Counting Homomorphisms from Hypergraphs of Bounded Generalised Hypertree Width

A Logical Characterisation

Benjamin Scheidt Nicole Schweikardt

AlMoTh 2023

Humboldt-Universität zu Berlin

What is this about?

Contact me at `benjamin.scheidt@hu-berlin.de`
Slides available at `hu.berlin/AlMoTh23`
arXiv is in the works

Counting homomorphisms is a hot topic!

Many (classical and recent) results on graphs of the following form:

Theorem

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

$$\underbrace{\mathrm{Hom}_{\mathcal{C}}(G) = \mathrm{Hom}_{\mathcal{C}}(H)}_{\text{homomorphism indistinguishable over } \mathcal{C}} \iff G \equiv_X H$$

X may be

- a logic
- an algorithm
- …

An incomplete history of homomorphism indistinguishability

G and H homomorphism indistinguishable over

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

What about hypergraphs?

Main Result – Counting what...?

Theorem

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

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

Theorem (Dvořák 2010)

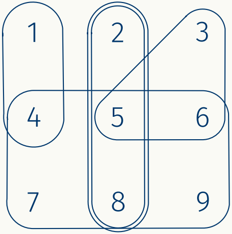
Let G, H be graphs, $k \in \mathbb{N}_{\geqslant 1}$.

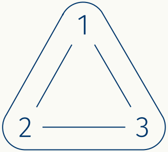
$$\mathsf{Hom}_{\mathsf{TW}_k}(G) = \mathsf{Hom}_{\mathsf{TW}_k}(H) \quad \iff \quad G \equiv_{\mathsf{C}^{k+1}} H$$

GC^k is a new logic introduced by us.

What are Hypergraphs?

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





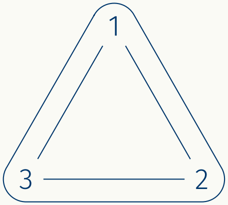
Incidence Graphs

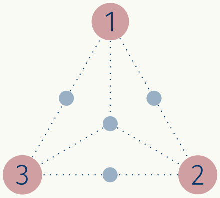
Definition

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.





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:

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

Introducing GC^k — The Basics

- Blue and red Variables

$$\begin{aligned} \mathrm{VAR} &:= \{e_1, \dots, e_k\} && \text{represent edges} \\ &\cup \{v_1, v_2, v_3, \dots\} && \text{represent vertices} \end{aligned}$$

- Attached guard function $g : \mathbb{N}_{\geqslant 1} \to [k]$

$$\Delta_g := \bigwedge_{i \in \mathrm{dom}(g)} E(e_{g(i)}, v_i) \qquad \text{“}v_i \in f(e_{g(i)})\text{”}$$

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

- red variables are always guarded by some blue variable

Find φ that expresses:

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

$$\varphi \coloneqq \neg\exists^{\geq 1}(e_1, e_2) \,.\, \Big(\underbrace{\top}_{\text{Guard}} \wedge (\neg e_1 = e_2 \wedge \psi_{\text{not disjoint}} \wedge \psi_{\geqslant 3\text{ vertices}})\Big)$$

with

$$\psi_{\text{not disjoint}} \coloneqq \exists^{\geq 1}(v_1) \,.\, \Big(\underbrace{E(e_1, v_1)}_{\text{Guard}} \wedge \bigwedge_{j \in \{1,2\}} E(e_j, v_1)\Big)$$

$$\psi_{\geqslant 3\text{ vertices}} \coloneqq \bigwedge_{j \in \{1,2\}} \exists^{\geq 3}(v_1) \,.\, (\underbrace{E(e_j, v_1)}_{\text{Guard}} \wedge E(e_j, v_1))$$

About that Proof...

Step 1: Relate homomorphisms over hypergraphs and incidence graphs

- **Recall**: They are not the same!
- Böker 2019 already did all the work

Step 2: Relate ghw and a restricted variant

- Manageable with the right idea
- Gives us an inductive characterisation

Step 3: Relate restricted variant and logic GC^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

Concluding Remarks

Theorem

The following statements are equivalent:

1. *Ex.* $\varphi \in \mathsf{GC}^k$ *s.t.* $I_G \vDash \varphi$ *and* $I_H \nvDash \varphi$
2. *Ex.* F *with* $\mathrm{ghw}(F) \leq k$ *s.t.* $\mathrm{hom}(F, G) \neq \mathrm{hom}(F, H)$

So...

- ...how hard is it to compute the homomorphism vector?
- ...EF-games for GC^k?
- ...does Grohe 2020a generalise to hypertree-depth?
- ...k-WL 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.

📝 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`.

On Hypertree Decompositions

Definition (Gottlob et al. 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 $bag(n) \subseteq \bigcup_{e \in cover(n)} f(e)$.
4. F.a. $n \in V(T)$ we have $bag(n) = \bigcup_{e \in cover(n)} f(e)$.
5. 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.

GC^k — Atomic formulas

$(\Delta_g \wedge \varphi) \in \mathrm{GC}^k$ for atomic formulas:

- $\varphi \coloneqq E(e_i, v_j)$ and $j \in \mathrm{dom}(g)$
- $\varphi \coloneqq e_i{=}e_j$ and $g \coloneqq \varnothing$
- $\varphi \coloneqq v_i{=}v_j$ and $i, j \in \mathrm{dom}(g)$

GC^k — Simple Recursive Cases

1. If $(\Delta_g \wedge \varphi) \in \mathsf{GC}^k$
   - then $(\Delta_g \wedge \neg\varphi) \in \mathsf{GC}^k$
2. If $(\Delta_f \wedge \alpha) \in \mathsf{GC}^k$ and $(\Delta_h \wedge \beta) \in \mathsf{GC}^k$
   - And $f(i) = h(i)$ f.a. $i \in \mathrm{dom}(f) \cap \mathrm{dom}(h)$
     - then $(\Delta_g \wedge (\alpha \wedge \beta)) \in \mathsf{GC}^k$
     - for $g = f \cup h$

GC^k — Advanced Recursive Cases

3. If $(\Delta_f \land \varphi) \in \mathrm{GC}^k$ and $i_1, \dots, i_\ell \in \mathrm{dom}(f)$
   - then $\big(\Delta_g \land \exists^{\geq n}(v_{i_1}, \dots, v_{i_\ell}) \,.\, (\Delta_f \land \varphi)\big) \in \mathrm{GC}^k$
   - for $g := f - \{i_1, \dots, i_\ell\}$

4. If $(\Delta_f \land \varphi \in \mathrm{GC}^k)$ and $e_{i_1}, \dots, e_{i_\ell}$ are free
   - then $\big(\Delta_g \land \exists^{\geq n}(e_{i_1}, \dots, e_{i_\ell}).(\Delta_f \land \varphi)\big) \in \mathrm{GC}^k$
   - For every g that *safely transitions guards*
     - $\mathrm{dom}(g) = \mathrm{dom}(f)$ and
     - $g(i) = f(i)$ or
     - $g(i) \in \{i_1, \dots, i_\ell\}$ or
     - $g(i) \notin \mathrm{img}(f)$