# The Complexity of the Homomorphism Reconstructibility Problem

Finite and Algorithmic Model Theory 2025

Nina Runde

(Joint work with Jan Böker, Louis Härtel, Tim Seppelt and Christoph Standke)

 $F_1$  $F_2$  $F_3$ 





 $F_1$  $F_2$  $F_3$ 











$$\hom(\bullet, G) = 9$$

 $\hom(\bullet, G) = 9 \qquad \hom(\bullet - \bullet, G) = y \cdot 2$ 

$$\hom(\bullet, G) = 9 \qquad \hom(\bullet - \bullet, G) = y \cdot 2 \qquad \hom(\bullet - \bullet, G) = x \cdot 6$$

$$\hom(\bullet, G) = 9 \qquad \hom(\bullet - \bullet, G) = y \cdot 2 \qquad \hom(\bullet - \bullet, G) = x \cdot 6$$



$$\hom(\bullet, G) = 9 \qquad \hom(\bullet - \bullet, G) = y \cdot 2 \qquad \hom(\bullet - \bullet, G) = x \cdot 6$$



$$\hom(\bullet, G) = 9 \qquad \hom(\bullet - \bullet, G) = y \cdot 2 \qquad \hom(\bullet - \bullet, G) = x \cdot 6$$



 $\operatorname{HomRec}(\mathcal{F})$ 

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ 

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph G such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ?

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph G such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ?

Instance size:  $\sum_{i \in [m]} |F_i| + \log(c_i)$ 

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph G such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ?

Instance size:  $\sum_{i \in [m]} |F_i| + \log(c_i)$ 

Example:  $hom(\bullet, G) = c_1$ 

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph G such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ?

Instance size:  $\sum_{i \in [m]} |F_i| + \log(c_i)$ 

Example: hom(•, G) =  $c_1$ size  $n = \log(c_1) + 1$ 

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph G such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ?

Instance size:  $\sum_{i \in [m]} |F_i| + \log(c_i)$ 

Example: hom(•, G) =  $c_1$ size  $n = \log(c_1) + 1$ 

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph G such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ?

Instance size:  $\sum_{i \in [m]} |F_i| + \log(c_i)$ 

Example: hom( $\bullet$ , G) =  $c_1$ size  $n = \log(c_1) + 1$ 

HOMREC( $\mathcal{F}$ ) Input:  $(F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph G such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ?

BHOMREC( $\mathcal{F}$ ) Input:  $N, (F_1, c_1), ..., (F_m, c_m) \in \mathcal{F} \times \mathbb{N}$ Question: Is there a graph Gwith  $|V(G)| \leq N$  such that hom $(F_i, G) = c_i$  for all  $i \in [m]$ ? Instance size:  $\sum_{i \in [m]} |F_i| + \log(c_i)$ 

Example: hom(•, G) =  $c_1$ size  $n = \log(c_1) + 1$ 

Instance size:  $N + \sum_{i \in [m]} |F_i| + \log(c_i)$ 



Graph *H* is 3-colorable  $\iff hom(H, \clubsuit) > 0$ 



Graph *H* is not 3-colorable  $\iff hom(H, \clubsuit) = 0$ 



Graph *H* is not 3-colorable  $\iff hom(H, \clubsuit) = 0$ 

NON-3-COL HOMREC



Graph *H* is not 3-colorable  $\iff hom(H, \clubsuit) = 0$ 

NON-3-COL HOMREC

Input graph H



Graph *H* is not 3-colorable  $\iff hom(H, \clubsuit) = 0$ 

NON-3-COL HomRec

 $F_1 = H$   $c_1 = 0$ 

Input graph H





Input graph H  $F_1 = H \qquad c_1 = 0$   $F_2 = \bullet \qquad c_2 = 3$   $F_3 = \bullet \circ \circ \qquad c_3 = 6$ 

Graph class ${\cal F}$	Hom $\operatorname{Rec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$

Graph class ${\cal F}$	Hom $\operatorname{Rec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs		

Graph class ${\cal F}$	Hom $\operatorname{Rec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs	NP <sup>#P</sup> -hard	NP <sup>#P</sup> -complete

Graph class ${\cal F}$	Hom $\operatorname{Rec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs	$NP^{\#P} ext{-hard}$	NP <sup>#P</sup> -complete

 $coNP \subseteq PH \subseteq P^{\#P} \subseteq NP^{\#P} (= NP^{PP}) \subseteq PSPACE \subseteq NEXP$ 

Graph class ${\cal F}$	Hom $\operatorname{Rec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs	NP <sup>#P</sup> -hard	NP <sup>#P</sup> -complete
bounded tree-width		

Graph class ${\cal F}$	Hom $\operatorname{Rec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs	NP <sup>#P</sup> -hard	NP <sup>#P</sup> -complete
bounded tree-width	NP-hard	NP-hard

Graph class ${\cal F}$	$\operatorname{HomRec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs	$NP^{\#P} ext{-hard}$	NP <sup>#P</sup> -complete
bounded tree-width	NP-hard	NP-hard
finite set of graphs		

Graph class ${\cal F}$	$\operatorname{HomRec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs	NP <sup>#P</sup> -hard	NP <sup>#P</sup> -complete
bounded tree-width	NP-hard	NP-hard
finite set of graphs	NP-hard	

Graph class ${\cal F}$	$\operatorname{HomRec}(\mathcal{F})$	$\operatorname{BHomRec}(\mathcal{F})$
all graphs	NP <sup>#P</sup> -hard	NP <sup>#P</sup> -complete
bounded tree-width	NP-hard	NP-hard
finite set of graphs	NP-hard	not NP-hard (unless $P=NP$ )

# QPoly

#### Input:

 $a,b,c\in\mathbb{N}$ 

# Question:

Are there  $x, y \in \mathbb{N}$ 

such that

 $ax^2 + by = c$ 

## A bit less easy Reduction

QPOLY **Input:**   $a, b, c \in \mathbb{N}$  **Question:** Are there  $x, y \in \mathbb{N}$ such that  $ax^2 + by = c$ 





## **Tractability Results**

# **Tractability Results**

FPT algorithm

• FPT algorithm (for computable f, running time of  $f(\sum |F_i|) \cdot poly(n)$ )

- FPT algorithm (for computable f, running time of  $f(\sum |F_i|) \cdot poly(n)$ )
- One constraint tuple (*F*<sub>1</sub>, *c*<sub>1</sub>)

- FPT algorithm (for computable f, running time of  $f(\sum |F_i|) \cdot poly(n)$ )
- One constraint tuple ( $F_1, c_1$ )  $\checkmark$

- FPT algorithm (for computable f, running time of  $f(\sum |F_i|) \cdot poly(n)$ )
- One constraint tuple  $(F_1, c_1)$  🗸
- Multiple, connected graphs of the same size

- FPT algorithm (for computable f, running time of  $f(\sum |F_i|) \cdot poly(n)$ )
- One constraint tuple  $(F_1, c_1)$  🗸
- Multiple, connected graphs of the same size, but only if we look at subgraph counts instead of homomorphism counts

- FPT algorithm (for computable f, running time of  $f(\sum |F_i|) \cdot poly(n)$ )
- One constraint tuple  $(F_1, c_1)$  🗸

- FPT algorithm (for computable f, running time of  $f(\sum |F_i|) \cdot poly(n)$ )
- One constraint tuple  $(F_1, c_1)$  🗸
- That's it.





#### It's a really hard problem



It's a really hard problem (at least NP $^{\#P}$  and still NP-hard with massively restricted constraint graph classes).



It's a really hard problem (at least NP $^{\#P}$  and still NP-hard with massively restricted constraint graph classes).

Open questions!



It's a really hard problem (at least NP $^{\#P}$  and still NP-hard with massively restricted constraint graph classes).

Open questions!

How hard is the problem exactly? (we only know it's in NEXP)



It's a really hard problem (at least NP $^{\#P}$  and still NP-hard with massively restricted constraint graph classes).

Open questions!

- How hard is the problem exactly? (we only know it's in NEXP)
- Other tractability results? What about two constraints? Approximations?



It's a really hard problem (at least NP $^{\#P}$  and still NP-hard with massively restricted constraint graph classes).

Open questions!

- How hard is the problem exactly? (we only know it's in NEXP)
- Other tractability results? What about two constraints? Approximations?
- What happens if we have homomorphism density constraints instead of counts?