Warsaw University of Technology

FACULTY OF
MATHEMATICS AND INFORMATION SCIENCE

Master's diploma thesis

in the field of study Mathematics

Complexity of variants of graph homomorphism problem in selected graph classes

Karolina Okrasa

student record book number 254071

thesis supervisor

Paweł Rzążewski, PhD

WARSAW 2019

....

supervisor's signature

....

author's signature

Abstract

Complexity of variants of graph homomorphism problem in selected graph classes

A homomorphism is a function which maps vertices of a graph G to vertices of a graph H in which each edge of G is mapped to some edge of H. For a fixed graph H, by $\textsc{Hom}(H)$ we denote the computational problem of deciding whether a given graph G admits a homomorphism to H. Graph homomorphisms are generalization of graphs colorings, as if H is a complete graph on k vertices, then $\textsc{Hom}(K_k)$ is equivalent to k-COLORING. A result of Hell and Nešetřil states that if H is bipartite or has a vertex with a loop then $\textsc{Hom}(H)$ is polynomial-time solvable and otherwise it is NP-complete.

In this thesis we consider complexity bounds of NP-complete cases of $\textsc{Hom}(H)$, parameterized by the treewidth of the instance graph G. Using both algebraic and combinatorial tools, we show that for almost all graphs H the complexity obtained by a straightforward dynamic programming on a tree decomposition of G cannot be improved, unless the Strong Exponential Time Hypothesis (a standard assumption from the complexity theory) fails.

In the second part of the thesis, we analyse the cases of graphs H for which the bound obtained by the dynamic programming method can be improved. We prove another lower bound with an additional restriction on H and show that it is tight for all graphs H, if we assume two conjectures from algebraic graph theory. In particular, there are no known graphs H which are not covered by our result.
Keywords: graph homomorphisms, treewidth, fine-grained complexity, projective graphs

Streszczenie

Złożoność wariantów problemu homomorfizmu w wybranych klasach grafów

Funkcję f, która wierzchołkom grafu G przyporządkowuje wierzchołki grafu H w taki sposób, że jeśli uv jest krawędzią w G, to $f(u)f(v)$ jest krawędzią w H, nazywamy homomorfizmem z G w H. Dla ustalonego grafu H przez $\textsc{Hom}(H)$ oznaczamy problem decyzyjny, w którym pytamy, czy dany graf G ma homomorfizm w H. Homomorfizmy grafów są pewnym uogólnieniem problemu kolorowania grafów – jeśli H jest grafem pełnym o k wierzchołkach, wówczas $\textsc{Hom}(K_k)$ jest równoważne problemowi k-COLORING, w którym pytamy, czy dany graf G da się poprawnie pokolorować na k kolorów. Hell i Nešetřil udowodnili, że jeśli H jest grafem dwudzielnym lub zawiera wierzchołek z pętlą, wtedy $\textsc{Hom}(H)$ można rozwiązać w czasie wielomianowym, a w przeciwnym wypadku problem ten jest NP-zupełny.

W niniejszej pracy pokazujemy ścisłe ograniczenia na złożoność obliczeniową problemu $\textsc{Hom}(H)$ (w przypadkach, dla których jest NP-zupełny), w zależności od liczby wierzchołków i szerokości drzewowej instancji G. Używając zarówno narzędzi kombinatorycznych, jak i algebraicznych, dowodzimy, że, przy standardowych założeniach teorii złożoności, programowanie dynamiczne na dekompozycji drzewowej grafu daje algorytm optymalny dla prawie wszystkich grafów H.

W drugiej części pracy analizujemy znane przypadki $\textsc{Hom}(H)$, które można rozwiązać szybciej i pokazujemy znajdujemy optymalną złożoność problemu przy dodatkowym założeniu na graf H. Dowodzimy również, że znaleziona złożoność jest optymalna dla wszystkich grafów H, przy założeniu dwóch znanych hipotez z algebraicznej teorii grafów. W szczególności oznacza to, że nie jest znany żaden graf H, dla którego nasze twierdzenie nie zachodzi.
Słowa kluczowe: homomorfizmy grafów, szerokość drzewowa, drobnoziarnista analiza złożoności, grafy idempotentnie trywialne

Warsaw, ...

Declaration

I hereby declare that the thesis entitled "Complexity of variants of graph homomorphism problem in selected graph classes", submitted for the Master degree, supervised by Paweł Rzążewski, PhD, is entirely my original work apart from the recognized reference.

....

Contents

1. Introduction **13**

1.1. Motivation 13

1.2. Our results and organization of the thesis 15

2. Definitions and preliminaries **17**

2.1. Homomorphisms and cores 17

2.2. Graph products and projectivity 21

3. Complexity of the Hom(H) problem **25**

4. Lower bounds **28**

4.1. Projective graphs 28

4.2. Non-projective graphs 30

5. Summary **37**

Bibliography **40**

To Paweł,

for his advice, his patience and his faith in me,

for being the best supervisor, the best teacher and the best friend.

Karolina

1. Introduction

1.1. Motivation

For two graphs G and H, consider a function f, which maps vertices of G to vertices of H in a way that if uv is an edge of G, then $f(u)f(v)$ is an edge of H. We call such a function f a *homomorphism* from G to H and call H the *target* of the homomorphism (see Figure 1.1). Graph homomorphisms are a natural generalization of graph colorings, as the existence of a k-coloring of any graph G is equivalent to the existence of a homomorphism from G to the complete graph K_k. So, intuitively, we can think about a homomorphism to the target H as a coloring in which adjacent vertices must receive colors which form an edge in H.

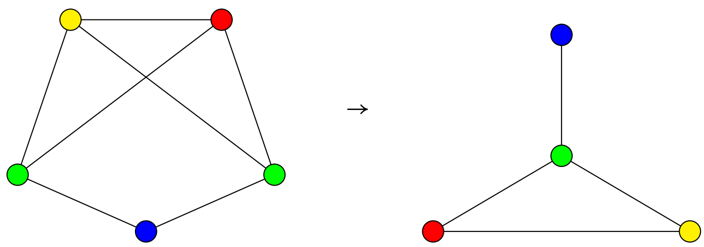

Figure 1.1: An example of a homomorphism from a graph G (left) to the target graph H (right). Colors of the vertices indicate the mapping. Note that G cannot have, for example, a blue-yellow edge.

For a fixed graph H, by $\textsc{Hom}(H)$ we denote the computational problem of deciding whether a given graph G admits a homomorphism to H. Clearly, the computational problem of deciding if a given graph G is k-colorable, denoted by k-COLORING, is equivalent to $\textsc{Hom}(K_k)$. The k-COLORING problem is one of the most known and best studied graph problems – a classical result states that it is polynomial-time solvable if $k \leq 2$ and NP-complete if $k \geq 3$ (see [13]). This was generalized to $\textsc{Hom}(H)$ by Hell and Nešetřil in [17]. They obtained a full complexity dichotomy, i.e., proved that if H is bipartite or has a vertex with a loop, then $\textsc{Hom}(H)$ is polynomial-time solvable, and otherwise it is NP-complete.

Assuming that P ≠ NP, we know that problems that are NP-hard cannot be solved in polynomial time. However, for many hard problems, algorithms better than a straightforward brute-force approach exist. For example, if we consider the k-COLORING problem, it is known that there are algorithms which solve it in time $c^n \cdot n^{\mathcal{O}(1)}$, for some constant $c > 0$ which does not depend on k. Currently, the best known bound is $2^n \cdot n^{\mathcal{O}(1)}$, and was obtained by Björklund, Husfeldt, and Koivisto [3]. So when we consider graph homomorphisms, a natural question arises: does there exist an absolute constant c such that for every H the Hom(H) problem can be solved in time $c^n \cdot n^{\mathcal{O}(1)}$? And, if not – how to prove such a result?

To prove a tight lower bound for running times of the algorithms, we need to introduce a stronger hypothesis than P ≠ NP. Two standard assumptions used commonly in the complexity theory, are the Exponential Time Hypothesis (ETH) and the Strong Exponential Time Hypothesis (SETH), both conjectured by Impagliazzo and Paturi [21, 20]. The first one implies that 3-Sat with n variables and m clauses cannot be solved in time $2^{o(n+m)}$, while the second one implies that Cnf-Sat with n variables and m clauses cannot be solved in time $(2-\epsilon)^n \cdot m^{\mathcal{O}(1)}$, for any $\epsilon > 0$. Cygan et al. [7] showed that assuming the ETH, there is no algorithm for Hom(H) working in time $c^n \cdot n^{\mathcal{O}(1)}$, for any constant $c > 0$ which does not depend on H.

One of the possible research directions is analysing how the complexity depends on some other structural parameter of the input instance, which may contain additional relevant information. One of the most widely-studied parameters is the *treewidth* of a graph, denoted by $\text{tw}(G)$, which we formally define in chapter 2. We can solve many classic NP-hard problems, like Independent Set, Dominating Set or Hamiltonian Cycle in time $f(\text{tw}(G)) \cdot n^{\mathcal{O}(1)}$, where f is some computable function (see [2, 8]), which means in particular that if an instance graph has bounded treewidth, we can solve the mentioned problems in polynomial time.

Of course, it is important to understand how the optimal function f can look. For Hom(H), the standard dynamic programming on a tree decomposition of the instance graph G gives us the complexity $|H|^{\text{tw}(G)} \cdot n^{\mathcal{O}(1)}$, assuming that an optimal tree decomposition of G is given [4, 8]. On the other hand, there is a result by Lokshtanov, Marx, and Saurabh which shows that if we assume the SETH, then, at least for complete graphs, we cannot hope for a significant improvement.

Theorem 1.1 (Lokshtanov et al. [26]). *Let $k \geq 3$. The k-coloring problem cannot be solved in time $(k-\epsilon)^t \cdot n^d \cdot c$ for graphs on n vertices and treewidth t, for any constants $c, d > 0$, any $\epsilon > 0$, unless the SETH fails.*

The main purpose of this thesis is to investigate if this bound can be somehow extended to other target graphs.

Observe that, while working on graph colorings, a similar, more general problem can be considered – a setting in which every vertex of the instance graph has its own list of allowed colors, and we ask for a coloring in which every vertex receives a color from its list. We can define such a variant also for graph homomorphism problem. Consider the computational problem LHom(H) in which the instance consists of a graph G whose every vertex v is equipped with a list $L(v)$ of vertices of H. We ask if there exists a homomorphism f from G to H such that for every v we have $f(v) \in L(v)$. Observe that while Hom(H) is trivial when H contains a vertex with a loop, this is not the case for LHom(H), as this vertex does not have to appear in all lists. Feder and Hell proved that if H is reflexive (which means that every vertex has a loop), then LHom(H) is polynomial-time solvable if H is an interval graph, and NP-complete otherwise [12]. Investigating the tight complexity bound of reflexive cases of LHom(H), parameterized by treewidth, Egri, Marx and Rzążewski defined a new graph invariant $i^*(H)$, based on incomparable sets of vertices and a new graph decomposition, and proved the following.

Theorem 1.2 (Egri, Marx, Rzążewski [10]). *Let H be a fixed, non-interval, reflexive graph with $i^*(H) = k$. Let n and t be, respectively, the number of vertices and the treewidth of an instance graph G.*

(a) Assuming a tree decomposition of G of width t is given, the LHom*(H) problem can be solved in time $k^t \cdot n^d \cdot c$, for some constants $c, d > 0$.*

(b) There is no algorithm solving LHom*(H) in time $(k-\epsilon)^t \cdot n^d \cdot c$ for any $\epsilon > 0$, and any constants $c, d > 0$, unless the SETH fails.*

Although Hom(H) and LHom(H) look quite similar, the techniques used to show lower bounds are very different. This is because in LHom(H) it is sufficient to perform a reduction for some induced subgraph H' of H, since every instance of LHom(H') is also an instance of LHom(H), in which the vertices which do not belong to H' do not appear on lists. On the other hand, in LHom(H) we need to capture the structure of the whole graph H, which is difficult using combinatorial tools only. This is why typical tools used in this area come from abstract algebra and algebraic graph theory.

1.2. Our results and organization of the thesis

Chapter 2 contains the important definitions and notation. In chapter 3 we introduce the upper bounds for Hom(H) and prove, in particular, that in some cases this problem can be solved faster than $|H|^{\mathrm{tw}(G)} \cdot n^{\mathcal{O}(1)}$. Also, we explain why we investigate only restricted classes of target graphs H, i.e., connected *cores*. A core is a graph which does not admit a homomorphism into any of its proper subgraphs.

Chapter 4 is split in two main parts. First, we consider the class of projective graphs H (which is formally defined in the chapter 2) and prove that for projective cores, which are non-trivial (i.e., have at least three vertices), the result given by the standard dynamic programming is asymptotically tight, assuming the SETH.

Theorem 1.3. *Let H be a fixed non-trivial projective core on k vertices and let n and t be, respectively, the number of vertices and the treewidth of an instance graph G.*

(a) *Assuming a tree decomposition of G of width t is given, the* Hom*(H) problem can be solved in time $k^t \cdot n^d \cdot c$, where $c, d > 0$ are constants.*

(b) *There is no algorithm to solve* Hom*(H) in time $(k-\epsilon)^t \cdot n^d \cdot c$ for any $\epsilon > 0$, and any constants $c, d > 0$, unless the SETH fails.*

The main tool used in the proof of the lower bound is the construction of a so-called *edge gadget*, which allows us to perform an elegant reduction from k-coloring. There, an edge gadget is a graph F with two specified vertices u^* and v^*, such that:

(a) for any distinct vertices x, y of H, there is a homomorphism from F to H, which maps u^* to x and v^* to y, and

(b) in any homomorphism from F to H, the vertices u^* and v^* are mapped to distinct vertices of H.

In the reduction, we take an instance G of the k-coloring problem and replace every edge xy of G by the edge gadget, unifying u^* with x and v^* with y. This way we create an instance G^* of Hom(H), such that G^* is a yes-instance of Hom(H) if and only if G is a yes-instance of k-coloring. Also, the treewidth of G differs only by an additive constant from the treewidth of G^*, which, by Theorem 1.1, is enough to prove the tight lower bound, assuming the SETH.

It is worth to note that, although the definition of projective graphs may look restricted, in fact Theorem 1.3 works for the wide class of graphs. In particular, it is true that asymptotically almost all graphs are projective cores. Indeed, Łuczak and Nešetřil proved in [27] that asymptotically almost all graphs are projective. On the other hand, it is known that asymptotically almost all graphs are cores (see Corollary 3.28 in the book of Hell and Nešetřil, [19]). One can verify that by some basic probability laws this implies that asymptotically almost all graphs are projective cores. This, combined with Theorem 1.3 (b) gives us the following corollary.

Corollary 1.4. *Assuming the SETH, for almost all graphs H on k vertices there is no algorithm solving* Hom*(H) in time $(k-\epsilon)^t \cdot n^d \cdot c$ for any $\epsilon > 0$, and any constants $c, d > 0$, for instance graphs on n vertices and treewidth t.*

In the second part of chapter 4 we analyse the lower bounds on the complexity of the remaining cases of Hom(H), i.e., when H is a non-projective graph. We show that the approach from the previous case cannot be extended for non-projective targets H. Then we investigate a class of non-projective graphs for which there exist algorithms working faster than $|H|^{\mathrm{tw}(G)} \cdot n^{\mathcal{O}(1)}$ and prove a matching lower bound in some cases. The reduction in section 4.2 is very similar to the first one but more technically involved. Then in chapter 5 we show how our results are related to open problems studied in the literature.

2. Definitions and preliminaries

For an integer n we denote by $[n]$ the set of integers $\{1,\dots,n\}$. All graphs considered in this paper are finite, undirected and with no multiple edges, but loops are allowed, unless stated otherwise. For a graph G, we denote by $V(G)$ and $E(G)$, respectively, the set of vertices and the set of edges of G, and by $\omega(G)$, $\chi(G)$, and $\mathrm{og}(G)$, respectively, the size of the largest clique contained in G, the chromatic number of G, and the odd girth of G. Also, by $|G|$ we denote the number of vertices of G. Let K_1^* be the single-vertex graph with a loop. We say that a graph G is *ramified* if there is no pair of distinct vertices u,v in G, such that the neighborhood of u is a subset of the neighborhood of v.

A *tree decomposition* of a graph G is a pair $\left(\mathcal{T},\{X_a\}_{a\in V(\mathcal{T})}\right)$, in which $\mathcal{T}$ is a tree, whose vertices are called *nodes* and $\{X_a\}_{a\in V(\mathcal{T})}$ is the family of subsets (called *bags*) of $V(G)$, such that

1. every $v\in V(G)$ belongs to at least one bag X_a,
2. for every $uv\in E(G)$ there is at least one bag X_a such that $u,v\in X_a$,
3. for every $v\in V(G)$ the set $\mathcal{T}_v := \{a\in V(\mathcal{T}) \mid v\in X_a\}$ induces a connected subgraph of $\mathcal{T}$.

The *width* of a tree decomposition $\left(\mathcal{T},\{X_a\}_{a\in V(\mathcal{T})}\right)$ is the number $\max_{a\in V(\mathcal{T})}|X_a|-1$. The minimum possible width of a tree decomposition of G is called the *treewidth* of G and denoted by $\mathrm{tw}(G)$.

2.1. Homomorphisms and cores

For graphs G and H, a function $f:V(G)\to V(H)$ is a *homomorphism*, if it preserves edges, i.e., for every $uv\in E(G)$ it holds that $f(u)f(v)\in E(H)$ (see Figure 1.1). If G admits a homomorphism to H, we denote this fact by $G\to H$ and we write $f:G\to H$ if f is a homomorphism from G to H. If there is no homomorphism from G to H, we write $G\not\to H$. Since graph homomorphisms generalize graph colorings, we call $f:G\to H$ an H-coloring of G and refer to the vertices of H as colors. Graphs G and H are *homomorphically equivalent* if $G\to H$ and $H\to G$, and *incomparable* if $G\not\to H$ and $H\not\to G$. Observe that homomorphic equivalence is an equivalence relation on the class of all graphs, because identity mapping is always a homomorphism and also the composition of homomorphisms is a homomorphism. We say that f is an *endomorphism* of G if f is a homomorphism from G to G.

A graph G is a *core* if $G \not\to H$ for every proper subgraph H of G. Equivalently, we can say G is a core if and only if every endomorphism of G is an automorphism. If H is a subgraph of G such that $G \to H$ and H is a core, we say that H is a core of G. Notice that if H is a subgraph of G, then it always holds that $H \to G$, so every graph is homomorphically equivalent to its core. Moreover, if H is a core of G, then H is always an induced subgraph of G, because every endomorphism $f : G \to H$ restricted to H must be an automorphism. Also, because of that, if $f : G \to H$ is a homomorphism from G to its core H, then it must be surjective. Observe that if there exists a bijective homomorphism from G to H, then G must be a spanning subgraph of H.

Observe that if a graph has two cores, they must be isomorphic.

Observation 2.1 (Hell, Nešetřil, [18]). *Every graph has a unique core, up to isomorphism.*

Proof. Assume that H_1 and H_2 are cores of G. Clearly, G and H_1 are homomorphically equivalent, and so are G and H_2. We observed that homomorphic equivalence is transitive, so H_1 and H_2 are also homomorphically equivalent. It means that there exist $f : H_2 \to H_1$ and $g : H_1 \to H_2$. Clearly, $f \circ g$ is an endomorphism of H_1, so, because H_1 is a core, it is an automorphism. It means that g must be injective and f must be surjective. Analogously, H_2 is a core, so $g \circ f$ must be an automorphism of H_2, which means that f must be injective and g must be surjective. Combining these results we get that f and g are bijective homomorphisms, so H_1 and H_2 must be isomorphic. □

Simple examples of cores are complete graphs and odd cycles [15]. We say that a core is *trivial* if it is isomorphic to K_1, K_1^*, or K_2.

Observation 2.2. *Let H be a core graph. Then H is trivial if and only if it has fewer that three vertices.*

Proof. Clearly, if H is trivial, then it has at most two vertices. To see the "if" part, note that if H has a loop, then we have $G \to H$ for every graph G, because a function which maps all vertices of G to the vertex with a loop is a homomorphism. So K_1^* is the only core which contains a loop. Since K_1 and K_2 are cores, the only remaining case of a graph on fewer than 3 vertices is an edgeless graph on 2 vertices, but clearly its core is K_1. □

Observe that if a graph has no edges, then its core is isomorphic to K_1. Assuming that G is bipartite but has at least one edge, we know that the core of G has at least two vertices, because $G \not\to K_1$ and $K_1^* \not\to G$ (and G must be homomorphically equivalent to its core). Clearly, $G \to K_2$, because $\chi(G) = 2$, and any 2-coloring of G is a homomorphism into K_2, which is an induced subgraph of G. This means that K_2 must be the core of G. We summarize this as follows.

Observation 2.3. *Let G be a graph, whose core H is trivial.*

(a) $H \simeq K_1^*$ *if and only if* G *has a loop,*

(b) $H \simeq K_1$ *if and only if* G *has no edges,*

(c) $H \simeq K_2$ *if and only if* G *is bipartite and has at least one edge.* □

If we assume that G and H are loopless, then the following conditions are necessary for G to have a homomorphism into H.

Observation 2.4. *Assume that* $G \to H$ *and* G *and* H *have no loops. Then* $\omega(G) \le \omega(H)$, $\chi(G) \le \chi(H)$ *and* $\mathrm{og}(G) \ge \mathrm{og}(H)$. *In particular, if* H *is a core of* G, *then* $\omega(G) = \omega(H)$, $\chi(G) = \chi(H)$ *and* $\mathrm{og}(G) = \mathrm{og}(H)$.

Proof. Let $f : G \to H$ and let K be the set of $\omega(G)$ vertices of G inducing a clique. Clearly, f must map K to $\omega(G)$ distinct vertices of H, as otherwise some edge would be mapped to a loop. On the other hand, if there exist vertices $a, b \in f(K)$ such that $ab \notin E(H)$, then $xy \notin E(G)$ for some $x, y \in K$ such that $f(x) = a$ and $f(y) = b$. So $f(K)$ is a clique of size $\omega(G)$ in H, which means that $\omega(H) \ge \omega(G)$.

Note that for every graph G with $\chi(G) > 1$ it holds that $G \to K_{\chi(G)}$, but also $G \not\to K_{\chi(G)-1}$. Suppose that $\chi(G) > \chi(H)$. It implies that there exists a homomorphism $g : H \to K_{\chi(G)-1}$. However, since the composition of homomorphisms is also a homomorphism, if $f : G \to H$, then we have that $g \circ f : G \to K_{\chi(G)-1}$, a contradiction.

To see that $\mathrm{og}(G) \ge \mathrm{og}(H)$, first observe that an odd cycle cannot be mapped to a bipartite graph, because this would mean that odd cycles are 2-colorable. Also, observe that an odd cycle $C_{2\ell+1}$ can be mapped to C_{2r+1} if and only if $\ell \ge r$. Indeed, assume that $\ell \ge r$. If we denote by $x_1, \dots, x_{2\ell+1}$ and by $y_1, \dots, y_{2r+1}$, respectively, the consecutive vertices of $C_{2\ell+1}$ and C_{2r+1}, we can define a homomorphism $g : C_{2\ell+1} \to C_{2r+1}$ as follows:

$$g(x_i) := \begin{cases} y_i & \text{if } i \le 2r+1, \\ y_1 & \text{if } i \text{ is even and } i > 2r+1, \\ y_{2r+1} & \text{if } i \text{ is odd and } i > 2r+1. \end{cases}$$

On the other hand, if there exists $g : C_{2\ell+1} \to C_{2r+1}$ and $\ell < r$, then clearly the set $g(V(C_{2\ell+1}))$ induces a proper subgraph of C_{2r+1} because $C_{2\ell+1}$ has fewer vertices. But each proper subgraph of an odd cycle is bipartite, and we already observed that odd cycle cannot be mapped to a bipartite graph. So if $f : G \to H$ and $C_{2\ell+1}$ is an odd cycle in G, its image under f must contain an odd cycle of at most the same length, which means that $\mathrm{og}(G) \ge \mathrm{og}(H)$.

If H is a core of G, then there exists $g : H \to G$, so we must have also $\omega(H) \le \omega(G)$, $\chi(H) \le \chi(G)$ and $\mathrm{og}(H) \ge \mathrm{og}(G)$. □

We denote by $H_1 + \ldots + H_m$ the disconnected graph with connected components $H_1, \dots, H_m$.

Observation 2.5. *Let f be a homomorphism from G to H. Then f must map each connected component of G into some connected component of H.*

Proof. Let G_1 be some connected component of G and let H_1 be some connected component of H. For contradiction, suppose, without the loss of generality, that there exist $x, y \in V(G_1)$ such that $f(x) \in V(H_1)$ and $f(y) \notin V(H_1)$. As G_1 is connected, there is a path in G_1 connecting x and y. It must contain two consecutive vertices u and v such that $f(u) \in V(H_1)$ and $f(v) \notin V(H_1)$. But then f cannot map the edge uv to an edge of H, because there is no edge between H_1 and vertices from the set $V(H) - V(H_1)$, so f is not a homomorphism, a contradiction. □

Observe that the graph does not have to be connected to be a core.

Observation 2.6. *A disconnected graph is a core if and only if its connected components are pairwise incomparable cores.*

Proof. Let $H = H_1 + \ldots + H_m$. First, observe that if H is a core, then $H_1, \ldots, H_m$ must be cores. Indeed, without the loss of generality, if H_1' is a proper subgraph of H_1 and there exists $f : H_1 \to H_1'$, then $g : V(H) \to V(H_1' + H_2 + \ldots + H_m)$ defined by

$$g(x) := \begin{cases} f(x) & \text{if } x \in V(H_1), \\ x & \text{otherwise,} \end{cases} \tag{2.1}$$

is a homomorphism from H to its proper subgraph, a contradiction. Similarly, we can prove that, without the loss of generality, $H_1 \not\to H_2$. Since H_2 is a proper subgraph of H, if there exists a homomorphism $f : H_1 \to H_2$, then there exists a homomorphism $g : H \to H_2 + H_3 + \ldots + H_m$ defined in the same way as (2.1). It means that H admits a homomorphism to its proper subgraph, a contradiction. By symmetry, $H_i \not\to H_j$ for every $i \neq j$.

Now let $H_1, \ldots, H_m$ be pairwise incomparable cores. Suppose that H' is a core of H and consider $f : H \to H'$. Recall that f must be surjective and maps every H_i to some connected component of H'. Note that H' must contain vertices from every H_i. In other case some H_i would be mapped by f to some subgraph of H_j, for $j \neq i$, but this cannot happen, because H_i and H_j are incomparable. It means that H' must have at least m connected components. On the other hand, since f is surjective and H has m connected components, H' must have at most m connected components. So it has exactly m connected components. Denote by $H_1', \ldots, H_m'$, respectively, the connected components of H' induced by the vertices from $H_1, \ldots, H_m$. Observe that, since for every $i \in [m]$ graph H_i is incomparable with every H_j for $i \neq j$, $f|_{H_i}$ must be in fact a homomorphism into H_i'. This means that if H_i' is a proper subgraph of H_i, then H_i admits a homomorphism into its proper subgraph, which is a contradiction. So $f|_{H_i}$ must be an isomorphism between H_i and H_i'. It implies that $H_1 + \ldots + H_m$ must be isomorphic to $H_1' + \ldots + H_m'$. □

An example of a pair of incomparable cores is shown on the Figure 2.1. It is the *Grötzsch graph*, denoted by G_G, and the clique K_3. The Grötzsch graph is a core, because it is vertex-critical, i.e., its every proper induced subgraph has a lower chromatic number [6]. This, by Observation 4.4, implies that there is no homomorphism from G_G into any of its proper subgraphs. Clearly, $\mathrm{og}(G_G) > \mathrm{og}(K_3)$ and $\chi(G_G) > \chi(K_3)$, so by Observation 2.4, they are incomparable. Observation 2.6 implies that graph $G_G + K_3$ is a core.

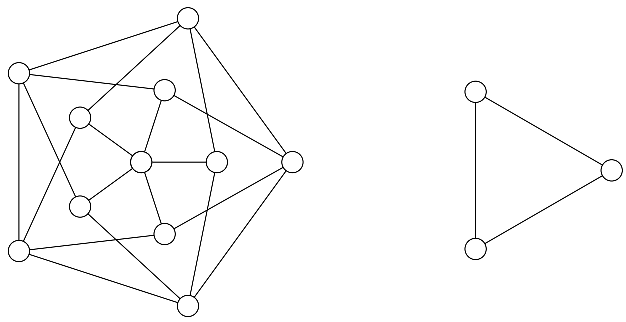

Figure 2.1: An example of incomparable cores, the *Grötzsch graph* (left) and K_3 (right).

Observe that for every non-trivial core H we can find a core H' such that H and H' are incomparable. In fact, we can construct an arbitrarily large families of pairwise incomparable cores, starting from an arbitrary non-trivial core H_0. A classic result of Erdős [11] states that for every positive integers $\ell, r > 0$ there exists graph H with $\mathrm{og}(H) > \ell$ and $\chi(H) > r$. So if $H_0, \ldots, H_m$ is already a family of incomparable cores, there exists a graph H such that $\mathrm{og}(H) > \max_{i \in \{0,\ldots,m\}} \mathrm{og}(H_i)$ and $\chi(H) > \max_{i \in \{0,\ldots,m\}} \chi(H_i)$. By Observation 2.4, H is incomparable with H_i for every $i \in \{0, \ldots, m\}$. Denote by H_{m+1} the core of H. It is homomorphically equivalent with H, so must be also incomparable with H_i, for every $i \in \{0, \ldots, m\}$, which means that $H_0, \ldots, H_m, H_{m+1}$ is also a family of pairwise incomparable cores.

Also note that if H is a trivial core, then for every graph H' it always holds that $H \to H'$ or $H' \to H$. In particular, there are no disconnected cores with trivial components.

2.2. Graph products and projectivity

Define the *direct product* of graphs H_1 and H_2, denoted by $H_1 \times H_2$, as follows:

$$V(H_1 \times H_2) = \{(x, y) \mid x \in V(H_1) \text{ and } y \in V(H_2)\}$$

and

$$E(H_1 \times H_2) = \{(x_1, y_1)(x_2, y_2) \mid x_1x_2 \in E(H_1) \text{ and } y_1y_2 \in E(H_2)\}.$$

We call H_1 and H_2 the *factors* of $H_1 \times H_2$. Clearly, the binary operation $\times$ is commutative, so we can identify $H_1 \times H_2$ and $H_2 \times H_1$. Since $\times$ is also associative, we can extend the definition for more than two factors:

$$H_1 \times \cdots \times H_{m-1} \times H_m := (H_1 \times \cdots \times H_{m-1}) \times H_m.$$

We say that $H_1 \times \ldots \times H_m$ is a *factorization* of H. If H is isomorphic to $H_1 \times \ldots \times H_m$ for some $m \geq 2$ and every factor H_i has at least three vertices, we call $H_1 \times \ldots \times H_m$ a *non-trivial factorization* of H.

The direct product appears in the literature under different names: *tensor product*, *cardinal product*, *Kronecker product*, *relational product*. It is also called *categorical product*, because it is the product in the category of graphs (see [16, 29] for details).

In the next chapters we sometimes consider the products of factors which are products themselves, so the vertices of such graphs are formally tuples of tuples. If it does not lead to confusion, for $\bar{x} := (x_1, \ldots, x_{k_1})$ and $\bar{y} := (y_1, \ldots, y_{k_2})$, we will treat tuples $(\bar{x}, \bar{y})$, $(x_1, \ldots, x_{k_1}, y_1, \ldots, y_{k_2})$, $(\bar{x}, y_1, \ldots, y_{k_2})$, and $(x_1, \ldots, x_{k_1}, \bar{y})$ as equivalent. This notation is generalized to more factors in a natural way.

Observe that a graph $H_1 \times H_2$ with at least one edge is isomorphic to H_1 if and only if H_2 is isomorphic to K_1^*. We say that a graph H is *directly indecomposable* (or *indecomposable* for short) if the fact that H is isomorphic to $H_1 \times H_2$ implies that H_1 or H_2 is isomorphic to K_1^*. If H is isomorphic to $H_1 \times \ldots \times H_m$ and each factor H_i is directly indecomposable and not isomorphic to K_1^*, we call $H_1 \times \ldots \times H_m$ a *prime factorization* of H. In particular, if $m = 1$, then H is indecomposable, otherwise, assuming $H \not\simeq K_1^*$, we call H *decomposable*. Clearly, K_1^* does not have a prime factorization.

The property of uniqueness is very useful, when dealing with graph factorizations (see also Theorem 8.17 in [16]). Note that, in particular, factors can have loops.

Theorem 2.7 (McKenzie, [28]). *Any connected non-bipartite graph with more than one vertex has a unique prime factorization into directly indecomposable factors.*

The following theorem is a corollary from the result of Weichsel [32] (see Corollary 5.10 in [16]).

Theorem 2.8 (Weichsel, [32]). *Let $H_1, \ldots, H_m$ be connected graphs, such that $|H_i| > 1$ for every $i \in [m]$. Graph $H_1 \times \ldots \times H_m$ is connected if and only if at most one factor H_i is bipartite.*

Let $H_1 \times \ldots \times H_m$ be some factorization of H (not necessary prime) and let $i \in [m]$. A function $\pi_i : V(H) \to V(H_i)$ such that for every $(x_1, \ldots, x_m) \in V(H)$ it holds that $\pi_i(x_1, \ldots, x_m) = x_i$ is a *projection on the i-th coordinate.* It follows from the definition of the product that every projection π_i is a homomorphism from H to H_i.

We denote by H^m the product of m copies of H. Below we summarize some basic properties of direct products.

Observation 2.9. *Let H be a graph on k vertices. Then*

(a) the core of $H \times K_1$ is K_1,

(b) if H has at least one edge, then the core of $H \times K_2$ is K_2,

(c) the graph H^m contains an induced subgraph isomorphic to H. In particular, if $m \geq 2$, then H^m is never a core,

(d) if $H_1 \times \ldots \times H_m$ is the factorization of H, then for every graph G it holds that

$$G \to H \text{ if and only if } G \to H_i \text{ for every } i \in [m],$$

Proof. (a) Observe that $H \times K_1$ is an edgeless graph on k vertices, so by Observation 2.3 (b) its core is isomorphic to K_1.

(b) If we denote by y_1 and y_2 the vertices of K_2, observe that $\{(x_i, y_1) \mid x_i \in H\}$ and $\{(x_i, y_2) \mid x_i \in H\}$ induce the bipartition classes of $H \times K_2$. As H has at least one edge, so does $H \times K_2$. Since $H \times K_2$ is a bipartite graph with at least one edge, from Observation 2.3 (c) we know that its core is K_2.

(c) Note that H^m contains a subgraph induced by vertices $\{(x, \ldots, x) \mid x \in V(H)\}$, which must be isomorphic to H. The projection $\pi_1 : H^m \to H$ is a homomorphism, so if $m \geq 2$, then for sure H^m is not a core.

(d) Assume that $f : G \to H$. Clearly, $H \to H_i$ for every $i \in [m]$ because each projection $\pi_i : H \to H_i$ is a homomorphism. So $\pi_i \circ f$ is a homomorphism from G to H_i. On the other hand, if we have some $f_i : G \to H_i$ for every $i \in [m]$, then we can define $f : G \to H$ by $f(x) := (f_1(x), \ldots, f_m(x))$. □

A homomorphism $f : H^m \to H$ is *idempotent*, if for every $x \in V(H)$ it holds that $f(x, x, \ldots, x) = x$. We say that H is *projective* (or *idempotent trivial*), if for every $m \geq 2$ every idempotent homomorphism from H^m to H is a projection. The projectivity property in graphs seems to be related to many areas of study and plays a key role in some important proofs (see for example [1, 30, 14]).

Observe that if H is a projective core, then homomorphisms from H^m to H have a very specific form.

Observation 2.10. *If H is a projective core and $f : H^m \to H$ is a homomorphism, then $f \equiv g \circ \pi_i$ for some $i \in [m]$ and some automorphism g of H.*

Proof. If f is idempotent, then it is a projection and we are done. Assume f is not idempotent and define a function $g : V(H) \to V(H)$ by $g(x) = f(x, \ldots, x)$. Function g is an endomorphism of H and H is a core, this implies that g is in fact an automorphism of H. Observe that $g^{-1} \circ f$ is an idempotent homomorphism, so it is equal to π_i for some $i \in [m]$, because H is projective. From this we get that $f \equiv g \circ \pi_i$. □

Observe that the definition of projective graphs does not imply that recognizing graphs with this property is decidable. However, an algorithm to recognize these graphs follows from the useful characterization, obtained by Larose and Tardif.

Theorem 2.11 (Larose, Tardif, [24]). *A connected graph H on at least three vertices is projective if and only if every idempotent homomorphism from H^2 to H is a projection.*

It appears that the properties of projectivity and being a core are indepentent. Figure 2.2 shows an example of a projective graph G, which is not a core. Clearly, G can be mapped to a triangle, which is its proper subgraph, so it is not a core. However, Larose [23] proved that all non-bipartite, ramified, connected graphs which do not contain C_4 as a (not necessarily induced) subgraph, are projective (later we will use his stronger result, which implies this one, see Theorem 5.2).

On the other hand, there are cores which are not projective (such an example is $G_G \times K_3$), and we analyse them in section 4.2. Also, it is known that projective graphs are always connected [24] and in Observation 2.6 we proved that there exist disconnected cores.

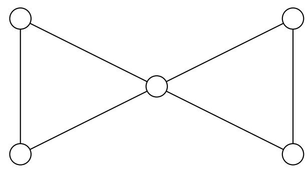

Figure 2.2: An example of a projective graph which is not a core.

3. Complexity of the HOM(H) problem

In this chapter we introduce upper bounds for the complexity of HOM(H). Note that if two graphs H_1 and H_2 are homomorphically equivalent, then HOM(H_1) and HOM(H_2) problems are also equivalent. So in particular, because every graph is homomorphically equivalent to its core, we may restrict our attention to graphs H which are cores. Also, from Observation 2.3 we know that HOM(H) can be solved in polynomial time if H is isomorphic to K_1^*, K_1 or K_2. Recall from the introduction that Hell and Nešetřil in [17] proved that if H is bipartite or has a vertex with a loop then HOM(H) is polynomial-time solvable and otherwise it is NP-complete. So we are interested only in cores H for which the HOM(H) problem is NP-complete, i.e., cores on at least three vertices. In particular we assume that H is not bipartite and has no loops.

We investigate the complexity of the HOM(H) problem, parameterized by treewidth of the input graph. The standard dynamic programming approach (see for example Cygan et al., [8]) gives us the following upper bound.

Theorem 3.1. *Let H be a fixed graph on k vertices and let n and t be, respectively, the number of vertices and the treewidth of the instance graph G. Assuming a tree decomposition of width t of G is given, the* HOM*(H) problem can be solved in time $k^t \cdot n^d \cdot c$ for some constants $c, d > 0$.*

As mentioned in the introduction (see Theorem 1.1), it is known that this bound is tight under the SETH for complete graphs on at least three vertices, i.e. for the k-COLORING problem. On the other hand, there are graphs for which we can obtain a better upper bound. Consider the family of decomposable graphs, defined in the previous chapter. Recall from Observation 2.9 (d) that if H has a prime factorization $H_1 \times \ldots \times H_m$, then for every graph G it holds that

$$G \to H \text{ if and only if } G \to H_i \text{ for every } i \in [m].$$

So consider an algorithm which takes an instance G of HOM(H), solves HOM(H_i) on the same instance G for each $i \in [m]$ and returns a positive answer for HOM(H) if and only if it gets a positive answer for each HOM(H_i). This way we obtain the following result.

Theorem 3.2. *Let H be a fixed core with prime factorization $H_1 \times \ldots \times H_m$. Let n and t be, respectively, the number of vertices and the treewidth of an instance graph G. Assuming the tree decomposition of width t of G is given, the* HOM*(H) problem can be solved in time $\max_{j \in [m]} |H_j|^t \cdot n^d \cdot c$ for some constants $c, d > 0$.* □

In the next chapter we analyse for which graphs H we can prove that assuming the SETH the bound in Theorem 3.2 is tight. Now we conclude this chapter with an observation about the complexity of Hom(H) for disconnected cores H.

Theorem 3.3. *Let H be a disconnected core isomorphic to $H_1 + \ldots + H_m$. Let n and t be, respectively, the number of vertices and the treewidth of an instance graph G. Assume that a tree decomposition of G of width t is given.*

(a) If for every $i \in [m]$ the Hom*(H_i) problem can be solved in time $\alpha^t \cdot n^d \cdot c$, where $\alpha, c, d > 0$ are constants, then the* Hom*(H) problem can also be solved in time $\alpha^t \cdot n^d \cdot c'$ for some constant $c' > 0$.*

(b) If the Hom*(H) problem can be solved in time $\alpha^t \cdot n^d \cdot c$, where $\alpha, c, d > 0$ are constants, then for every $i \in [m]$ the* Hom*(H_i) problem can also be solved in time $\alpha^t \cdot n^d \cdot c'$ for some constant $c' > 0$.*

Proof. First, observe that if G is disconnected and its connected components are $G_1, \ldots, G_\ell$, then $G_1 + \ldots + G_\ell \to H$ if and only if $G_i \to H$ for every $i \in [\ell]$. Also, $\mathrm{tw}(G) = \max_{i \in [\ell]} \mathrm{tw}(G_i)$. It means that if the instance graph is disconnected, we can just consider the problem separately for each of its connected components.

So we assume that G is connected. First, observe that $G \to H$ if and only if $G \to H_i$ for some $i \in [m]$. Indeed, if $G \to H_i$ for some $i \in [m]$ then clearly $G \to H$. On the other hand, if $G \to H$, then, because G is connected, we use Observation 2.5 and get that $G \to H_i$ for some $i \in [m]$. So we can solve Hom(H_i) for each H_i for the same instance G and return a positive answer for Hom(H) if and only if we get a positive answer of Hom(H_i) for at least one $i \in [m]$. We do it in time $\alpha^t \cdot n^d \cdot c'$ for $c' := c \cdot m$. This proves (a).

To see (b), let G be an instance of Hom(H_i) on n vertices and treewidth t. Let $V(H_i) = \{z_1, \ldots, z_k\}$ and let u be some fixed vertex of G. We construct an instance G^* of Hom(H) by taking a copy G' of G and a copy $\widetilde{H}_i^k$ of H_i^k and identifying the vertex corresponding to u in G' and the vertex corresponding to $(z_1, \ldots, z_k)$ in $V(\widetilde{H}_i^k)$. Denote this vertex of G^* by $\widetilde{z}$. Observe that H_i is connected and non-trivial, so Theorem 2.8 implies that $\widetilde{H}_i^k$ is connected. Since G is also connected, G^* must be connected.

We claim that $G \to H_i$ if and only if $G^* \to H$. Indeed, if $f : G \to H_i$, then there exists $j \in [k]$ such that $f(\widetilde{z}) = z_j$, so we can define a homomorphism $g : G^* \to H_i$ (which is also a homomorphism from G^* to H) by

$$g(v) = \begin{cases} f(v) & \text{if } v \in G', \\ \pi_j(v) & \text{if } v \in \widetilde{H}_i^k. \end{cases}$$

Clearly, both f and π_j are homomorphisms and $\widetilde{z}$ is a cut-vertex in G^* for which $f(\widetilde{z}) = \pi_j(\widetilde{z})$, so g is a homomorphism from G^* to H.

Conversely, if we have $g : G^* \to H$, recall that from Observation 2.5 we know that g maps G^* to a connected component H_j, for some $j \in [m]$ (as we assume G^* is connected). But G^* contains an induced copy $\widetilde{H}_i^k$ of H_i^k, so also the induced copy of H_i, say $\widetilde{H}_i$ (recall Observation 2.9 (c)). So $g|_{V(\widetilde{H}_i)}$ is in fact a homomorphism from H_i to H_j. Recall from Observation 2.4 that since $H_1 + \ldots + H_m$ is a core, its connected components are pairwise incomparable cores – so j must be equal to i. It means that $g|_{V(G')}$. is a homomorphism from G' to H_i, so we can conclude that $G \to H_i$.

Observe that the number of vertices of G^* is equal to $n + |H_i^k| - 1 \leq n|H_i^k|$. Now let $\big(\mathcal{T}, \{X_a\}_{a \in V(\mathcal{T})}\big)$ be a tree decomposition of G of width t, and let b be a node of $\mathcal{T}$, such that $u \in X_b$. Define $X_{b'} := X_b \cup V(\widetilde{H}_i^k)$ and let $V(\mathcal{T}^*) = V(\mathcal{T}) \cup \{b'\}$ and $E(\mathcal{T}^*) = E(\mathcal{T}) \cup \{bb'\}$. Clearly, $\big(\mathcal{T}^*, \{X_a\}_{a \in V(\mathcal{T}^*)}\big)$ is a tree decomposition of G^*. This means that $\mathrm{tw}(G^*) \leq t + |H_i^k|$. The graph H_i is fixed, so the number of vertices of H_i^k is a constant. By our assumption, we can decide if $G^* \to H$ in time $\alpha^{\mathrm{tw}(G^*)} \cdot |G^*|^d \cdot c$, so we are able to decide if $G \to H_i$ in time

$$\alpha^{\mathrm{tw}(G^*)} \cdot |G^*|^d \cdot c \leq \alpha^t \cdot \alpha^{|H_i^k|} \cdot n^d \cdot |H_i^k|^d \cdot c = \alpha^t \cdot n^d \cdot c',$$

where $c' := c \cdot \alpha^{|H_i^k|} \cdot |H_i^k|^d$. □

Theorem 3.3 implies that for our purpose it is sufficient to consider connected cores.

4. Lower bounds

In this chapter we investigate the lower bounds for the complexity of $\textsc{Hom}(H)$. We split the analysis to two cases. In the first one we consider projective cores and prove that for them the bound in Theorem 3.1 is tight. In the second one we investigate the remaining cases and analyse for which target graphs the bound in Theorem 3.2 is tight.

4.1. Projective graphs

In this section we are going to prove the Theorem 1.3.

Theorem 1.3. *Let H be a fixed non-trivial projective core on k vertices and let n and t be, respectively, the number of vertices and the treewidth of an instance graph G.*
(a) Assuming a tree decomposition of G of width t is given, the $\textsc{Hom}(H)$ problem can be solved in time $k^t \cdot n^d \cdot c$, where $c, d > 0$ are constants.
(b) There is no algorithm to solve $\textsc{Hom}(H)$ in time $(k-\epsilon)^t \cdot n^d \cdot c$ for any $\epsilon > 0$, and any constants $c, d > 0$, unless the SETH fails.

Observe that the part (a) of Theorem 1.3 follows from Theorem 3.1, so we need to show the hardness counterpart, i.e., the statement (b). The following construction of the edge gadget is a crucial tool in our proof.

Lemma 4.1. *For every non-trivial projective core H, there exists a graph F with two special vertices u^* and v^*, satisfying the following:*
(a) for every $x, y \in V(H)$ such that $x \neq y$, there exists a homomorphism $f : F \to H$ such that $f(u^*) = x$ and $f(v^*) = y$,
(b) for every $f : F \to H$ it holds that $f(u^*) \neq f(v^*)$.
We call such F an edge gadget *for H.*

Proof. Let $V(H) = \{z_1, \ldots, z_k\}$. For $i \in [k]$ denote by z_i^{k-1} the $(k-1)$-tuple $(z_i, \ldots, z_i)$ and by $\overline{z_i}$ the $(k-1)$-tuple $(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots z_k)$. We claim that $F := H^{(k-1)k}$ and vertices

$$u^* := (z_1^{k-1}, \ldots, z_k^{k-1}) \quad \text{and} \quad v^* := (\overline{z_1}, \ldots, \overline{z_k})$$

satisfy the statement of the lemma. Note that the vertices of F are $k(k-1)$-tuples.

To see that (a) holds, observe that if x and y are distinct vertices of H, then there always exists $i \in [k(k-1)]$ such that $\pi_i(u^*) = x$ and $\pi_i(v^*) = y$. This means that π_i is a homomorphism from $H^{k(k-1)}$ to H satisfying $\pi_i(u^*) = x$ and $\pi_i(v^*) = y$.

To prove (b), recall that since H is projective, by Observation 2.10 the homomorphism f is a composition of some automorphism g of H and $\pi_i : H^{(k-1)k} \to H$ for some $i \in [k(k-1)]$ Observe that u^* and v^* were defined in a way such that $\pi_j(u^*) \neq \pi_j(v^*)$ for every $j \in [k(k-1)]$. As g is an automorphism, it is injective, which gives us $f(u^*) = g(\pi_i(u^*)) \neq g(\pi_i(v^*)) = f(v^*)$. □

Now we are ready to prove Theorem 1.3 *(b)*.

Proof of Theorem 1.3 (b). Note that H is non-trivial, so by Observation 2.2 we know that $k \geq 3$. We reduce from k-COLORING. Let G be an instance of k-COLORING with n vertices and treewidth t. We construct the instance G^* of Hom(H) as follows. First, for every $v \in V(G)$ we introduce to G^* a vertex v'. Denote the set of these vertices by V'. Now for every edge xy of $E(G)$ we introduce to G^* a copy of the edge gadget F from Lemma 4.1, denoted by F_{xy}. Then, for every edge xy of G we identify vertices u^* and v^* of F_{xy}, with x' and y' from V', respectively. This completes the construction of G^*. Observe that the number of vertices of G^* is less than $n + |F| \cdot \binom{n}{2} \leq |F| \cdot n^2$.

We claim that G is k-colorable if and only if $G^* \to H$. Indeed, let φ be a k-coloring of G. For simplicity of notation, we label the colors used by φ in the same way as the vertices of H, i.e., $z_1, z_2, \dots, z_k$. Define $g : V' \to V(H)$ by $g(v') := \varphi(v')$. Now consider the gadget F_{xy} for an edge $xy \in E(G)$. As φ is a proper coloring, for every pair of vertices x' and y' of V' which correspond to an edge xy in G it holds that $g(x') \neq g(y')$. So because of Lemma 4.1 (a), we can find a homomorphism $f_{xy} : F_{xy} \to H$ such that $f_{xy}(x') = g(x')$ and $f_{xy}(y') = g(y')$. Repeating this for every edge gadget, we can extend g to a homomorphism from G^* to H. Observe that there are no edges between two distinct edge gadgets, so because each f_{xy} is an H-coloring of edge gadget, g must be an H-coloring of G^*.

Conversely, from Lemma 4.1 *(b)*, we know that if g is an H-coloring of G^*, then for any edge xy of G we have that $g(x) \neq g(y)$. So any homomorphim from G^* to H induces a k-coloring of G.

Now let $(\mathcal{T}, \{X_a\}_{a \in V(\mathcal{T})})$ be a tree decomposition of G of width t. We extend it to a tree decomposition of G^*. For each edge xy of G there exists a bag X_b such that $x, y \in X_b$. Define $X_{b'} := X_b \cup V(F_{xy})$ and construct a tree $\mathcal{T}^*$ by adding for every edge xy of G a node b' to $V(\mathcal{T})$ and an edge bb' to $E(\mathcal{T})$. Observe that $(\mathcal{T}^*, \{X_a\}_{a \in V(\mathcal{T}^*)}\}$ is a tree decomposition of G^*. This means that $\text{tw}(G^*) \leq t + |F|$. Since the number of vertices of F depends only on $|H|$, it is a constant number, because H is fixed. So if we can decide if $G^* \to H$ in time $(k-\epsilon)^{\text{tw}(G^*)} \cdot |G^*|^d \cdot c$, then we are be able to decide if G is k-colorable in time

$$(k-\epsilon)^{\text{tw}(G^*)} \cdot |G^*|^d \cdot c \leq (k-\epsilon)^t \cdot (k-\epsilon)^{|F|} \cdot n^{2d} \cdot |F|^d \cdot c = (k-\epsilon)^t \cdot n^{d'} \cdot c'$$

for constants $c' = c \cdot (k-\epsilon)^{|F|} \cdot |F|^d$ and $d' = 2d$. This, by Theorem 1.1 is a contradiction with the SETH. □

4.2. Non-projective graphs

In this section we consider graphs not analyzed before, i.e., non-trivial, connected non-projective cores. First, we argue that if H is a non-projective core on k vertices, then the approach based on Lemma 4.1 does not work. We can show that it is impossible to construct an edge gadget with properties from Lemma 4.1 for a non-projective core H. Our argument uses the notion of *constructible* set, see Larose and Tardif [24]. A set $C \subseteq V(H)$ is said to be *constructible* if there exists a graph K, and vertices $x_0, \ldots, x_\ell \in V(K)$ and $y_1, \ldots, y_\ell \in V(H)$ such that

$$\{f(x_0) \in V(H) \mid f : K \to H \text{ is any homomorphism such that } f(x_i) = y_i \text{ for every } i \in [\ell]\} = C.$$

Less formally, C is constructible if we can find a graph K with a special vertex x_0 and some partial H-coloring c of K, in which $c(x_i) = y_i$ for every $i \in [\ell]$, such that C is the set of colors which can appear on x_0 when we extend c to some H-coloring of K.

The tuple

$$(K, x_0, x_1, \ldots, x_\ell, y_1, y_2, \ldots, y_\ell)$$

is called a *construction* of C.

It appears that the notion of constructible sets is closely related to projectivity.

Theorem 4.2 (Larose, Tardif [24]). *A graph H on at least three vertices is projective if and only if every subset of its vertices is constructible.*

We show that Lemma 4.1 cannot work for non-projective cores.

Proposition 4.3. *Assume that we can construct an edge gadget F with properties listed in Lemma 4.1 for a core H. Then every subset of its vertices is constructible.*

Proof. Fix a $C \subseteq V(H)$, let $\ell := |C|$ and let $k := |H|$. Suppose there exist an edge gadget F with properties listed in Lemma 4.1. Take $k-\ell$ copies of F, say $F_1, \ldots, F_{k-\ell}$ and denote the vertices u^* and v^* of the i-th copy F_i respectively by u_i^* and v_i^*. Identify the vertices u_i^* of all these copies, denote the obtained vertex by u^*, and the obtained graph by K. Now set $x_i = v_i^*$ for each $i \in [k-\ell]$, set $x_0 = u^*$, and set $\{y_1, \ldots, y_{k-\ell}\}$ to be the complement of C in $V(H)$.

It is easy to verify that this is a construction of the set C. Indeed, suppose that $x \in C$, so $x \notin \{y_1, \ldots, y_{k-\ell}\}$. Then, by Lemma 4.1 (a), for each copy F_i there exists a homomorphism $f_i : F_i \to H$ such that $f_1(v_i^*) = f_1(x_i) = y_i$ and $f_1(u^*) = f_1(x_0) = x$, because $y_i \neq x$ for every $i \in [k-\ell]$. Combining these homomorphisms, we get a homomorphism $f : K \to H$. On the other hand, suppose that $x \notin C$, so $x = y_i$ for some $i \in [k-\ell]$. But then from property (b) in Lemma 4.1 we know that for every homomorphism $f : F_i \to H$ it holds that $x = y_i = f(v_i^*) \neq f(u^*) = f(x_0)$. So x_0 cannot be mapped to x by any homomorphism from K to H. □

Observe that if a graph H is projective, then it must be indecomposable. Indeed, assume H has a non-trivial factorization $H_1 \times H_2$ for some H_1, H_2. Consider a homomorphism $f : (H_1 \times H_2)^2 \to H_1 \times H_2$, defined as $f((x,y),(x',y')) = (x,y')$. Note that it is idempotent, but not a projection, so H is not projective. An example of a non-projective graph core is $H = K_3 \times G_G$, where the G_G is the *Grötzsch graph* (see Figure 2.1).

On the other hand, it is natural to ask when indecomposability implies projectivity. Larose and Tardif in [24] leave an open question about non-projective graphs, in particular, cores: if a connected non-bipartite core is indecomposable, does it imply it is projective? Up to our knowledge, there is still no known example of a indecomposable, connected core, which would be non-projective.

Conjecture 1. *Let H be a connected non-trivial core. Then H is projective if and only if it is indecomposable.*

Because of these reasons, in this section we consider all remaining known cases of cores. From now we will assume that H is a decomposable, non-trivial connected core, so, by Theorem 2.7, it has a unique prime factorization $H_1 \times \ldots \times H_m$ for some $m \geq 2$.

Below we summarize some properties of a non-trivial factorization of a core H.

Observation 4.4. *Let H be a connected, non-trivial core with some non-trivial factorization $H_1 \times H_2$. Then both H_1 and H_2:*
(a) are cores,
(b) are non-trivial,
(c) are connected,
(d) are incomparable with each other.
In particular, because $\times$ is commutative and associative, if $H_1 \times \ldots \times H_m$ is a prime factorization of H, then (a)-(d) hold for every H_i.

Proof. We prove (a)-(c) only for H_1, but by symmetry it works for H_2 as well.

(a) Suppose that H_1 is not a core and let H_1^* be the core of H_1, so a proper induced subgraph of H_1. Let $f : H_1 \to H_1^*$. Define $H^* = H_1^* \times H_2$. Note that because H_1^* is a proper induced subgraph of H_1, then H^* is a proper induced subgraph of H. Consider a function $f' : V(H) \to V(H^*)$ defined by $f'(x_1, x_2) := (f(x_1), x_2)$. Observe that because f is a homomorphism, so is f'. This is a contradiction, because H is a core.

(b) We excluded the case when H_1 is isomorphic to K_1^* in the assumption. If H_1 is isomorphic to K_1, then by Observation 2.9 (a) its core is isomorphic to K_1. Also, if H_1 is isomorphic to K_2, then by Observation 2.9 (b) its core is isomorphic to K_2. But we assumed H is not non-trivial, a contradiction.

(c) This follows from Theorem 2.8, but the direct proof is not complicated: let u and v be some vertices in H_1. Fix $w \in V(H_2)$. The graph H is connected, so there exists a path $((u,w),(x_1,y_1),\ldots,(x_\ell,y_\ell),(v,w))$ in H. Clearly, the sequence $(u,x_1,\ldots,x_\ell,v)$ must induce a walk in H_1.

(d) To see that $H_1 \not\to H_2$, assume otherwise and let $g : H_1 \to H_2$. Observe that the set of vertices $\{(x_1, g(x_1)) \in V(H) \mid x_1 \in V(H_1)\}$ forms an induced subgraph of H, isomorphic to H_1. Since we always have $H \to H_1$ (for example, π_1 is always a homomorphism from H to H_1), it means that H has a homomorphism into its subgraph. As H_2 is not isomorphic to K_1^*, then H is not isomorphic to H_1, so H_1 is a proper subraph of H, a contradiction. Analogously we can prove that $H_2 \not\to H_1$. □

Again, we will consider the complexity of the Hom(H) problem. Recall from Theorem 3.2 that if H admits a prime factorization $H_1 \times \ldots \times H_m$, then Hom($H$) can be solved in time $\max_{j\in[m]} |H_j|^t \cdot n^d \cdot c$, where n and t are, respectively, the number of vertices and the treewidth of an instance graph, and c, d are positive constants. It would be interesting to know if it is tight for connected, non-trivial decomposable cores.

Below we investigate a restricted case of this problem. We say that a graph H_i is *truly projective* if it has at least three vertices and for every $s \geq 2$ and every connected core W incomparable with H_i, it holds that the only homomorphisms $f : H_i^s \times W \to H_i$ which satisfy $f(x,x,\ldots,x,y) = x$ for any $x \in V(H_i), y \in V(W)$, are projections. Observe that such defined graphs are projective.

Observation 4.5. *If a graph H is truly projective, then it is projective.*

Proof. Because of Theorem 2.11, it is enough to prove that if $g : H^2 \to H$ is an idempotent homomorphism, then it is a projection. Let W be a core incomparable with H. Define a homomorphism $f : H^2 \times W \to H$ as $f(x_1,x_2,w) \coloneqq g(x_1,x_2)$. But H is truly projective, which means that f is a projection, and so is g. □

Graphs with very similar but more restrictive property were studied by Larose in [22] in connection with problems raised by Greenwell and Lovász in [14]. In chapter 5 we use results presented there to obtain some interesting corollaries. But here we prove the following.

Theorem 4.6. *Let H be a fixed non-trivial connected core, with prime factorization $H_1 \times \ldots \times H_m$. Let n and t be, respectively, the number of vertices and the treewidth of an instance graph G. Assume there exists $i \in [m]$ such that H_i is truly projective. Then there is no algorithm to solve* Hom*(H) in time $(|H_i| - \epsilon)^t \cdot n^d \cdot c$ for any constants $c, d > 0$, any $\epsilon > 0$, unless the SETH fails.*

To simplify the notation, for any given homomorphism $f : G \to H_1 \times \ldots \times H_m$, we define $f_i \equiv \pi_i \circ f$. Then for each vertex v of G it holds that

$$f(x) = (f_1(x), \ldots, f_m(x)),$$

where f_i is a homomorphism from G to H_i.

To prove Theorem 4.6 we need another edge gadget. It is similar to the one in the previous chapter, but its properties are slightly modified. The following theorem will be useful to prove the correctness of our construction. To avoid introducing new definitions, we state the theorem in a sightly weaker form, using the terminology used in this paper, but see also Theorem 8.18 in [16].

Theorem 4.7 (Dörfler, [9]). *Let φ be an automorphism of a connected, non-bipartite, ramified graph H, with the prime factorization $H_1 \times \ldots \times H_m$. Then for each $i \in [m]$ there exists an automorphism $\varphi^{(i)}$ of H_i such that $\varphi_i(t_1, \ldots, t_m) \equiv \varphi^{(i)}(t_i)$.*

In particular, it implies the following.

Corollary 4.8. *Let μ be an automorphism of a connected, non-trivial core H, with some non-trivial factorization $H_1 \times R$, such that H_1 is indecomposable. Then there exist automorphisms $\mu^{(1)} : H_1 \to H_1$ and $\mu^{(2)} : R \to R$ such that $\mu(t, t') \equiv (\mu^{(1)}(t), \mu^{(2)}(t'))$.*

Proof. As R must be a non-trivial core (see Observation 4.4), it admits the unique prime factorization, say $H_2 \times \ldots \times H_m$, so $H_1 \times H_2 \times \ldots \times H_m$ is the unique prime factorization of H. From Theorem 4.7 we know that for each $i \in [m]$ there exists an automorphism $\varphi^{(i)}$ of H_i such that $\mu(t_1, \ldots, t_m) \equiv (\varphi^{(1)}(t_1), \ldots, \varphi^{(m)}(t_m))$. We define $\mu^{(1)}(t) := \varphi^{(1)}(t)$ for every vertex $t \in V(H_1)$. Clearly, $\mu^{(1)}$ is an automorphism of H_1. For every vertex $t' = (t_2, \ldots, t_m)$ of R such that $t_i \in H_i$ for $i \in \{2, \ldots, m\}$, define $\mu^{(2)}(t') := (\varphi^{(2)}(t_2), \ldots, \varphi^{(m)}(t_m))$. Observe that $(t_2, \ldots, t_m)(t'_2, \ldots, t'_m)$ is an edge of R if and only if $t_i t'_i$ is an edge of H_i for every $i \in \{2, \ldots, m\}$. This happens if and only if $\varphi^{(i)}(t_i)\varphi^{(i)}(t'_i)$ is an edge of H_i for every $i \in \{2, \ldots, m\}$, because each $\varphi^{(i)}$ is an automorphism. Clearly, $\varphi^{(i)}(t_i)\varphi^{(i)}(t'_i)$ is an edge of H_i for every $i \in \{2, \ldots, m\}$ if and only if

$$(\varphi^{(2)}(t_2), \ldots, \varphi^{(m)}(t_m))(\varphi^{(2)}(t'_2), \ldots, \varphi^{(m)}(t'^{(m)})) \text{ is an edge of } R,$$

so $\mu^{(2)}$ is indeed an automorphism of R. □

Below we construct an edge gadget with properties which are needed to perform a hardness reduction. The gadget is similar to the one constructed in Lemma 4.1, but more technically complicated.

Lemma 4.9. *Let $H = H_1 \times R$ be a connected, non-trivial core, such that H_1 is truly projective and $R \not\simeq K_1^*$. Let w be a fixed vertex of R. Then there exists a graph F and vertices u^*, v^* of F, satisfying the following conditions:*

(a) for every $xy \in E(H_1)$ there exists $f : F \to H$ such that $f(u^*) = (x, w)$ and $f(v^*) = (y, w)$,

(b) for any $f : F \to H$ it holds that $f_1(u^*)f_1(v^*) \in E(H_1)$.

Proof. Let $E(H_1) = \{e_1, \dots, e_s\}$ and let $e_i = u_i v_i$ for every $i \in [s]$ (clearly, one vertex can appear many times as some u_i or v_j). Consider the vertices

$$u := (u_1, \dots, u_s, v_1, \dots, v_s)$$
$$v := (v_1, \dots, v_s, u_1, \dots, u_s)$$

of H_1^{2s}. Let $F := H_1^{2s} \times R$, and let $u^* := (u, w)$ and $v^* := (v, w)$. We will treat vertices u and v as $2s$-tuples, and vertices u^* and v^* as $(2s+1)$-tuples.

Observe that, if $xy \in E(H_1)$, then there exists $i \in [2s]$ such that $x = \pi_i(u^*)$ and $y = \pi_i(v^*)$, it follows from the definition of u^* and v^*. Define a function $f : V(F) \to V(H)$ as $f(x_1, \dots, x_{2s}, w) := (\pi_i(x_1, \dots, x_{2s}), w)$. Observe that this is a homomorphism, for which $f(u^*) = f(u, w) = (x, w)$ and $f(v^*) = f(v, w) = (y, w)$, which is exactly the condition (a) in the statement of Lemma 4.9.

We prove (b) in two steps. First, we observe the following.

Claim 1. *Let $\varphi : F \to H$. If for every $z \in V(H_1)$ and $r \in V(R)$ it holds that $\varphi_1(z, \dots, z, r) = z$ then $\varphi_1(u^*)\varphi_1(v^*) \in E(H_1)$.*

Proof of Claim. Recall that R is a connected core incomparable with H_1, and H_1 is truly projective. It means that if $\varphi_1 : H_1^{2s} \times R \to H_1$ satisfies the assumption of the claim, then it is equal to π_i for some $i \in [2s]$. Directly from the definition of u^* and v^* we have that $\pi_i(u^*)\pi_i(v^*) \in E(H_1)$. ■

Observe that F contains a subgraph isomorphic to H, say $\widetilde{H}$, induced by the set $V(\widetilde{H}) = \{(z, \dots, z, r) \in F \mid z \in V(H_1), r \in V(R)\}$. Indeed, there is an isomorphism $\sigma : \widetilde{H} \to H$ defined as $\sigma(z, \dots, z, r) := (z, r)$.

Second, assume that f is a homomorphism from F to H and observe that $f\big|_{V(\widetilde{H})}$ is an isomorphism from $\widetilde{H}$ to H, because H is a core. If $f\big|_{V(\widetilde{H})} \equiv \sigma$ then for every $z \in V(H)$ and $r \in V(R)$ it holds that $f_1(z, \dots, z, r) = \sigma_1(z, \dots, z, r) = z$, so, by the claim above, we are done. If not, observe that there exists the inverse isomorphism $g : H \to \widetilde{H}$ such that $g \circ f\big|_{V(\widetilde{H})} = id_{\widetilde{H}}$. Let $\mu := \sigma \circ g$. Observe that μ is an endomorphism of $H_1 \times R$, so an automorphism, since $H_1 \times R$ is a core. Also note that $(\mu \circ f) : F \to H_1 \times R$ is a homomorphism such that for every $(z, \dots, z, r) \in V(\widetilde{H})$ it holds that

$$(\mu \circ f)(z, \dots, z, r) = (\sigma \circ g \circ f)(z, \dots, z, r) = (\sigma \circ id_{\widetilde{H}})(z, \dots, z, r) = \sigma(z, \dots, z, r) = (z, r),$$

which means that $(\mu \circ f)_1(z, \dots, z, r) = z$. This means that $\mu \circ f$ satisfies the assumption of the claim, so

$$(\mu \circ f)_1(u^*)(\mu \circ f)_1(v^*) \in E(H_1). \tag{4.1}$$

Clearly, for every vertex $\tilde{z}$ of F it holds that

$$(\mu \circ f)(\tilde{z}) = \mu\Big(f_1(\tilde{z}), f_2(\tilde{z})\Big) = \left(\mu_1\Big(f_1(\tilde{z}), f_2(\tilde{z})\Big), \mu_2\Big(f_1(\bar{z}), f_2(\tilde{z})\Big)\right). \tag{4.2}$$

Note that Corollary 4.8 implies that there exists automorphisms $\mu^{(1)}$ and $\mu^{(2)}$ of, respectively, H_1 and R, such that for every $\tilde{z} \in V(F)$ it holds that

$$\begin{aligned} \mu_1\left(f_1(\tilde{z}), f_2(\tilde{z})\right) &= \mu^{(1)}(f_1(\tilde{z})) \\ \mu_2\left(f_1(\tilde{z}), f_2(\tilde{z})\right) &= \mu^{(2)}(f_2(\tilde{z})), \end{aligned} \tag{4.3}$$

So (4.2) and (4.3) show that in particular $(\mu \circ f)_1 = \mu^{(1)} \circ f_1$. Combining this with (4.1) we get that

$$\left(\mu^{(1)} \circ f_1\right)(u^*)\left(\mu^{(1)} \circ f_1\right)(v^*) \in E(H_1). \tag{4.4}$$

Since $\mu^{(1)}$ is an automorphism of H_1, there exists the inverse automorphism $\left(\mu^{(1)}\right)^{-1}$ of H_1 for which $\left(\mu^{(1)}\right)^{-1} \circ \mu^{(1)}$ is the identity of H_1. Because $\left(\mu^{(1)}\right)^{-1}$ is an automorphism, (4.4) implies that

$$\left(\left(\mu^{(1)}\right)^{-1} \circ \mu^{(1)} \circ f_1\right)(u^*)\left(\left(\mu^{(1)}\right)^{-1} \circ \mu^{(1)} \circ f_1\right)(v^*) \in E(H_1).$$

Since $\left(\left(\mu^{(1)}\right)^{-1} \circ \mu^{(1)}\right)$ is the identity, we conclude that $f_1(u^*)f_1(v^*) \in E(H_1)$, which completes the proof. □

Now we can proceed to the proof of the Theorem 4.6.

Proof of Theorem 4.6. Since $\times$ is commutative, then without loss of generality we can assume that H_1 is truly projective. Define $R := H_2 \times \ldots \times H_m$, so $H = H_1 \times R$. Since H_1 is truly projective, Observation 4.5 implies that it is projective, so it is connected and by Theorem 1.3 there is no algorithm for Hom(H_1) which works in time $(|H_1| - \epsilon)^t \cdot n^d \cdot c$, for instance graphs on n vertices and treewidth t, for any constants $c, d, \epsilon > 0$, unless the SETH fails. We will show that an existence of an algorithm for Hom(H) working in time $(|H_1| - \epsilon)^t \cdot n^{d'} \cdot c'$ for some constants $c', d' > 0$ would contradict the SETH.

For a given instance G of Hom(H_1) consider an instance G^* of Hom(H), which is constructed as follows. Let w be a fixed vertex of R and let F be a graph obtained for H and w from Lemma 4.9. To construct G^*, first, for every vertex v of G we introduce a vertex v' of G^*. Denote by V' the set of these vertices in G^*. Then we add a copy F_{xy} of F for every pair $x', y' \in V'$ which corresponds to an edge xy in G, and identify vertices x' and y' respectively with u^* and v^* of F_{xy}. Note that G^* has at most $|F| \cdot n^2$ vertices.

We claim that there exists $f : G \to H_1$ if and only if there exists $f^* : G^* \to H$. Clearly, for $f^* : G^* \to H$ by Lemma 4.9 (b) we know that the function $f_1^*|_{V'}$ corresponds to a homomorphism from G to H_1.

For $f : G \to H_1$, define $f^*(v) := (f(v), w)$ for every $v \in V'$. Consider the edge gadget F_{xy} for an edge xy of G. From Lemma 4.9 (a) we know that we can extend f^* to every vertex of F_{xy}. So $G^* \to H$ if and only if $G \to H_1$.

Finally, observe that each edge gadget has a constant size (depending only on H). Again we can construct a tree decomposition of G^* from a given decomposition $\mathcal{T}$ of G of width t. For each edge xy of G there exists a bag X_b such that $x, y \in X_b$. We define $X_{b'} = X_b \cup V(F_{xy})$ and construct a tree $\mathcal{T}^*$ by adding for every edge $y \in G$ a node b' to $V(\mathcal{T})$ and an edge $\{bb'\}$ to $E(\mathcal{T})$. Clearly, $(\mathcal{T}^*, \{X_a\}_{a \in V(\mathcal{T}^*)}\}$ is a tree decomposition of G^*, so $\mathrm{tw}(G^*) \le t + |F|$. It means that if we could decide if $G^* \to H$ in time $(|H_1| - \epsilon)^{\mathrm{tw}(G^*)} \cdot |G^*|^d \cdot c$, then we would be able to decide if $G \to H_1$ in time

$$\begin{aligned}(|H_1| - \epsilon)^{\mathrm{tw}(G^*)} \cdot |G^*|^d \cdot c &\le (|H_1| - \epsilon)^t \cdot (|H_1| - \epsilon)^{|F|} \cdot n^{2d} \cdot |F|^d \cdot c \\ &\le (|H_1| - \epsilon)^t \cdot n^{d'} \cdot c'\end{aligned}$$

for $c' := c \cdot (|H_1| - \epsilon)^{|F|} \cdot |F|^d$ and $d' = 2d$. By Theorem 1.3(b) this is a contradiction with the SETH. □

Using Theorem 3.2 we obtain a tight bound for the complexity of Hom(H), for cores H, whose largest factor is truly projective.

Corollary 4.10. *Let H be a connected core with prime factorization $H_1 \times \ldots \times H_m$ and let H_i be the factor with the largest number of the vertices. Assume that H_i is truly projective. Let n and t be, respectively, the number of vertices and the treewidth of an instance graph G.*

(a) If a tree decomposition of G of width t is given, the Hom*(H) problem can be solved in time $|H_i|^t \cdot n^d \cdot c$, for some constants $c, d > 0$.*

(b) There is no algorithm to solve Hom*(H) in time $(|H_i| - \epsilon)^t \cdot n^d \cdot c$ for any $\epsilon > 0$ and any constants $c, d > 0$, unless the SETH fails.* □

5. Summary

To summarize our results, recall that we know that a truly projective core must be projective and a projective core must be indecomposable. So if we could prove that all connected, non-trivial indecomposable cores are truly projective, we would obtain a tight complexity bound in Theorem 4.6 and Corollary 4.10 for all graphs H. Let us discuss the possibility of obtaining such a result.

As mentioned before, Larose in [22] considered the concept of *strongly projective* graphs. We say that a graph H_1 on at least three vertices is *strongly projective*, if for every connected graph W on at least two vertices and every $s \geq 2$, the only homomorphisms $f : H_1^s \times W \to H_1$ satisfying $f(x, \ldots, x, y) = x$ for all $x \in V(H_1)$ and $y \in V(W)$, are projections. If we compare this with the definition of truly projective graphs, we see that for truly projective graphs H we restricted the homomorphisms from $H^s \times W$ to H only for connected cores W, that are incomparable with H. So the following is clear.

Observation 5.1. *Every strongly projective graph is truly projective.* □

In [22] Larose considers many examples of families of strongly projective graphs. Restricting the results presented there to cores, we can provide some examples of non-projective cores, for which the complexity bound in Theorem 3.2 is tight.

We say that graph is *square-free* if it does not contain a subgraph isomorphic to C_4 (not necessarily induced).

Theorem 5.2 (Larose, [22])**.** *If H is a square-free, connected, ramified, non-bipartite graph, then it is strongly projective.*

Example 1. Consider the graph G_B, shown of Figure 5.1, called the *Brinkmann graph*. Observe that $|G_B| = 21$. It is connected, non-bipartite, and its girth is equal to 5, which means it is square-free. Also, it is vertex-critical, so it must be a core. Thus by Theorem 5.2 we know that G_B is strongly projective. By exhaustive computer search we verified that $K_3 \times G_B$ is a core. Let us consider the complexity of $\textsc{Hom}(K_3 \times G_B)$ for input graphs with n vertices and treewidth t, given along with its optimal tree decomposition. The dynamic programming approach from Theorem 3.1 gives us the running time $63^t \cdot c \cdot n^d$, for some constants $c, d > 0$. But from Theorem 4.6 we can conclude that we can solve $\textsc{Hom}(K_3 \times G_B)$ in time $21^t \cdot n^d \cdot c$. Moreover, by Corollary 4.10, assuming the SETH, there is no algorithm to solve $\textsc{Hom}(K_3 \times G_B)$ in time $(21 - \epsilon)^t \cdot n^d \cdot c$.

Consider another family of strongly projective graphs. A graph is said to be *primitive* if there is no non-trivial partition of its vertices which is invariant under all automorphisms of this graph (see e.g., [31]).

Theorem 5.3 (Larose, [22]). *If H is a directly indecomposable primitive core, then it is strongly projective.*

From this theorem we can conclude that, for example, Kneser graphs, which are known to be projective cores (see [25],[15]) are strongly projective. However, Kneser graphs do not have to be square-free, so it is not implied by Theorem 5.2. In particular, complete graphs are strongly projective.

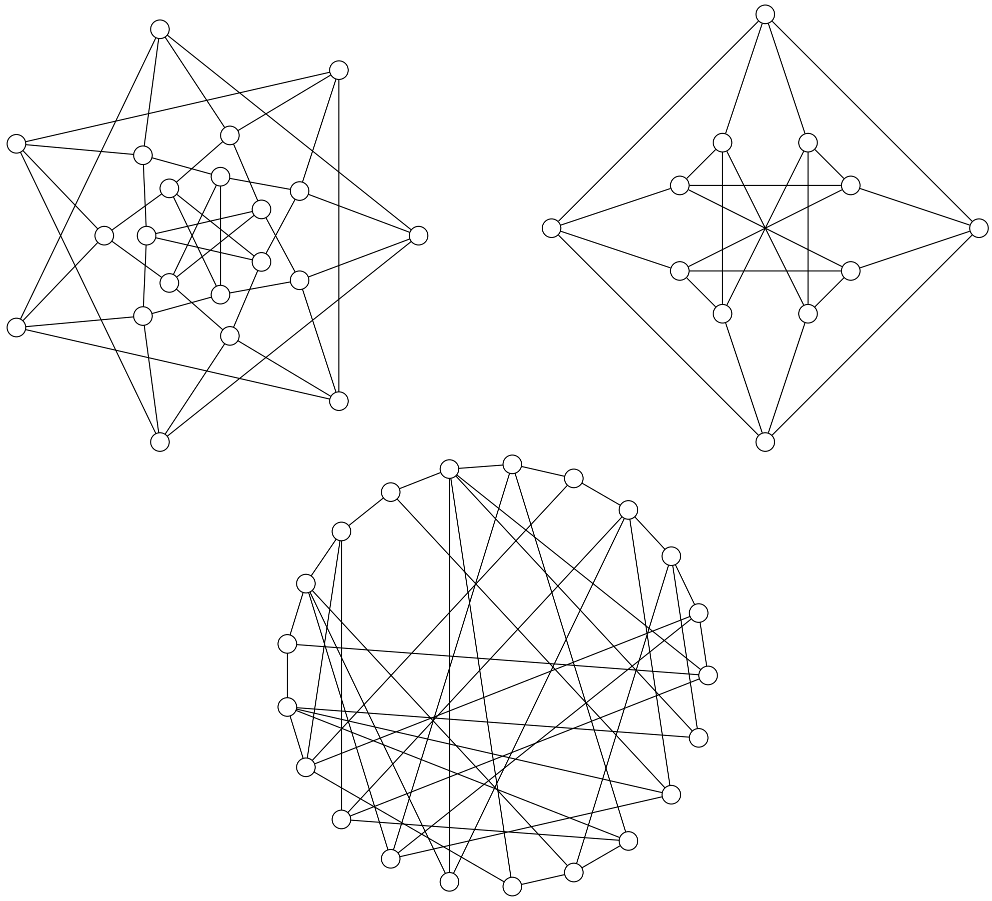

Figure 5.1: The Brinkmann graph G_B (left), the Chvátal graph G_C (right) and the graph G_M (down).

Larose in [22] and [23] studies also the properties of strongly projective graphs. In particular, his results imply that this property is decidable (which, again, does not follow directly from the definition). Also, he shows that all known families of projective graphs on at least three vertices contain only strongly projective graphs and poses a question if every projective graph on at least three vertices is in fact strongly projective. As we have shown in Observations 5.1 and 4.5, respectively, every strongly projective graph is truly projective and every truly projective graph is projective, so a positive answer to Larose's question would imply that these three classes are in fact equivalent. We recall this problem in a weaker form, which would be sufficient in our setting.

Conjecture 2. *Let H be a non-trivial core. Then it is truly projective if and only if it is projective.*

Clearly, if both Conjecture 1 and Conjecture 2 are true, there is another characterization of indecomposable connected cores.

Observation 5.4. *Assume that Conjecture 1 and Conjecture 2 hold. Let H be a connected non-trivial core. Then H is indecomposable if and only if it is truly projective.* □

We can summarize the main results of chapter 4 in the following theorem.

Theorem 5.5. *Assume that Conjectures 1 and 2 hold.*
Let H be a fixed non-trivial connected core. Let $H_1 \times \ldots \times H_m$ be a prime decomposition of H. Let $k := \max_{i\in[m]} |H_i|$.

(a) Assuming a tree decomposition of G of width t is given, the Hom*(H) problem can be solved in time $k^t \cdot n^d \cdot c$, for instance graphs on n vertices, where $c, d > 0$ are constants.*
(b) There is no algorithm solving Hom*(H) in time $(k-\epsilon)^t \cdot n^d \cdot c$ for any $\epsilon > 0$ and any constants $c, d > 0$, unless the SETH fails.* □

To conclude this analysis, we point out that in the light of the problem we consider, it is natural to ask when the product of graphs is a core. Observation 4.4 provides us some necessary conditions, it would be interesting to know if they are sufficient for a product of indecomposable graphs to be a core.

Conjecture 3. *Let H_1 and H_2 be indecomposable, incomparable cores. Then $H_1 \times H_2$ is a core.*

We confirmed this conjecture by exhaustive computer search for some small graphs. In particular, the conjecture is true for graphs $K_3 \times H$, where H is any 4-vertex-critical, triangle-free graph with at most 14 vertices [5], including the Grötzsch graph (see Figure 2.1) and the Brinkmann graph (see Figure 5.1 (left)). It is true also for the Chvátal graph (see Figure 5.1 (right)) and the graph G_M (see Figure 5.1 (down)), which are two of the 18 smallest graphs with chromatic number 4 and girth 5. In particular, G_B and G_M are strongly projective by Theorem 5.2.

Bibliography

[1] N. Alon, I. Dinur, E. Friedgut, and B. Sudakov. Graph products, Fourier analysis and spectral techniques. *Geometric & Functional Analysis GAFA*, 14(5):913–940, 2004.

[2] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete applied mathematics*, 23(1):11–24, 1989.

[3] A. Björklund, T. Husfeldt, and M. Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009.

[4] H. L. Bodlaender, P. Bonsma, and D. Lokshtanov. The fine details of fast dynamic programming over tree decompositions. In *International Symposium on Parameterized and Exact Computation*, pages 41–53. Springer, 2013.

[5] G. Brinkmann, K. Coolsaet, J. Goedgebeur, and H. Mélot. House of Graphs: A database of interesting graphs. *Discrete Applied Mathematics*, 161(1-2):311–314, 2013.

[6] V. Chvátal. The minimality of the Mycielski graph. In *Graphs and combinatorics*, pages 243–246. Springer, 1974.

[7] M. Cygan, F. V. Fomin, A. Golovnev, A. S. Kulikov, I. Mihajlin, J. Pachocki, and A. Socała. Tight lower bounds on graph embedding problems. *Journal of the ACM (JACM)*, 64(3):18, 2017.

[8] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*, volume 4. Springer, 2015.

[9] W. Dörfler. Primfaktorzerlegung und Automorphismen des Kardinalproduktes von Graphen. *Glasnik Matematički*, 9:15–27, 1974.

[10] L. Egri, D. Marx, and P. Rzążewski. Finding list homomorphisms from bounded-treewidth graphs to reflexive graphs: A complete complexity characterization. 2018.

[11] P. Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[12] T. Feder and P. Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B*, 72(2):236–250, 1998.

[13] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[14] D. Greenwell and L. Lovász. Applications of product colouring. *Acta Mathematica Hungarica*, 25(3-4):335–340, 1974.

[15] G. Hahn and C. Tardif. Graph homomorphisms: structure and symmetry. In *Graph symmetry*, pages 107–166. Springer, 1997.

[16] R. Hammack, W. Imrich, and S. Klavžar. *Handbook of product graphs*. CRC press, 2011.

[17] P. Hell and J. Nešetřil. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990.

[18] P. Hell and J. Nešetřil. The core of a graph. *Discrete Mathematics*, 109(1-3):117–126, 1992.

[19] P. Hell and J. Nešetřil. *Graphs and homomorphisms*. Oxford University Press, 2004.

[20] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[21] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[22] B. Larose. Families of strongly projective graphs. *Discussiones Mathematicae Graph Theory*, 22:271–292, 2002.

[23] B. Larose. Strongly projective graphs. *Canadian Journal of Mathematics*, 54(4):757–768, 2002.

[24] B. Larose and C. Tardif. Strongly rigid graphs and projectivity. *Multiple-Valued Logic*, 7:339–361, 2001.

[25] B. Larose and C. Tardif. Projectivity and independent sets in powers of graphs. *Journal of Graph Theory*, 40(3):162–171, 2002.

[26] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018.

[27] T. Łuczak and J. Nešetřil. Note on projective graphs. *Journal of Graph Theory*, 47(2):81–86, 2004.

[28] R. McKenzie. Cardinal multiplication of structures with a reflexive relation. *Fundamenta Mathematicae*, 70(1):59–101, 1971.

[29] D. J. Miller. The Categorical Product of Graphs. *Canadian Journal of Mathematics*, 20:1511–1521, 1968.

[30] J. Nešetřil and X. Zhu. On sparse graphs with given colorings and homomorphisms. *Journal of Combinatorial Theory, Series B*, 90(1):161–172, 2004.

[31] D. H. Smith. Primitive and imprimitive graphs. *The Quarterly Journal of Mathematics*, 22(4):551–557, 1971.

[32] P. M. Weichsel. The Kronecker product of graphs. *Proceedings of the American mathematical society*, 13(1):47–52, 1962.