Counting and Finding Homomorphisms is Universal for Parameterized Complexity Theory

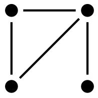
Marc Roth (Merton College, Oxford University, United Kingdom),
Philip Wellnitz (MPII, SIC, Saarbrücken, Germany)

SODA 2020

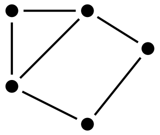
Finding Graph Homomorphisms

Graph Homomorphism

Mapping from graph H to G that preserves edges;
Write $\mathrm{Hom}(H \to G)$ for the set of all graph hom's from H to G.



H

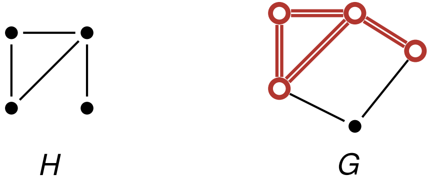


G

Finding Graph Homomorphisms

Graph Homomorphism

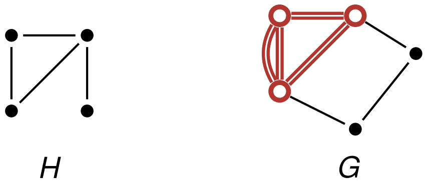
Mapping from graph H to G that preserves edges;
Write $\mathrm{Hom}(H \to G)$ for the set of all graph hom's from H to G.



Finding Graph Homomorphisms

Graph Homomorphism

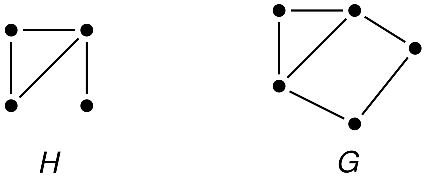
Mapping from graph H to G that preserves edges;
Write $\mathrm{Hom}(H \to G)$ for the set of all graph hom's from H to G.



Finding Graph Homomorphisms

Graph Homomorphism

Mapping from graph H to G that preserves edges;
Write $\mathrm{Hom}(H \to G)$ for the set of all graph hom's from H to G.

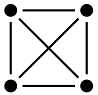


$$\#\mathrm{Hom}(H \to G) = 14$$

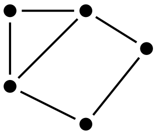
Finding Graph Homomorphisms

Graph Homomorphism

Mapping from graph H to G that preserves edges;
Write $\mathrm{Hom}(H \to G)$ for the set of all graph hom's from H to G.



H

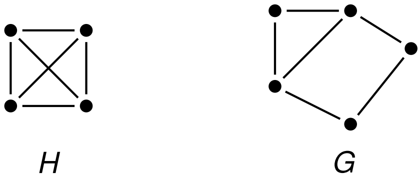


G

Finding Graph Homomorphisms

Graph Homomorphism

Mapping from graph H to G that preserves edges;
Write $\mathrm{Hom}(H \to G)$ for the set of all graph hom's from H to G.



No homomorphisms from H to G.

Finding Graph Homomorphisms

HOM($\mathcal{H} \to \mathcal{G}$)

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

Finding Graph Homomorphisms

HOM($\mathcal{H} \to \mathcal{G}$)

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

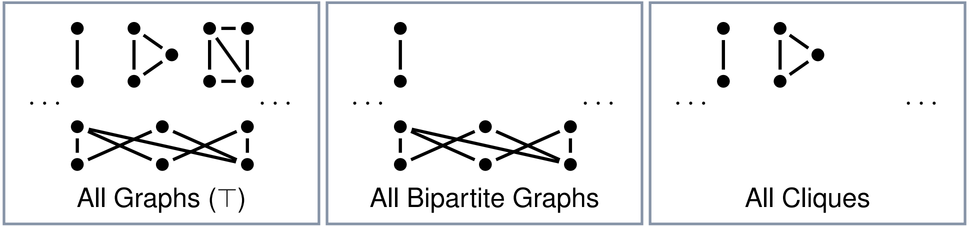
Graph classes

Finding Graph Homomorphisms

HOM($\mathcal{H} \to \mathcal{G}$)

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

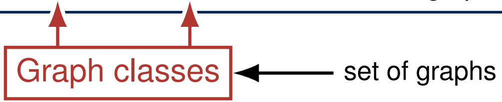
Graph classes

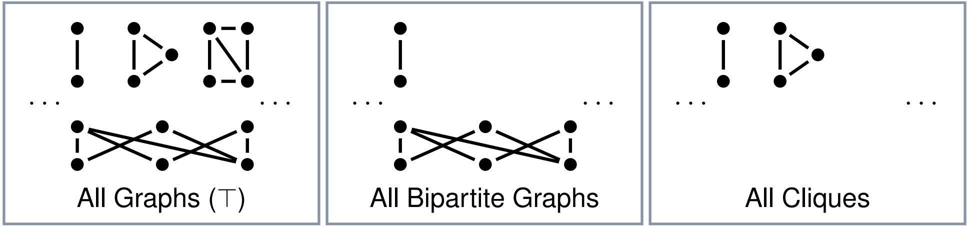


Finding Graph Homomorphisms

HOM($\mathcal{H} \to \mathcal{G}$)

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.





Known Results

HOM($\mathcal{H} \to \mathcal{G}$)

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

Known Results

$\textsc{Hom}(\mathcal{H} \to \mathcal{G})$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

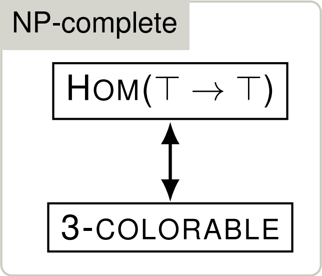
NP-complete

$\textsc{Hom}(\top \to \top)$

Known Results

Hom($\mathcal{H} \to \mathcal{G}$)

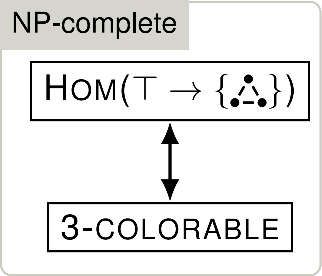
Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.



Known Results

HOM($\mathcal{H} \to \mathcal{G}$)

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.



Known Results

$\textsc{Hom}(\mathcal{H} \to \mathcal{G})$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

Are there fast algorithms for special cases of $\textsc{Hom}(\top \to \top)$?

Known Results

$\textsc{Hom}(\mathcal{H} \to \mathcal{G})$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

What makes $\textsc{Hom}(\top \to \top)$ hard?

Known Results

$\textsc{Hom}(\mathcal{H} \to \mathcal{G})$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

	poly-time solvable	NP-complete
$\textsc{Hom}(\top \to \mathcal{G})$	$\mathcal{G}$ contains only bipartite graphs [Hell, Nešetřil ’90]	$\mathcal{G}$ contains a non-bipartite graph [Hell, Nešetřil ’90]

Known Results

#Hom($\mathcal{H} \to \mathcal{G}$)

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

	poly-time solvable	#P-complete
#Hom($\top \to \mathcal{G}$)	(explicit criterion exists) [Dyer, Greenhill '00]	(explicit criterion exists) [Dyer, Greenhill '00]

Known Results

$\textsc{Hom}(\mathcal{H} \to \mathcal{G})$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

What about *the other side*, $\textsc{Hom}(\mathcal{H} \to \top)$?

Known Results

HOM($\mathcal{H} \to \mathcal{G}$) **Parameter: $|V(H)|$**

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

What about *the other side*, HOM($\mathcal{H} \to \top$)?

Known Results

Hom($\mathcal{H} \to \mathcal{G}$) — Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, check if there is a graph hom from H to G.

	FPT ($f(\|V(H)\|) \cdot \text{poly}(\|V(G)\|)$ time)	W[1]-hard (not faster than k-Clique)
Hom($\mathcal{H} \to \top$)	"$\mathcal{H}$ contains only graphs with small treewidth" [Grohe '03]	"$\mathcal{H}$ contains graphs with arbitrary large tw" [Grohe '03]

Known Results

#HOM($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

	FPT ($f(\|V(H)\|) \cdot \text{poly}(\|V(G)\|)$ time)	#W[1]-hard (not faster than #K-CLIQUE)
#HOM($\mathcal{H} \to \top$)	"$\mathcal{H}$ contains only graphs with small treewidth" [Dalmau, Jonsson '04]	"$\mathcal{H}$ contains a graph with large treewidth" [Dalmau, Jonsson '04]

Known Results

#Hom($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Complexity dichotomies when restricting either $\mathcal{G}$ or $\mathcal{H}$.

Known Results

#HOM($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Complexity dichotomies when restricting either $\mathcal{G}$ or $\mathcal{H}$.

What if we restrict *both sides*?

Known Results

#Hom($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Complexity dichotomies when restricting either $\mathcal{G}$ or $\mathcal{H}$.

What if we restrict *both sides*?

This talk.

Main Result

$\#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$ Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Theorem

For any problem P in #W[1] (or W[1]), there are graph classes $\mathcal{H}_P$ and $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ (or $\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$).

Main Result

#Hom($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Theorem

For any problem P in #W[1] (or W[1]), there are graph classes $\mathcal{H}_P$ and $\mathcal{G}_P$ such that P is equivalent to #Hom($\mathcal{H}_P \to \mathcal{G}_P$) (or Hom($\mathcal{H}_P \to \mathcal{G}_P$)).

- Cannot hope for clear categorization into FPT/W[1]-hard for all pairs $(\mathcal{H}, \mathcal{G})$ (think of Ladner's Theorem)

Proof Ideas

#HOM($\mathcal{H} \to \mathcal{G}$) — Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Theorem

For any problem P in #W[1] (or W[1]), there are graph classes $\mathcal{H}_P$ and $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$) (or HOM($\mathcal{H}_P \to \mathcal{G}_P$)).

Proof Ideas

#Hom($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Theorem

For any problem P in #W[1] (or W[1]), there are graph classes $\mathcal{H}_P$ and $\mathcal{G}_P$ such that P is equivalent to #Hom($\mathcal{H}_P \to \mathcal{G}_P$) (or Hom($\mathcal{H}_P \to \mathcal{G}_P$)).

Recall: #Hom($\mathcal{H} \to \top$) is #W[1]-hard if $\mathcal{H}$ has "unbounded treewidth" [DalJon'04]

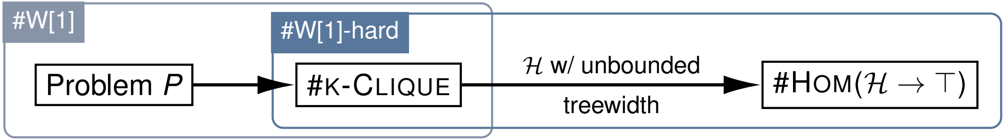
Proof Ideas

#HOM($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count the number of graph hom's from H to G.

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$).



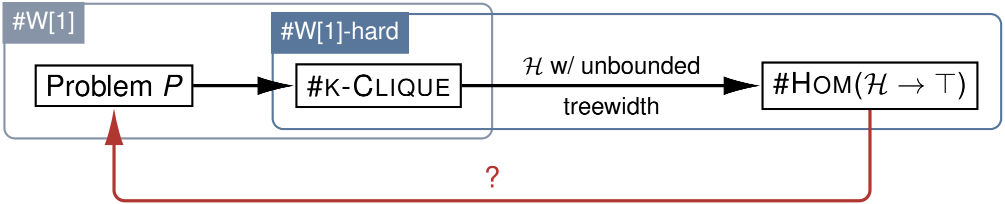
Proof Ideas

#HOM($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count the number of graph hom's from H to G.

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$).



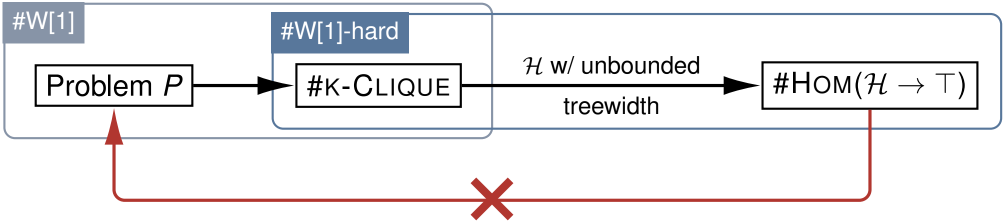
Proof Ideas

#HOM($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count the number of graph hom's from H to G.

Theorem

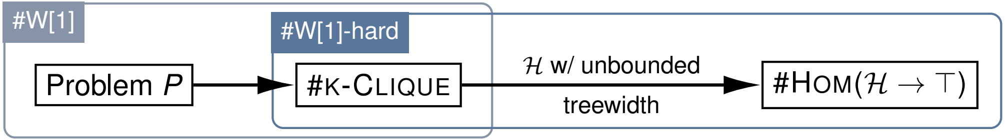
For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$).



Proof Ideas

Theorem

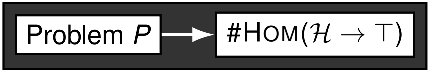
For any P in #W[1], there are $\mathcal{H}_P, \mathcal{G}_P$ such that P is equivalent to #HOM$(\mathcal{H}_P \to \mathcal{G}_P)$.



Proof Ideas

Theorem

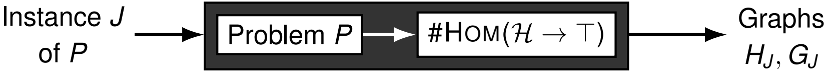
For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Proof Ideas

Theorem

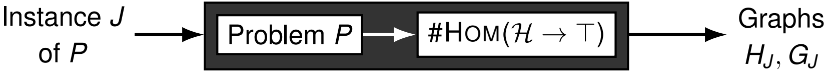
For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



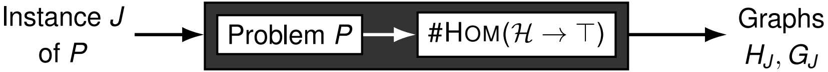
Approach:

$$\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$$
$$\mathcal{G}_P := \{G_J \mid \text{instance } J \text{ of } P\}$$

Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P, \mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



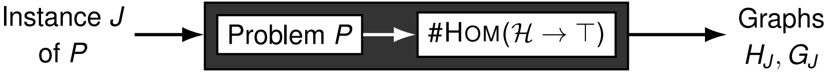
Approach:
$$\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$$
$$\mathcal{G}_P := \{G_J \mid \text{instance } J \text{ of } P\}$$

$\mathrm{P} \preccurlyeq \#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ ✓

Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P, \mathcal{G}_P$ such that P is equivalent to $\#\text{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Approach: $\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$
$\mathcal{G}_P := \{G_J \mid \text{instance } J \text{ of } P\}$

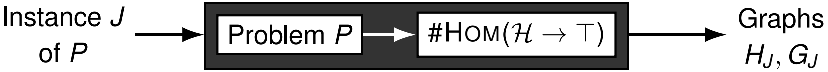
$\mathrm{P} \preccurlyeq \#\text{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ ✓

$\#\text{Hom}(\mathcal{H}_P \to \mathcal{G}_P) \overset{?}{\preccurlyeq} \mathrm{P}$

Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Approach: $\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$
$\mathcal{G}_P := \{G_J \mid \text{instance } J \text{ of } P\}$

$\mathrm{P} \preccurlyeq \#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ ✓

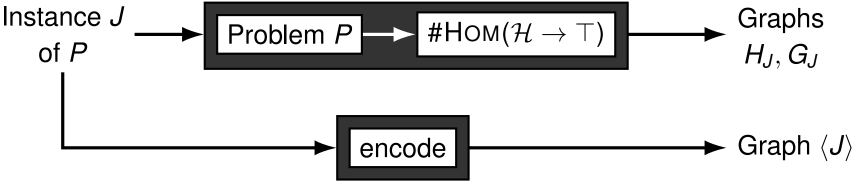
$\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P) \not\preccurlyeq \mathrm{P}$

How do we obtain instance J from (H_J, G_J)?

Proof Ideas

Theorem

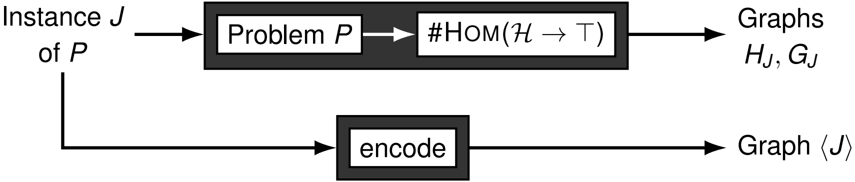
For any P in #W[1], there are $\mathcal{H}_P, \mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Approach:

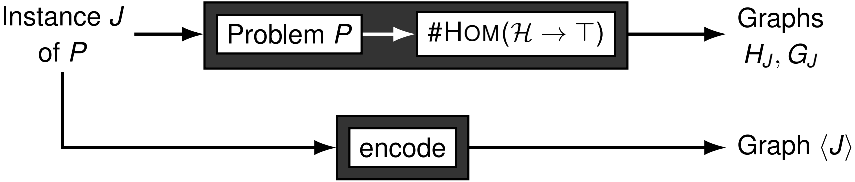
$$\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$$

$$\mathcal{G}_P := \{G_J \cup \langle J \rangle \mid \text{instance } J \text{ of } P\}$$

Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P, \mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Approach:

$$\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$$
$$\mathcal{G}_P := \{G_J \cup \langle J \rangle \mid \text{instance } J \text{ of } P\}$$

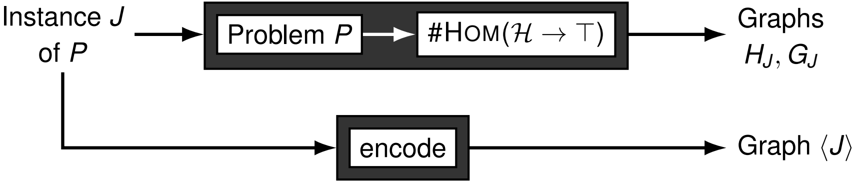
$P \preccurlyeq \#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ ✓

(ensure $\#\mathrm{Hom}(H_J \to \langle J \rangle) = 0$)

Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Approach:

$$\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$$
$$\mathcal{G}_P := \{G_J \cup \langle J \rangle \mid \text{instance } J \text{ of } P\}$$

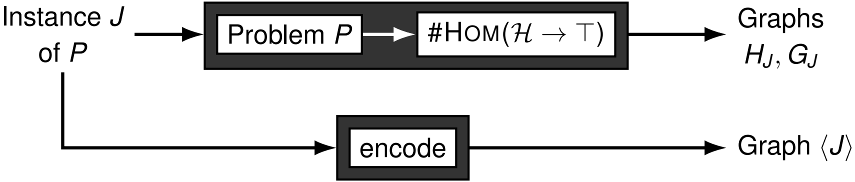
$P \preccurlyeq \#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ ✓
(ensure $\#\text{Hom}(H_J \to \langle J \rangle) = 0$)

$\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P) \overset{?}{\preccurlyeq} P$

Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Approach:

$$\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$$
$$\mathcal{G}_P := \{G_J \cup \langle J \rangle \mid \text{instance } J \text{ of } P\}$$

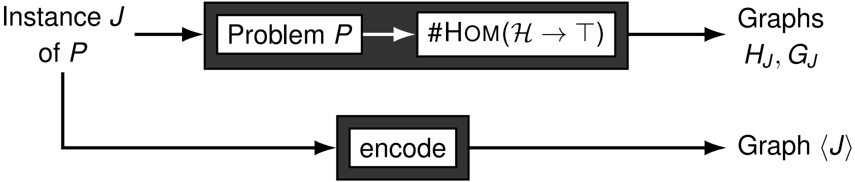
P $\preccurlyeq \#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ ✓
(ensure $\#\mathrm{Hom}(H_J \to \langle J \rangle) = 0$)

$\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P) \overset{?}{\preccurlyeq}$ P
How do we handle malformed input (H_J, G_L)?

Proof Ideas

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.



Approach:

$$\mathcal{H}_P := \{H_J \mid \text{instance } J \text{ of } P\}$$
$$\mathcal{G}_P := \{G_J \cup \langle J \rangle \mid \text{instance } J \text{ of } P\}$$

$P \preccurlyeq \#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$ ✓

(ensure $\#\mathrm{Hom}(H_J \to \langle J \rangle) = 0$)

$\#\textsc{Hom}(\mathcal{H}_P \to \mathcal{G}_P) \overset{?}{\preccurlyeq} P$

How do we ensure $\#\mathrm{Hom}(H_J \to G_L \cup \langle L \rangle) = 0$?

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$).

P $\preccurlyeq$ #HOM($\mathcal{H}_P \to \mathcal{G}_P$)	#HOM($\mathcal{H}_P \to \mathcal{G}_P$) $\preccurlyeq$ P
Can solve instance J with $(H_J, G_J \cup \langle J \rangle)$ by computing #Hom($H_J \to G_J \cup \langle J \rangle$)	Can extract instance J from pair $(H_J, G_J \cup \langle J \rangle)$
(ensuring #Hom($H_J \to \langle J \rangle$) = 0)	How do we ensure #Hom($H_J \to G_L \cup \langle L \rangle$) = 0?

Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$).

P $\preccurlyeq$ #HOM($\mathcal{H}_P \to \mathcal{G}_P$)

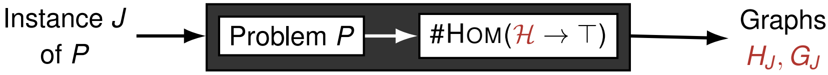
Can solve instance J with $(H_J, G_J \cup \langle J \rangle)$ by computing #Hom($H_J \to G_J \cup \langle J \rangle$)

(ensuring #Hom($H_J \to \langle J \rangle$) = 0)

#HOM($\mathcal{H}_P \to \mathcal{G}_P$) $\preccurlyeq$ P

Can extract instance J from pair $(H_J, G_J \cup \langle J \rangle)$

How do we ensure #Hom($H_J \to G_L \cup \langle L \rangle$) = 0?



Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$).

P $\preccurlyeq$ #HOM($\mathcal{H}_P \to \mathcal{G}_P$)

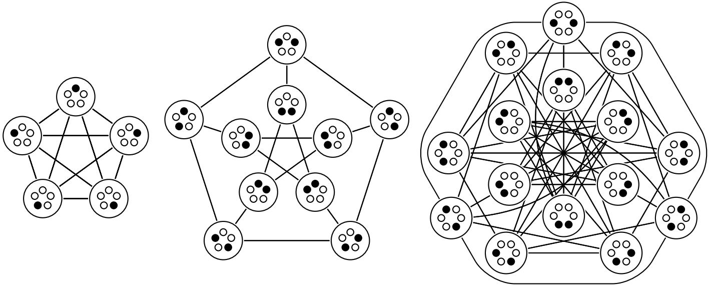
Can solve instance J with $(H_J, G_J \cup \langle J \rangle)$ by computing #Hom($H_J \to G_J \cup \langle J \rangle$)

(ensuring #Hom($H_J \to \langle J \rangle$) = 0)

#HOM($\mathcal{H}_P \to \mathcal{G}_P$) $\preccurlyeq$ P

Can extract instance J from pair $(H_J, G_J \cup \langle J \rangle)$

How do we ensure #Hom($H_J \to G_L \cup \langle L \rangle$) = 0?



Theorem

For any P in #W[1], there are $\mathcal{H}_P$, $\mathcal{G}_P$ such that P is equivalent to $\#\text{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$.

$P \preccurlyeq \#\text{Hom}(\mathcal{H}_P \to \mathcal{G}_P)$

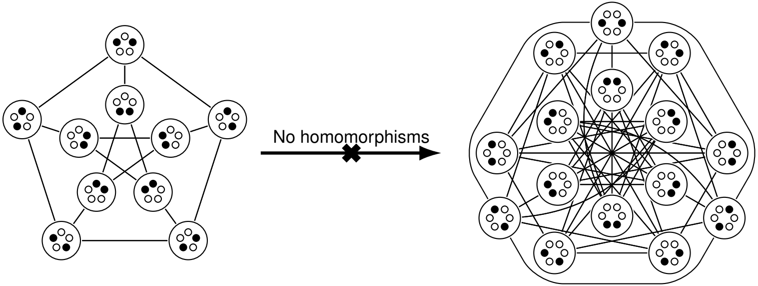
Can solve instance J with $(H_J, G_J \cup \langle J \rangle)$ by computing $\#\text{Hom}(H_J \to G_J \cup \langle J \rangle)$

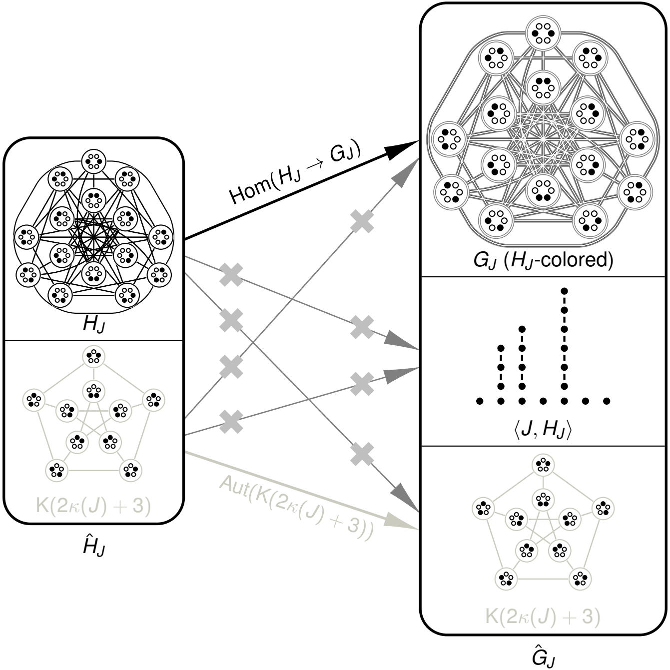
(ensuring $\#\text{Hom}(H_J \to \langle J \rangle) = 0$)

$\#\text{Hom}(\mathcal{H}_P \to \mathcal{G}_P) \preccurlyeq P$

Can extract instance J from pair $(H_J, G_J \cup \langle J \rangle)$

How do we ensure $\#\text{Hom}(H_J \to G_L \cup \langle L \rangle) = 0$?





Hom($H_J \to G_J$)
H_J
K($2\kappa(J)+3$)
$\hat{H}_J$
G_J (H_J-colored)
$\langle J, H_J \rangle$
K($2\kappa(J)+3$)
$\hat{G}_J$
Aut(K($2\kappa(J)+3$))

Main Result

#Hom($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Theorem ✓

For any problem P in #W[1] (or W[1]), there are graph classes $\mathcal{H}_P$ and $\mathcal{G}_P$ such that P is equivalent to #Hom($\mathcal{H}_P \to \mathcal{G}_P$) (or Hom($\mathcal{H}_P \to \mathcal{G}_P$)).

- Cannot hope for clear categorization into FPT/W[1]-hard for all pairs $(\mathcal{H}, \mathcal{G})$

$\rightsquigarrow$ Need to look at specific pairs of graph classes

Main Result

#HOM($\mathcal{H} \to \mathcal{G}$) Parameter: $|V(H)|$

Given graphs $H \in \mathcal{H}$ and $G \in \mathcal{G}$, count all graph homomorphisms from H to G.

Theorem ✓

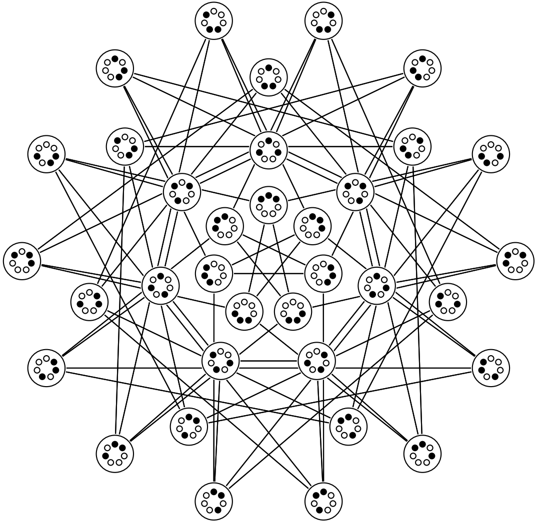
For any problem P in #W[1] (or W[1]), there are graph classes $\mathcal{H}_P$ and $\mathcal{G}_P$ such that P is equivalent to #HOM($\mathcal{H}_P \to \mathcal{G}_P$) (or HOM($\mathcal{H}_P \to \mathcal{G}_P$)).

- Cannot hope for clear categorization into FPT/W[1]-hard for all pairs $(\mathcal{H}, \mathcal{G})$

$\rightsquigarrow$ Need to look at specific pairs of graph classes

Open Problems

- Can we find a "hierarchy" of homomorphism problems?
  $\#\text{HOM}(\mathcal{H}_1 \to \mathcal{G}_1)$ "$\leq$" $\#\text{HOM}(\mathcal{H}_2 \to \mathcal{G}_2)$ "$\leq$" $\cdots$ "$\leq$" $\#\text{HOM}(\top \to \top)$
- (Grunt work?) Obtain algorithms/hardness for specific pairs of graph classes $\mathcal{H}$, $\mathcal{G}$
  (Done for $\mathcal{G} = F$-colorable graphs, line graphs, claw-free graphs, ...)



Thank you!

TikZ code for Kneser graphs available on GitHub

`github.com/PH111P/tikz-kneser`

Navigation

Start

Graph hom's 1 | Graph hom's 2 | Graph hom's 3

Graph classes

Known results 1 | Known results 2 | Known results 3 | Known results 4

Main result

Proof ideas 1 | Proof ideas 2 | Proof ideas 3

Open Problems

End