COMPLEXITY OF H-COLORING IN HEREDITARY GRAPH CLASSES

KAROLINA OKRASA
PAWEŁ RZĄŻEWSKI

UNIVERSITY OF WARSAW
&
WARSAN UNIVERSITY OF TECHNOLOGY

8th PCC, 14-18.09.2020

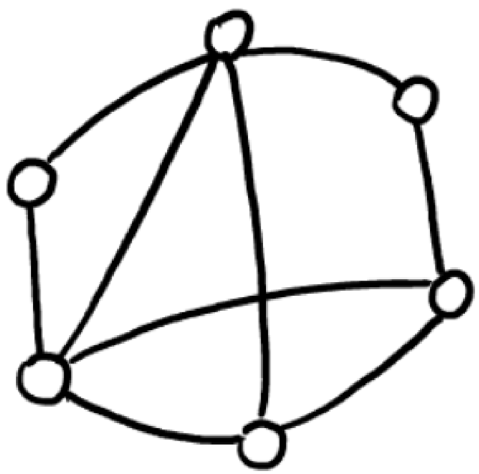
GRAPH HOMOMORPHISMS

- we have two graphs : G and H

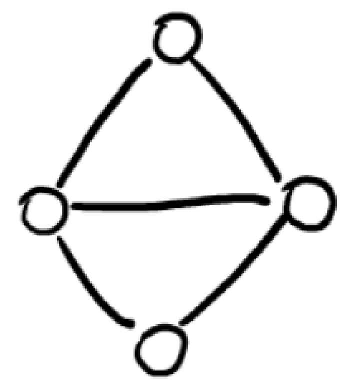
GRAPH HOMOMORPHISMS

- we have two graphs : G and H

e.g.



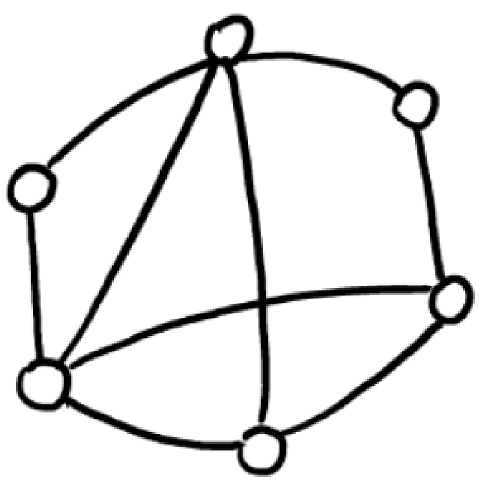
e.g.

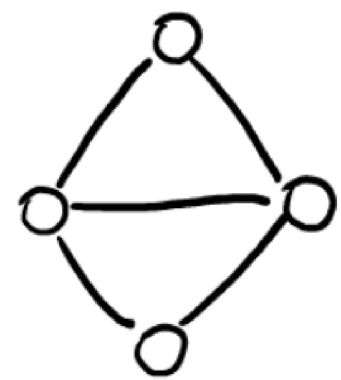


GRAPH HOMOMORPHISMS

- we have two graphs : G and H

e.g. e.g.





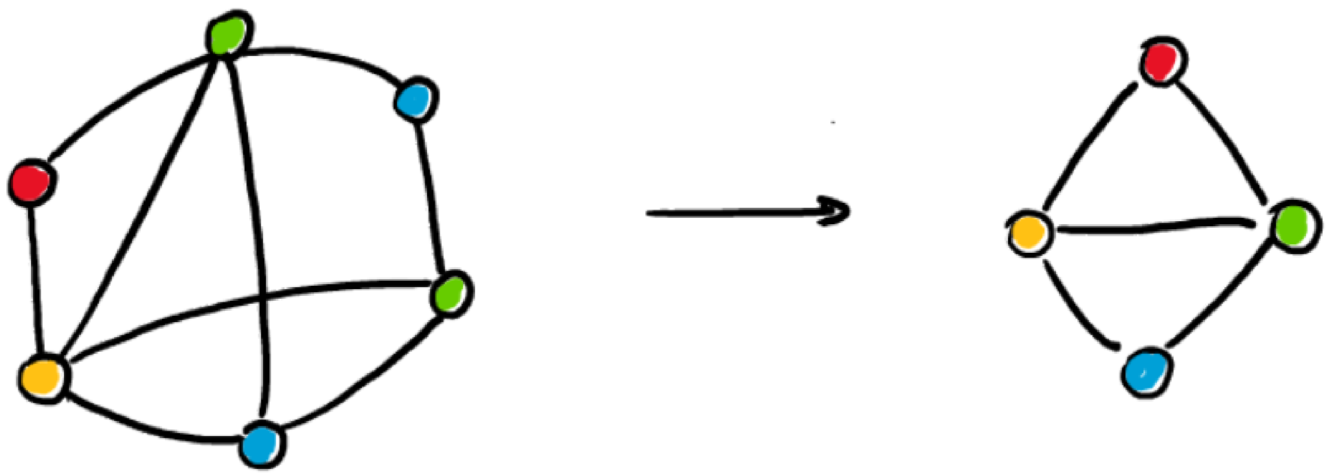
a **homomorphism** from G to H :

$$\boxed{f : V(G) \to V(H) \text{ s.t. } uv \in E(G) \Rightarrow f(u)\,f(v) \in E(H)}$$

GRAPH HOMOMORPHISMS

- we have two graphs: G and H

e.g. e.g.



(colors indicate the function f)

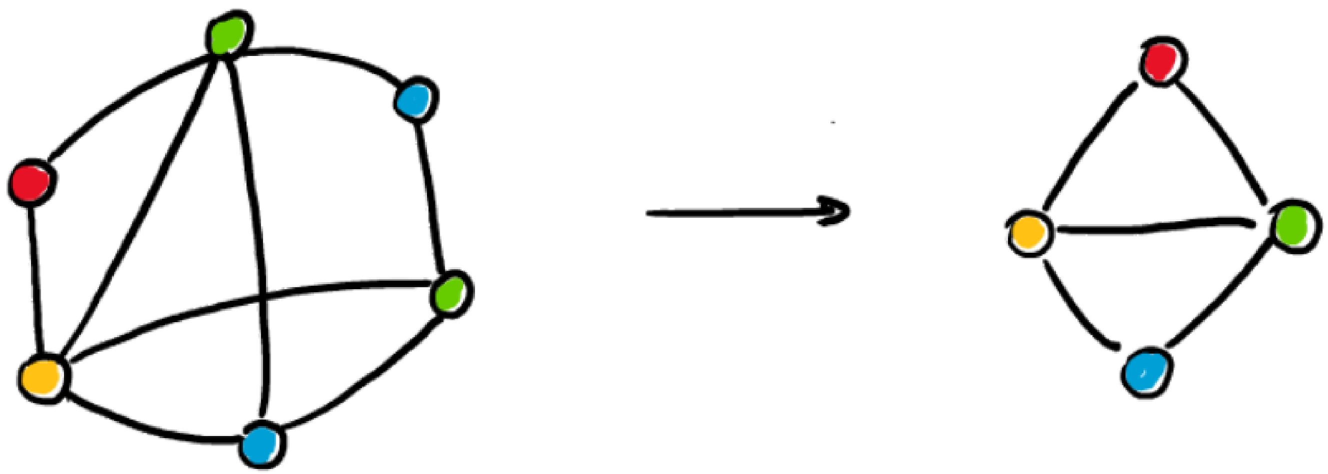
a **homomorphism** from G to H:

$$\boxed{f: V(G) \to V(H) \text{ s.t. } uv \in E(G) \Rightarrow f(u)f(v) \in E(H)}$$

GRAPH HOMOMORPHISMS

- we have two graphs : G and H

e.g. e.g.



(colors indicate the function f)

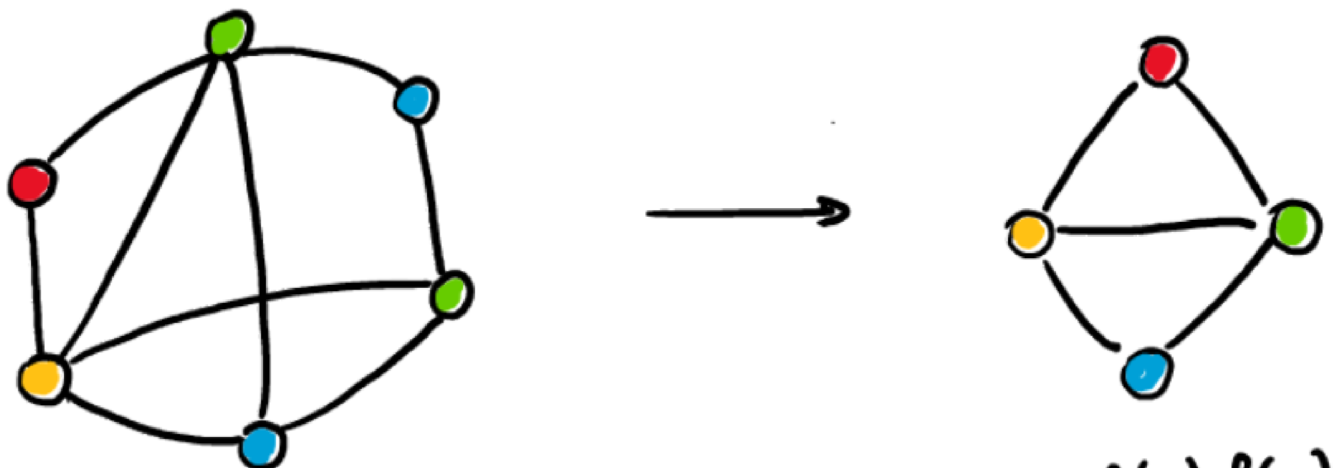
a homomorphism from G to H : (informally)

V(H) are colors
V(G) are colored by V(H)
in a way that colors can be adjacent in G only if they are adjacent in H

GRAPH HOMOMORPHISMS

- we have two graphs: G and H

e.g. e.g.



(colors indicate the function f)

$f: V(G) \to V(H)$ s.t. $uv \in E(G) \Rightarrow f(u)f(v) \in E(H)$

- no loops in $H \Rightarrow f$ is a proper $|H|$-coloring of G (possibly with additional restrictions, e.g., no red-blue edge)
- H is a k-clique $\Leftrightarrow f$ is a proper k-coloring of G
- $G \to H$: G admits a homomorphism to H

GRAPH HOMOMORPHISM PROBLEM

- fix graph H

GRAPH HOMOMORPHISM PROBLEM

- fix graph H
- for an instance graph G

GRAPH HOMOMORPHISM PROBLEM

- fix graph H
- for an instance graph G

w.l.o.g.
G connected

GRAPH HOMOMORPHISM PROBLEM

- fix graph H
- for an instance graph G

w.l.o.g. G connected

we want to check whether

$$G \to H$$

GRAPH HOMOMORPHISM PROBLEM

- fix graph H
- for an instance graph G

w.l.o.g. G connected

we want to check whether

$$G \to H$$

denote the problem by $\mathrm{HOM}(H)$

GRAPH HOMOMORPHISM PROBLEM

- fix graph H

w.l.o.g. G connected

- for an instance graph G

we want to check whether

$$G \to H$$

denote the problem by $\mathrm{HOM}(H)$

- $\mathrm{Hom}(K_k)$ problem $\equiv$ k-COLORING problem

GRAPH HOMOMORPHISM PROBLEM

- if H is bipartite or has a loop, we can solve HOM(H) in polynomial time (w.r.t. $|V(G)|$)
- otherwise, HOM(H) is NP-complete [1] & no subexponential-time algorithm assuming ETH [2]

GRAPH HOMOMORPHISM PROBLEM

- if H is bipartite or has a loop, we can solve HOM(H) in polynomial time (w.r.t. $|V(G)|$)
- otherwise, HOM(H) is NP-complete [1]
  & no subexponential-time algorithm assuming ETH [2]

subexponential time

e.g., $2^{O(\sqrt{n})}$, $2^{O(n^{0.9})}$ algorithms for instances on n vertices

VS

no subexponential time

no $2^{o(n)}$ algorithm assuming the Exponential Time Hypothesis (ETH-hard)

LIST HOMOMORPHISM PROBLEM

a generalization of HOM(H)
denoted by LHOM(H)

LIST HOMOMORPHISM PROBLEM

a generalization of $\mathrm{HOM}(H)$
denoted by $\mathrm{LHOM}(H)$

- each vertex v of instance G
  has a list $L(v) \subseteq V(H)$
  of allowed vertices

LIST HOMOMORPHISM PROBLEM

a generalization of $\mathrm{HOM}(H)$
denoted by $\mathrm{LHOM}(H)$

- each vertex v of instance G has a list $L(v) \subseteq V(H)$ of allowed vertices
- we look for homomorphism $f: G \to H$ such that $f(v) \in L(v)$

LIST HOMOMORPHISM PROBLEM

a generalization of HOM(H)
denoted by LHOM(H)

- each vertex v of instance G has a list $L(v) \subseteq V(H)$ of allowed vertices
- we look for homomorphism $f: G \to H$ such that $f(v) \in L(v)$
- LHOM(K_k) $\equiv$ LIST k-COLORING

LIST HOMOMORPHISM PROBLEM

a generalization of HOM(H)
denoted by LHOM(H)

- if H is bi-arc, then LHOM(H) polynomial-time solvable
- otherwise, LHOM(H) NP-complete

& no subexponential algorithm assuming ETH [3]

PROBLEM

$\mathcal{C}$ - some graph class

Can we solve $\text{Hom}(H) \mid \text{LHom}(H)$ faster assuming our instances are from $\mathcal{C}$?

PROBLEM

$\mathcal{C}$ - some graph class

Can we solve $\text{HOM}(H) \mid \text{LHOM}(H)$ faster assuming our instances are from $\mathcal{C}$?

What *faster* can mean?

PROBLEM

$\mathcal{C}$ - some graph class

Can we solve Hom(H) | LHom(H) faster assuming our instances are from $\mathcal{C}$?

What faster can mean?

- classic approach

  polynomial-time solvable vs NP-complete

- fine-grained approach

  subexponential-time solvable vs ETH-hard

FORBIDDEN STRUCTURES

F - fixed connected graph

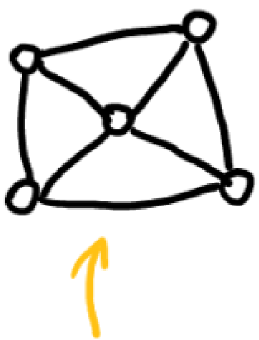
Graph G is F-free if it contains no F
as induced subgraph

FORBIDDEN STRUCTURES

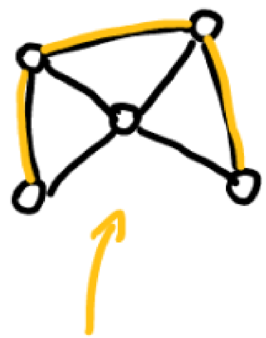
F - fixed connected graph

Graph G is F-free if it contains no F as induced subgraph

e.g.



P_4- free



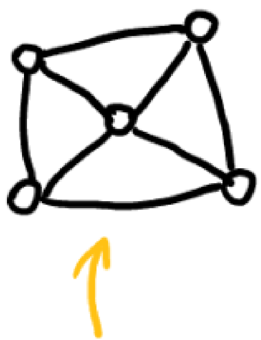
not P_4 -free

FORBIDDEN STRUCTURES

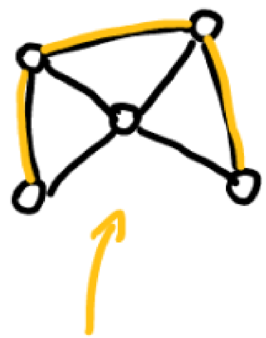
F - fixed connected graph

Graph G is F-free if it contains no F as induced subgraph

e.g.



P_4-free



not P_4-free

hereditary graph class:

G is F-free $\Rightarrow$ $G-v$ is F-free

PROBLEM

$\mathcal{C}$ - some graph class

Can we solve HOM(H) | LHOM(H) faster assuming our instances are from $\mathcal{C}$?

- For which pairs (F, H) we can solve HOM(H) and LHOM(H) faster if our instances are F-free?

PROBLEM

$\mathcal{C}$ - some graph class

Can we solve HOM(H) | LHOM(H) faster assuming our instances are from $\mathcal{C}$?

- For which pairs (F, H) we can solve HOM(H) and LHOM(H) faster if our instances are F-free?

e.g. (trivial)

$\mathcal{C} = \{G : G \text{ is } P_3\text{-free}\}$ ← cliques (we assumed G connected)

PROBLEM

$\mathcal{C}$ - some graph class

Can we solve HOM(H) | LHOM(H) faster assuming our instances are from $\mathcal{C}$?

- For which pairs (F, H) we can solve HOM(H) and LHOM(H) faster if our instances are F-free?

e.g. (trivial)

$\mathcal{C} = \{G : G \text{ is } P_3\text{-free}\}$ ← cliques (we assumed G connected)

if $G \in \mathcal{C}$ then

$G \to H \iff H$ has a clique of size $|V(G)|$

PROBLEM

$\mathcal{C}$ - some graph class
Can we solve HOM(H) | LHOM(H) faster
assuming our instances are from $\mathcal{C}$?

- For which pairs (F, H) we can solve HOM(H) and LHOM(H) faster if our instances are F-free ?

e.g. (trivial)

$\mathcal{C} = \{G : G \text{ is } P_3\text{-free}\}$ ← cliques (we assumed G connected)

if $G \in \mathcal{C}$ then

$G \to H \iff H$ has a clique of size $|V(G)|$

since H fixed, we check this in constant time $\Rightarrow$ HOM(H) trivial for (P_3, H) (for any H)

PROBLEM

$\mathcal{C}$ - some graph class

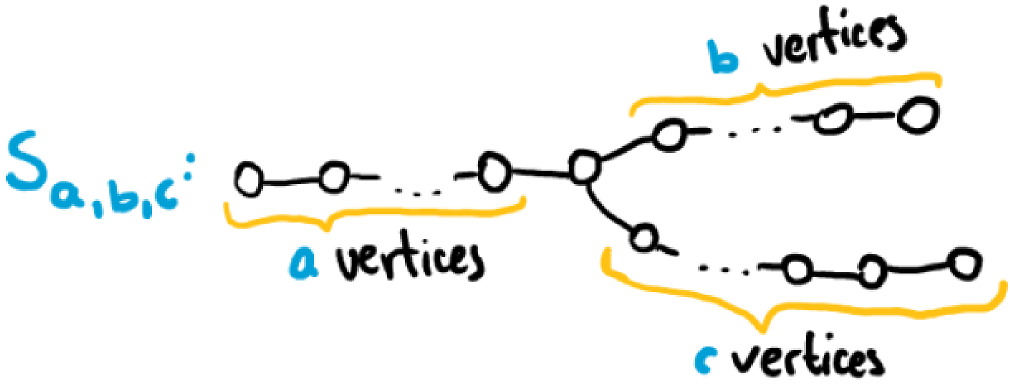
Can we solve HOM(H) | LHOM(H) faster assuming our instances are from $\mathcal{C}$?

- For which pairs (F, H) we can solve HOM(H) and LHOM(H) faster if our instances are F-free?

e.g. (non-trivial)

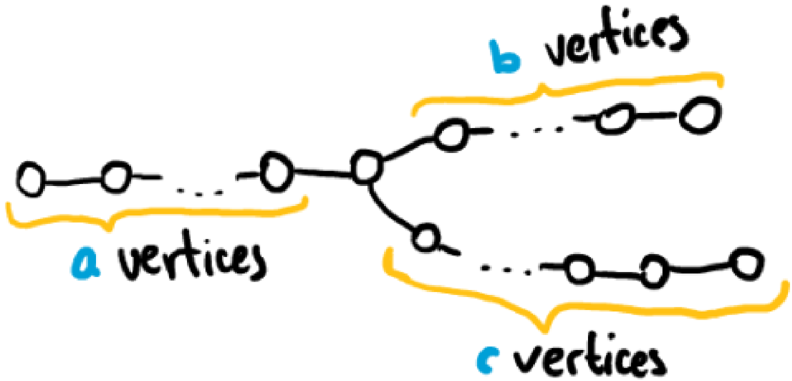
HOM(K_3) and LHOM(K_3) polynomial-time solvable in P_7-free graphs [4]

WHAT WE KNEW



WHAT WE KNEW

$S_{a,b,c}$:

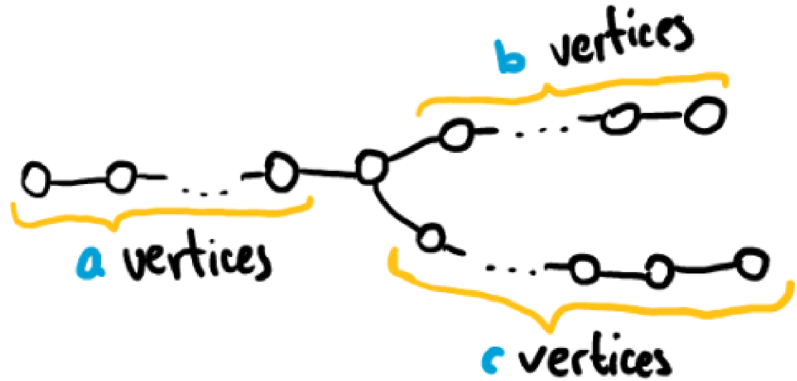


H non-bi-arc, $F \neq P_t$, $F \neq S_{a,b,c}$ [5]

$\rightarrow$ LHom(H) ETH-hard even in F-free graphs

WHAT WE KNEW

$S_{a,b,c}$:



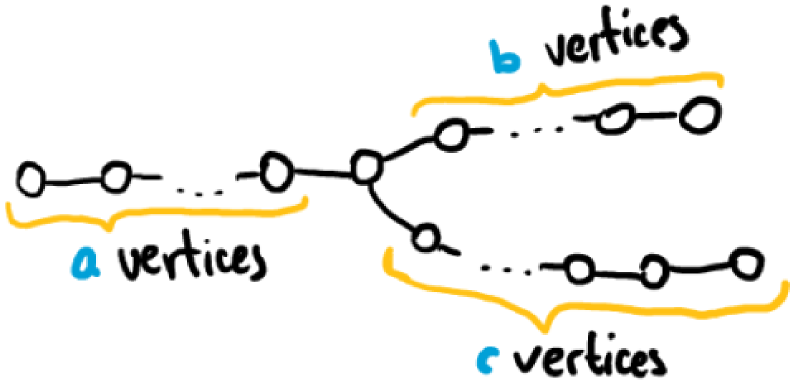
Remaining cases:

- complexity of HOM(H) in F-free graphs*

*some partial results are known

WHAT WE KNEW

$S_{a,b,c}$:



Remaining cases:

- complexity of $\text{HOM}(H)$ in F-free graphs*
- complexity of $\text{LHOM}(H)$ in P_t-free graphs or in $S_{a,b,c}$-free graphs

  algorithm for $\text{LHOM}(H)$ would work for $\text{HOM}(H)$

*some partial results are known

WHAT WE DID

P_t-free graphs:

for any H we defined a family $\mathcal{H}$ of factors of H and identified a property χ such that

WHAT WE DID

P_t-free graphs:

for any H we defined a family $\mathcal{H}$ of factors of H and identified a property $\times$ such that

- if there is $H' \in \mathcal{H}$ with property $\times$ then there is t' such that LHOM (H) is ETH-hard in P_t-free graphs for $t \geq t'$,

WHAT WE DID

P_t-free graphs:

for any H we defined a family $\mathcal{H}$ of factors of H and identified a property $\times$ such that

- if there is $H' \in \mathcal{H}$ with property $\times$ then there is t' such that LHOM(H) is ETH-hard in P_t-free graphs for $t \geq t'$,
- if there is no $H' \in \mathcal{H}$ with property $\times$ then we can solve LHOM(H) in time $2^{O(\log^2 |V(G)|)}$ in P_t-free graphs, for every fixed t.

WHAT WE DID

P_t-free graphs:

for any H we defined a family $\mathcal{H}$ of factors of H and identified a property $\times$ such that

- if there is $H' \in \mathcal{H}$ with property $\times$ then there is t' such that LHOM (H) is ETH-hard in P_t-free graphs for $t \geq t'$,
- if there is no $H' \in \mathcal{H}$ with property $\times$ then we can solve LHOM (H) in time $2^{O(\log^2 |V(G)|)}$ in P_t-free graphs, for every fixed t.

$S_{a,b,c}$-graphs:

- if $K_3 \leq H$, then LHOM(H) generalizes 3-COLORING

induced subgraph

$S_{a,b,c}$-graphs:

- if $K_3 \leq H$, then LHOM(H) generalizes 3-COLORING
- 3-COLORING ETH-hard in line grephs

induced subgraph

$S_{a,b,c}$-graphs:

- if $K_3 \leq H$, then LHOM(H) generalizes 3-COLORING
- 3-COLORING ETH-hard in line graphs
- line graphs are $S_{1,1,1}$-free

induced subgraph

$S_{a,b,c}$-graphs:

- if $K_3 \leq H$, then LHOM(H) generalizes 3-COLORING
- 3-COLORING ETH-hard in line graphs
- line graphs are $S_{1,1,1}$-free

induced subgraph

$\Rightarrow$ if $K_3 \leq H$, then LHOM(H) ETH-hard in $S_{a,b,c}$-free graphs for any $a,b,c \geq 1$

$S_{a,b,c}$-graphs:

- if $K_3 \leq H$, then LHOM(H) generalizes 3-COLORING
- 3-COLORING ETH-hard in line graphs
- line graphs are $S_{1,1,1}$-free

induced subgraph

$\Rightarrow$ if $K_3 \leq H$, then LHOM(H) ETH-hard in $S_{a,b,c}$-free graphs for any $a,b,c \geq 1$

- so we can assume $K_3 \not\leq H$

but H still can have e.g., 

$S_{a,b,c}$-graphs:

- we assume $K_3 \not\subseteq H$ and consider loopless H

$S_{a,b,c}$-graphs:

- we assume $K_3 \not\subseteq H$ and consider loopless H
- we find a special case of extended strip decomposition [6] of G if G is $S_{a,b,c}$-free and K_3-free and a subexponential-time algorithm for $\mathrm{LHOM}(H)$

$S_{a,b,c}$-graphs:

- we assume $K_3 \not\subseteq H$ and consider loopless H
- we find a special case of extended strip decomposition of G if G is $S_{a,b,c}$-free and K_3-free and a subexponential-time algorithm for LHOM(H)

$\Rightarrow$ we solve LHOM(H) in subexponential-time

$S_{a,b,c}$-graphs:

- we assume $K_3 \not\leq H$ and consider loopless H
- we find a special case of extended strip decomposition of G if G is $S_{a,b,c}$-free and K_3-free and a subexponential-time algorithm for $\textsc{LHom}(H)$

$\Rightarrow$ we solve $\textsc{LHom}(H)$ in subexponential-time

if $K_3 \leq G$ then no-instance
(no $\triangle \to H$)

if $K_3 \not\leq G$
we use the algorithm

$S_{a,b,c}$-graphs:

- if H contains loops $u,v,w \in V(H)$ and no $x,y \in \{u,v,w\}$ s.t. $N(x) \subseteq N(y)$ we can show there exist a',b',c' s.t. $LHom(H)$ ETH-hard in $S_{a,b,c}$-free graphs for $a \geq a'$, $b \geq b'$ and $c \geq c'$.

$S_{a,b,c}$-graphs:

- if H contains loops $u, v, w \in V(H)$ and no $x, y \in \{u, v, w\}$ s.t. $N(x) \subseteq N(y)$ we can show there exist a', b', c' s.t. $LHom(H)$ ETH-hard in $S_{a,b,c}$-free graphs for $a \geq a'$, $b \geq b'$ and $c \geq c'$.
- satisfied for all reflexive graphs H s.t. $LHom(H)$ is NP-complete

$S_{a,b,c}$-graphs:

- if H contains loops $u, v, w \in V(H)$ and no $x, y \in \{u, v, w\}$ s.t. $N(x) \subseteq N(y)$ we can show there exist a', b', c' s.t. $\mathrm{LHom}(H)$ ETH-hard in $S_{a,b,c}$-free graphs for $a \geq a'$, $b \geq b'$ and $c \geq c'$.

each vertex has a loop

- satisfied for all reflexive graphs H s.t. $\mathrm{LHom}(H)$ is NP-complete

$S_{a,b,c}$-graphs:

- if H contains loops $u,v,w \in V(H)$ and no $x,y \in \{u,v,w\}$ s.t. $N(x) \subseteq N(y)$ we can show there exist a',b',c' s.t. $LHom(H)$ ETH-hard in $S_{a,b,c}$-free graphs for $a \geq a'$, $b \geq b'$ and $c \geq c'$.

each vertex has a loop

- satisfied for all reflexive graphs H s.t. $LHom(H)$ is NP-complete

open case: mixed graphs H (with loops just on some vertices)

[1] Hell, Nešetřil, JCTB 1990

[2] Cygan et al, SODA 2016

[3] Feder, Hell, Huang, JGT 2003

[4] Bonomo et al, Combinatorica 2018

[5] Piecyk, Rzążewski (pers. communication)

[6] Chudnovsky et al, SODA 2020