Counting Homomorphisms from Hypergraphs of Bounded Generalised Hypertree Width

A Logical Characterisation

Benjamin Scheidt   Nicole Schweikardt

MFCS'23

Humboldt-Universität zu Berlin

What is this about?

Contact me at benjamin.scheidt@hu-berlin.de
Slides available at hu.berlin/mfcs23
arXiv:2303.10980

Motivation and Main Result

GC^k — A 2-sorted Counting Logic with Guards

Notes on the Proof

Conclusion

Counting homomorphisms

$$\text{hom}(F, G) = \#\text{homs } F \to G$$
$$\text{Hom}_{\mathfrak{C}}(G) = \big(\text{hom}(F, G)\big)_{F \in \mathfrak{C}}$$

Can be used to distinguish graphs. Q: How well does this work?

A: Very well. Many results that look like this:

Theorem

Let G, H be graphs and $\mathfrak{C}$ a class of graphs.

$$\text{Hom}_{\mathfrak{C}}(G) = \text{Hom}_{\mathfrak{C}}(H) \quad \iff \quad G \equiv_X H$$

- X may be a logic, an algorithm, ...
- Natural classes $\mathfrak{C}$ lead to natural X!

An incomplete history of homomorphism indistinguishability

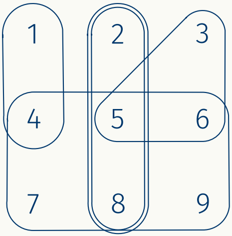
G and H are homomorphism indistinguishable over

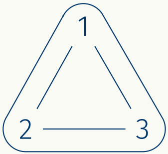
- all graphs iff isomorphic (Lovász 1967)
- Graphs of tree-width $\leq k$ iff indistinguishable by C^{k+1} (Dvořák 2010; Dell, Grohe, and Rattan 2018)
- Graphs of tree-depth $\leq m$ iff indistinguishable by C_m (Grohe 2020a)
- Berge-acyclic hypergraphs iff indistinguishable by Colour Refinement (Böker 2019)

What about hypergraphs?

What are Hypergraphs?

A hypergraph H is a finite set $V(H)$ of vertices and a finite **multi**set $E(H)$ of edges $e \subseteq V(H)$.





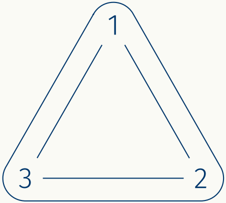
Incidence Graphs

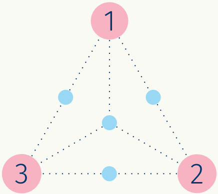
Definition

The incidence graph $I_H = (R, B, E)$ of H is defined as:

$$R = V(H) \qquad B = E(H) \qquad (e, v) \in E \iff v \in f_H(e)$$

i.e. draw an edge if v is contained in the hyperedge e.





Main Result – Counting Hyper-what?!

Theorem

Let G, H be hypergraphs, $k \in \mathbb{N}_{\geqslant 1}$.

$$\mathsf{Hom}_{\mathsf{GHW}_k}(G) = \mathsf{Hom}_{\mathsf{GHW}_k}(H) \quad \Longleftrightarrow \quad G \equiv_{\mathsf{GC}^k} H$$

Theorem (Dvořák 2010; Dell, Grohe, and Rattan 2018)

Let G, H be graphs, $k \in \mathbb{N}_{\geqslant 1}$.

$$\mathsf{Hom}_{\mathsf{TW}_k}(G) = \mathsf{Hom}_{\mathsf{TW}_k}(H) \quad \Longleftrightarrow \quad G \equiv_{\mathsf{C}^{k+1}} H$$

GC^k is a new logic introduced by us.

GHW_k are the hypergraphs of generalised hypertree width $\leq k$.

Introducing GC^k — The Basics

- Blue and red Variables

$$\begin{aligned} \mathsf{VAR} := \{\mathsf{e}_1, \dots, \mathsf{e}_k\} & \quad \text{represent edges} \\ \cup \{\mathsf{v}_1, \mathsf{v}_2, \mathsf{v}_3, \dots\} & \quad \text{represent vertices} \end{aligned}$$

- Guard function $g : \mathbb{N}_{\geqslant 1} \rightharpoonup [k]$

$$\Delta_g := \bigwedge_{i \in \mathsf{dom}(g)} E(\mathsf{e}_{g(i)}, \mathsf{v}_i) \qquad \text{“}v_i \in e_{g(i)}\text{”}$$

$$\Delta_g := \top \text{ for } \mathsf{dom}(g) = \emptyset$$

- red variables are always guarded by some blue variable

Introducing GC^k — Definition

We have...

Atomic formulas:

- $E(\mathsf{e}_i, \mathsf{v}_j), \quad \mathsf{v}_i = \mathsf{v}_j, \quad \mathsf{e}_i = \mathsf{e}_j$

Negation, Conjunction and Disjunction:

- $\neg\varphi, \quad (\varphi \wedge \psi), \quad (\varphi \vee \psi)$

Guarded Quantification:

- $\exists^{\geq n}(\mathsf{v}_{i_1}, \ldots, \mathsf{v}_{i_\ell}).(\Delta_g \wedge \varphi)$
- $\exists^{\geq n}(\mathsf{e}_{i_1}, \ldots, \mathsf{e}_{i_\ell}).(\Delta_g \wedge \varphi)$

where Δ_g has to guard *all* free red variables in φ.

GC^k – Example

Find φ that expresses:

All hyperedges containing at least 3 vertices are pair-wise disjoint.

$$\varphi := \neg\exists^{\geq 1}(\mathsf{e}_1, \mathsf{e}_2) \,.\, (\underbrace{\top}_{\text{Guard}} \wedge (\neg \mathsf{e}_1 = \mathsf{e}_2 \wedge \psi_{\text{not disjoint}} \wedge \psi_{\geqslant 3 \text{ vertices}}))$$

with

$$\psi_{\text{not disjoint}} := \exists^{\geq 1}(\mathsf{v}_1) \,.\, (\underbrace{E(\mathsf{e}_1, \mathsf{v}_1)}_{\text{Guard}} \wedge \bigwedge_{j \in \{1,2\}} E(\mathsf{e}_j, \mathsf{v}_1))$$

$$\psi_{\geqslant 3 \text{ vertices}} := \bigwedge_{j \in \{1,2\}} \exists^{\geq 3}(\mathsf{v}_1) \,.\, (\underbrace{E(\mathsf{e}_j, \mathsf{v}_1)}_{\text{Guard}} \wedge E(\mathsf{e}_j, \mathsf{v}_1))$$

Generalised Hypertree Width

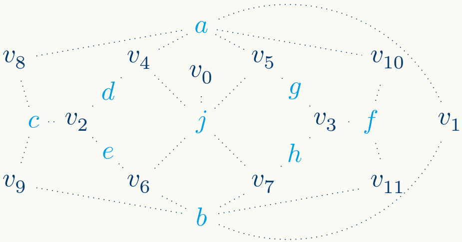
- Generalisation of treewidth to hypergraphs.
- Based on (Hyper)tree decompositions – as usual:
  - Bags of vertices associated with each node of the tree
  - F.a. vertices v: the bags containing v must form a connected subtree.

The twist:

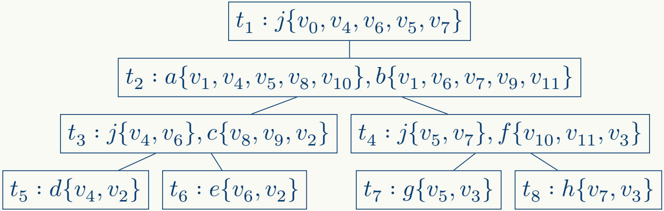
- Every bag has to be *covered* by a set of hyperedges
  - I.e. the union of the hyperedges must be a superset of the bag.
- The width of a decomposition is the size of the biggest cover.

Example

F:



a
v8
v4
v0
v5
v10
d
g
c
v2
j
v3
f
v1
e
h
v9
v6
v7
v11
b



$t_1 : j\{v_0, v_4, v_6, v_5, v_7\}$
$t_2 : a\{v_1, v_4, v_5, v_8, v_{10}\}, b\{v_1, v_6, v_7, v_9, v_{11}\}$
$t_3 : j\{v_4, v_6\}, c\{v_8, v_9, v_2\}$
$t_4 : j\{v_5, v_7\}, f\{v_{10}, v_{11}, v_3\}$
$t_5 : d\{v_4, v_2\}$
$t_6 : e\{v_6, v_2\}$
$t_7 : g\{v_5, v_3\}$
$t_8 : h\{v_7, v_3\}$

Notes on the proof 🧐

Caution when dealing with homomorphisms

Definition (Homomorphisms on Hypergraphs)

Let $h_V : V(H) \to V(G)$ and $h_E : E(H) \to E(G)$. (h_V, h_E) is a homomorphism from G to H if for all $e \in E(G)$:

$$f_H(h_E(e)) = \{ h_V(v) \mid v \in f_G(e) \}$$

$$\not\equiv$$

Böker 2019 implicitly shows that this is not problematic

Definition (Homomorphisms on Incidence Graphs)

Let $h_R : R(I_G) \to R(I_H)$ and $h_B : B(I_G) \to B(I_H)$. (h_R, h_B) is a homomorphism from I_G to I_H if:

$$(e, v) \in E(I_G) \implies (h_B(e), h_R(v)) \in E(I_H)$$

About that Proof...

Step 1: Relate homomorphisms over hypergraphs and incidence graphs

- **Recall**: Not the same — but Böker 2019 solved this already

Step 2: Relate ghw to restricted variant ehw

- Gives us an inductive characterisation

Step 3: Find suitable “normal form” RGC^k for GC^k

Step 4: Relate ehw and RGC^k

- Adapt beautiful machinery in spirit of Courcelle 1993 to incidence graphs
- **Nasty** — Like getting two cats to the vet using a single pet carrier 🐈📦🐈‍⬛
  - Stuff one cat in the box and the other escapes — Repeat until they accept their fate

Step 5: Profit 🎉

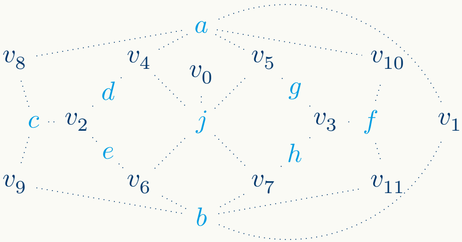
Generalised Hypertree Width

A generalised hypertree decomposition is a tree with

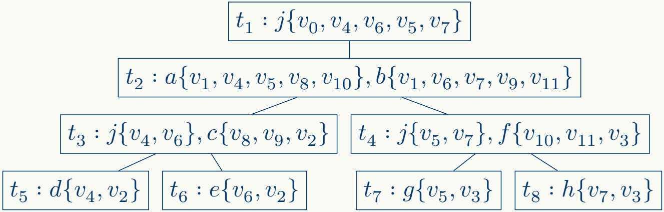
- Bags (set of vertices) and covers (set of hyperedges) associated with each node of the tree
- F.a. vertices v: the bags containing v must form a connected subtree
- F.a. hyperedges e: the bags covered by e must form a connected subtree
- Every bag has to be precisely *covered* by a set of hyperedges
  - I.e. the union of the hyperedges must be equal to the bag
- The width of a decomposition is the size of the biggest cover.

Example

F:



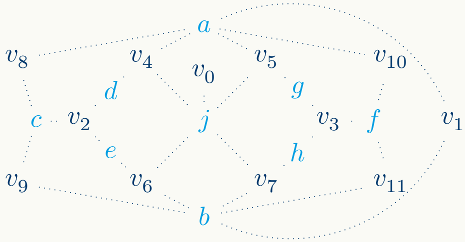
a
v_8
v_4
v_0
v_5
v_{10}
d
g
c
v_2
j
v_3
f
v_1
e
h
v_9
v_6
v_7
v_{11}
b



$t_1 : j\{v_0, v_4, v_6, v_5, v_7\}$
$t_2 : a\{v_1, v_4, v_5, v_8, v_{10}\}, b\{v_1, v_6, v_7, v_9, v_{11}\}$
$t_3 : j\{v_4, v_6\}, c\{v_8, v_9, v_2\}$
$t_4 : j\{v_5, v_7\}, f\{v_{10}, v_{11}, v_3\}$
$t_5 : d\{v_4, v_2\}$
$t_6 : e\{v_6, v_2\}$
$t_7 : g\{v_5, v_3\}$
$t_8 : h\{v_7, v_3\}$

Relating ghw and ehw

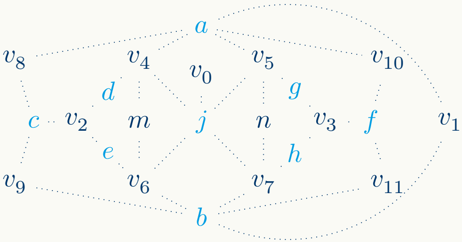
F:



- Assume we want to use F with $\mathrm{ghw}(F) = 2$ to distinguish some G and H by number of homomorphisms.
- But $\mathrm{ehw}(F) = 3$.
- Fix: Turn F into F' with $\mathrm{ehw}(F') = 2$ while keeping distinguishing property.

Simple Idea: Just add more edges 🤷‍♂️

F':



$t_1 : j\{v_0, v_4, v_6, v_5, v_7\}\}$

$t_2 : a\{v_1, v_4, v_5, v_8, v_{10}\}, b\{v_1, v_6, v_7, v_9, v_{11}\}$

$t_3 : m\{v_4, v_6\}, c\{v_8, v_9, v_2\}$ $\quad$ $t_4 : n\{v_5, v_7\}, f\{v_{10}, v_{11}, v_3\}$

$t_5 : d\{v_4, v_2\}$ $\quad$ $t_6 : e\{v_6, v_2\}$ $\quad$ $t_7 : g\{v_5, v_3\}$ $\quad$ $t_8 : h\{v_7, v_3\}$

But this changes the number of homomorphisms!

Relating ghw and ehw

Lemma (simplified)

Let G, H, F be hypergraphs s.t. $\hom(F, G) \neq \hom(F, H)$ *and let $e \in E(F)$.*

Then, for every $s \subseteq e$ and every $x \in \mathbb{N}$ there exists a $y \geq x$ such that $\hom(F + y \cdot s, G) \neq \hom(F + y \cdot s, H)$.

Simpler:

> *"If we want to add x copies of a subset of some hyper-edge to F, we can retain its distinguishing property as long as we can handle some additional copies."*

I.e. insert copies of (sub)edges to enforce precise coverage and connectedness for edges.

Proving this lemma

Core Idea

If adding x copies aligns the number of homomorphisms, just keep adding copies. At some point, it *must* break again.

Makes sense. But can we prove that? On a very high-level?

- → $s \subseteq e$ gives handle on how the number of homs change. Provides formula for

$$\text{hom}(F + x \cdot s, G) - \text{hom}(F + x \cdot s, H)$$

- → Contradict assumption that number of homomorphisms is equal for every x, i.e. the difference is always 0.
- → Translate to system of linear equations:

$$V \times B = 0 \quad \text{with } B \neq 0 \text{ follows from assumption}$$

Proving this lemma

- V is a Vandermonde matrix whose determinant is non-zero — thus V is invertible!
- I.e. $V \times B = 0 \iff B = V^{-1} \times 0 \iff B = 0$.
- This contradicts $B \neq 0$. $\square$

Take Home Message

Core Idea

If adding x copies aligns the number of homomorphisms, just keep adding copies. At some point, it *must* break again.

Heart of the proof

A Vandermonde matrix is invertible.

Concluding Remarks

Theorem

The following statements are equivalent:

1. *Ex.* $\varphi \in \mathsf{GC}^k$ *s.t.* $G \models \varphi$ *and* $H \not\models \varphi$
2. *Ex.* F *with* $\mathrm{ghw}(F) \leq k$ *s.t.* $\hom(F, G) \neq \hom(F, H)$

So…

- …how hard is it to compute the homomorphism vector?
- …EF-games for GC^k?
- …does Grohe 2020a generalise to hypertree-depth?
- …Weisfeiler-Leman on hypergraphs? Ongoing work!

Related Work i

- 📝 Böker, Jan (2019). "Color Refinement, Homomorphisms, and Hypergraphs".
  DOI: 10.1007/978-3-030-30786-8_26.
- 📝 Cai, Jin-Yi, Martin Fürer, and Neil Immerman (1992). "An optimal lower bound on the number of variables for graph identification".
  DOI: 10.1007/BF01305232.
- 📝 Courcelle, Bruno (1993). "Graph Grammars, Monadic Second-Order Logic And The Theory Of Graph Minors".
  DOI: 10.1090/conm/147.

Related Work ii

📝 Dell, Holger, Martin Grohe, and Gaurav Rattan (2018). "Lovász Meets Weisfeiler and Leman".
DOI: 10.4230/LIPIcs.ICALP.2018.40.

📝 Dvořák, Zdeněk (2010). "On recognizing graphs by numbers of homomorphisms".
DOI: 10.1002/jgt.20461.

📝 Gottlob, Georg, Nicola Leone, and Francesco Scarcello (2002). "Hypertree Decompositions and Tractable Queries".
DOI: 10.1006/jcss.2001.1809.

📝 Grohe, Martin (2020a). "Counting Bounded Tree Depth Homomorphisms".
DOI: 10.1145/3373718.3394739.

Related Work iii

📝 Grohe, Martin (2020b). "Word2vec, Node2vec, Graph2vec, X2vec: Towards a Theory of Vector Embeddings of Structured Data".
DOI: 10.1145/3375395.3387641.

📝 Grohe, Martin (2021). "The Logic of Graph Neural Networks".
DOI: 10.1109/LICS52264.2021.9470677.

📝 Lovász, László (1967). "Operations with structures".
DOI: 10.1007/BF02280291.

📝 Mančinska, Laura and David E. Roberson (2020). "Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs".
DOI: 10.1109/FOCS46700.2020.00067.

Definition (Gottlob, Leone, and Scarcello 2002)

A complete generalised hypertree decomposition $D := (T, cover, bag)$ of H is a tree T with functions $cover : V(T) \to E(H)$ and $bag : V(T) \to V(H)$ s.t.

1. F.a. $e \in E(H)$ ex. $n \in V(T)$ s.t. $e \in cover(n)$ and $f(e) \subseteq bag(n)$
2. F.a. $v \in V(H)$ the subgraph T_v induced by $V_v := \{n \in V(T) \mid v \in bag(n)\}$ is a tree.
3. F.a. $n \in V(T)$ we have $bag(n) \subseteq \bigcup_{e \in cover(n)} f(e)$.

Definition

A complete entangled hypertree decomposition $D := (T, cover, bag)$ of H is a tree T with functions $cover : V(T) \to E(H)$ and $bag : V(T) \to V(H)$ s.t.

1. F.a. $e \in E(H)$ ex. $n \in V(T)$ s.t. $e \in cover(n)$ and $f(e) \subseteq bag(n)$
2. F.a. $v \in V(H)$ the subgraph T_v induced by $V_v := \{n \in V(T) \mid v \in bag(n)\}$ is a tree.
3. F.a. $n \in V(T)$ we have $bag(n) = \bigcup_{e \in cover(n)} f(e)$.
4. F.a. $e \in E(H)$ the subgraph T_e induced by $V_e := \{n \in V(T) \mid e \in cover(n)\}$ is a tree.