to be the class of all graphs with at least 7 edges.

Existence vs. Counting ($\mathbb{B}$ vs. $\mathbb{N}$)

Fact: Let $\mathcal{C}$ be a class of structures.

- If $\mathcal{C}$ admits a left query algorithm over $\mathbb{B}$, then $\mathcal{C}$ admits a left query algorithm over $\mathbb{N}$.
- $\mathcal{C}$ may admit a left query algorithm over $\mathbb{N}$, but not over $\mathbb{B}$. For example, take $\mathcal{C}$ to be the class of all graphs with at least 7 edges.

However, this is an unfair comparison:
If $\mathcal{C}$ admits a left query algorithm over $\mathbb{B}$, then $\mathcal{C}$ is closed under homomorphic equivalence.

Existence vs. Counting ($\mathbb{B}$ vs. $\mathbb{N}$)

Question:

- Is there a class $\mathcal{C}$ of structures that is closed under homomorphic equivalence, admits a left query algorithm over $\mathbb{N}$, but it does not admit a left query algorithm over $\mathbb{B}$?

In other words, is counting more powerful than existence as regards homomorphic-equivalence closed classes?

Existence vs. Counting ($\mathbb{B}$ vs. $\mathbb{N}$)

Theorem (tCDKW - 2023) Let $\mathcal{C}$ be a class of structures that is closed under homomorphic equivalence. TFAE:

1. $\mathcal{C}$ admits a left query algorithm of the form $(\mathcal{F}, X)$ over $\mathbb{N}$, for some set $X \subseteq \mathbb{N}^k$.
2. $\mathcal{C}$ admits a left query algorithm of the form $(\mathcal{F}, X')$ over $\mathbb{B}$, for some set $X' \subseteq \{0, 1\}^k$.

Existence vs. Counting ($\mathbb{B}$ vs. $\mathbb{N}$)

Theorem (tCDKW - 2023) Let $\mathcal{C}$ be a class of structures that is closed under homomorphic equivalence. TFAE:

1. $\mathcal{C}$ admits a left query algorithm of the form $(\mathcal{F}, X)$ over $\mathbb{N}$, for some set $X \subseteq \mathbb{N}^k$.
2. $\mathcal{C}$ admits a left query algorithm of the form $(\mathcal{F}, X')$ over $\mathbb{B}$, for some set $X' \subseteq \{0, 1\}^k$.

Proof Outline: (1) $\Longrightarrow$ (2)

- Write X as the disjoint union $X = \bigcup_{j=1}^{m} X_j$ of basic sets X_j, i.e.,
  if $\mathbf{t}, \mathbf{t}' \in X_j$, then $\mathbf{t}(i) = 0 \Longleftrightarrow \mathbf{t}'(i) = 0$, for all $i \leq k$.
- Show that if $\mathscr{C}$ is closed under homomorphic equivalence and admits a left query algorithm $(\mathcal{F}, X)$ over $\mathbb{N}$ where X is a basic set, then $\mathcal{C}$ is definable by

$$\psi : \quad (\bigwedge_{\mathbf{t}(i) \neq 0} Q^{F_i}) \wedge (\bigwedge_{\mathbf{t}(i)=0} \neg Q^{F_i}).$$

Existence vs. Counting ($\mathbb{B}$ vs. $\mathbb{N}$)

Goal: Show that if $\mathscr{C}$ is closed under homomorphic equivalence and admits a left query algorithm $(\mathcal{F}, X)$ over $\mathbb{N}$ where X is a basic set, then $\mathcal{C}$ is definable by

$$\psi : \quad (\bigwedge_{\mathbf{t}(i)\neq 0} Q^{F_i}) \wedge (\bigwedge_{\mathbf{t}(i)=0} \neg Q^{F_i}).$$

Existence vs. Counting ($\mathbb{B}$ vs. $\mathbb{N}$)

Goal: Show that if $\mathscr{C}$ is closed under homomorphic equivalence and admits a left query algorithm $(\mathcal{F}, X)$ over $\mathbb{N}$ where X is a basic set, then $\mathcal{C}$ is definable by

$$\psi: \quad (\bigwedge_{\mathbf{t}(i)\neq 0} Q^{F_i}) \wedge (\bigwedge_{\mathbf{t}(i)=0} \neg Q^{F_i}).$$

Given B such that $B \models \psi$, show $B \in \mathcal{C}$.

- Take $A \in \mathcal{C}$, construct A' and B' such that
  1. A' is a disjoint union of "many" copies of A and a disjoint union of direct products of members of $\mathcal{F}$ and substructures of members of $\mathcal{F}$; similarly for B' and B.
  2. $A' \leftrightarrow A$ and $B' \leftrightarrow B$.
  3. $\hom(\mathcal{F}, A') = \hom(\mathcal{F}, B')$ (this uses an interpolation lemma for multivariate integer polynomials).
- By (2), $A' \in \mathcal{C}$; by (3), $B' \in \mathcal{C}$; by (2), $B \in \mathcal{C}$. $\square$

Synopsis

- Homomorphism counts capture interesting relaxations of isomorphism.
- Sharp differences in expressive power exist between restricted left profiles and restricted right profiles.
- Homomorphism counts give rise to algorithms for testing for membership in a class of structures.
- For left query algorithms and homomorphic-equivalence closed classes, counting homomorphisms is not more powerful than existence of homomorphisms.

Open Problems

- For right query algorithms and homomorphic-equivalence closed classes, is counting homomorphisms more powerful than existence of homomorphisms?

Open Problems

- For right query algorithms and homomorphic-equivalence closed classes, is counting homomorphisms more powerful than existence of homomorphisms?
- Characterize the logics L for which L-equivalence $\equiv_L$ is captured by a restricted left or by a restricted right profile.

  Tarski's Program: Characterize notions of "metamathematical origin" in "purely mathematical terms".