Complexity of the List Homomorphism Problem in Hereditary Graph Classes

Karolina Okrasa
Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland
Institute of Informatics, University of Warsaw, Poland

Paweł Rzążewski
Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland
Institute of Informatics, University of Warsaw, Poland

Abstract

A homomorphism from a graph G to a graph H is an edge-preserving mapping from $V(G)$ to $V(H)$. For a fixed graph H, in the list homomorphism problem, denoted by LHom(H), we are given a graph G, whose every vertex v is equipped with a list $L(v) \subseteq V(H)$. We ask if there exists a homomorphism f from G to H, in which $f(v) \in L(v)$ for every $v \in V(G)$. Feder, Hell, and Huang [JGT 2003] proved that LHom(H) is polynomial time-solvable if H is a so-called bi-arc-graph, and NP-complete otherwise.

We are interested in the complexity of the LHom(H) problem in F-free graphs, i.e., graphs excluding a copy of some fixed graph F as an induced subgraph. It is known that if F is connected and is not a path nor a subdivided claw, then for every non-bi-arc graph the LHom(H) problem is NP-complete and cannot be solved in subexponential time, unless the ETH fails. We consider the remaining cases for connected graphs F.

If F is a path, we exhibit a full dichotomy. We define a class called predacious graphs and show that if H is not predacious, then for every fixed t the LHom(H) problem can be solved in quasi-polynomial time in P_t-free graphs. On the other hand, if H is predacious, then there exists t, such that the existence of a subexponential-time algorithm for LHom(H) in P_t-free graphs would violate the ETH.

If F is a subdivided claw, we show a full dichotomy in two important cases: for H being irreflexive (i.e., with no loops), and for H being reflexive (i.e., where every vertex has a loop). Unless the ETH fails, for irreflexive H the LHom(H) problem can be solved in subexponential time in graphs excluding a fixed subdivided claw if and only if H is non-predacious and triangle-free. On the other hand, if H is reflexive, then LHom(H) cannot be solved in subexponential time whenever H is not a bi-arc graph.

2012 ACM Subject Classification Mathematics of computing → Graph coloring; Theory of computation → Problems, reductions and completeness; Theory of computation → Graph algorithms analysis

Keywords and phrases list homomorphism, fine-grained complexity, hereditary graph classes

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.54

Related Version *Full Version*: https://arxiv.org/abs/2010.03393 [30]

Funding *Karolina Okrasa*: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme Grant Agreement no. 714704.
Paweł Rzążewski: Supported by Polish National Science Centre grant no. 2018/31/D/ST6/00062.

1 Introduction

Many natural graph-theoretic problems, including Independent Set, k-Coloring, Max Cut, Odd Cycle Transversal, etc., can be defined in a uniform way as the question of the existence of certain graph homomorphisms. For two graphs G and H, a function

$f : V(G) \to V(H)$ is a *homomorphism from G to H* if for every $uv \in E(G)$ it holds that $f(u)f(v) \in E(H)$. If f is a homomorphism from G to H we denote it by $f : G \to H$. As an important special case, we observe that homomorphisms to K_k are precisely k-colorings of G. This is why homomorphisms to H are often called *H-colorings*. We will refer to the graph H as the *target* and to the vertices of H as *colors*. For fixed H, by Hom(H) we denote the computational problem of deciding if an instance graph admits an H-coloring.

The complexity dichotomy for Hom(H) was shown by Hell and Nešetřil [21]: If H is bipartite or has a vertex with a loop, then the problem is polynomial-time-solvable, and otherwise it is NP-complete. The study of variants of graph homomorphisms has attracted a significant attention [2, 26, 7, 8, 16, 15]. Arguably, the most natural generalization of the problem is the *list homomorphism problem*. For fixed H, an instance of the LHom(H) problem is a pair (G, L), where G is a graph and L is a function that to every vertex $v \in V(G)$ assigns its *H-list* (or *list*) $L(v) \subseteq V(H)$. We ask if there exists a homomorphism $f : G \to H$, such that for every $v \in V(G)$ it holds that $f(v) \in L(v)$. We write $f : (G, L) \to H$ if f is a list homomorphism from G to H which respects the lists L, and we write $(G, L) \to H$ to indicate that some such f exists.

The complexity classification for LHom(H) was proven in three steps. First, Feder and Hell [11] considered reflexive target graphs H, i.e., where every vertex has a loop. In this case LHom(H) is polynomial-time solvable if H is an interval graph and NP-complete otherwise. Then, Feder et al. [12] showed the dichotomy in the case that H is irreflexive, i.e., has no loops. This problem appears to be polynomial-time solvable if H is bipartite and its complement is a circular-arc graph, and NP-complete otherwise. Finally, Feder et al. [13] defined a new class of graphs with possible loops, called *bi-arc graphs*, and showed that if H is a bi-arc graph, then LHom(H) can be solved in polynomial time, and otherwise the problem is NP-complete. Reflexive bi-arc graphs coincide with interval graphs, and irreflexive bi-arc graphs are precisely bipartite graphs whose complement is a circular-arc graph. Let us point out that all mentioned hardness reductions for LHom(H) also exclude the existence of a subexponential-time algorithm, unless the ETH fails.

An active line of research is to study the complexity of computational problems, when the instance is assumed to belong some specific graph class. We usually assume that the considered classes are *hereditary*, i.e., closed under vertex deletion. Each such a hereditary class can be characterized by a (possibly infinite) set of forbidden induced subgraphs. For a family $\mathcal{F}$ of graphs, a graph is *$\mathcal{F}$-free* if it does not contain any member of $\mathcal{F}$ as an induced subgraph. Most attention is put into considering classes with only one forbidden subgraph, i.e., for $\mathcal{F} = \{F\}$. In this case we write F-free, instead of $\{F\}$-free. We will always assume that F is connected.

Let us define two important families of graphs. For an integer $t \geqslant 1$, by P_t we denote the path with t vertices. For $a, b, c \geqslant 0$, by $S_{a,b,c}$ we denote the graph obtained by taking three disjoint paths P_{a+1}, P_{b+1}, and P_{c+1} and merging one of the endvertices of each path into one vertex. Note that if at least one of a, b, c is equal to 0, then $S_{a,b,c}$ is an induced path. The members of $\{S_{a,b,c} \mid a, b, c \geqslant 0\}$ are called *subdivided claws*.

Let us briefly discuss the complexity of k-Coloring in F-free graphs. First, we observe that if F is not a path, then for every fixed $k \geqslant 3$, the k-Coloring remains NP-complete in F-free graphs. Indeed, Emden-Weinert et al. [10] proved that the problem is hard for graphs with no cycles shorter than p, for any constant p. Setting $p = |V(F)| + 1$ yields the hardness for F-free graphs whenever F contains a cycle. On the other hand, k-Coloring is NP-complete in line graphs [23, 27], which are in particular $S_{1,1,1}$-free. This implies the hardness for F-free graphs if F is a tree with maximum degree at least 3. Combining these, we conclude that the only connected graphs F, for which we might hope for a polynomial-time algorithm for k-Coloring in F-free graphs, are paths.

The complexity of k-Coloring in P_t-free graphs has been an active area of research in the last two decades, see the survey by Golovach et al. [18]. The current state of art is as follows. We know that for each fixed k, the problem is polynomial-time-solvable in P_5-free graphs [22]. On the other hand, for every $k \geqslant 5$, the problem is NP-complete in P_6-free graphs [24]. The complexity of 4-Coloring in P_t-free graphs is also fully understood: it is polynomial-time solvable for $t \leqslant 6$ [34] and NP-complete for $t \geqslant 7$ [24]. Finally, we know that 3-Coloring admits a polynomial time algorithm in P_7-free graphs [1]. Interestingly, we know no proof of NP-hardness of 3-Coloring in P_t-free graphs, for any value of t. The problem is believed to be solvable in polynomial time for every t, and obtaining such an algorithm is one of the main open questions in the area.

Let us point out that all mentioned hardness proofs rule out the existence of subexponential-time algorithms, unless the ETH fails. Furthermore, all algorithmic results hold even for List k-Coloring, except for the case $(k, t) = (4, 6)$, which is NP-complete in the list setting [19].

Even though our current toolbox seems to be insufficient to solve 3-Coloring in P_t-free graphs in polynomial time for all t, we can still solve the problem significantly faster than for general graphs. Groenland et al. [20] showed an algorithm with running time $2^{\mathcal{O}(\sqrt{n \log n})}$, for all fixed t. Very recently, Pilipczuk et al. [33] observed that the breakthough algorithm for Independent Set in P_t-free graphs by Gartland and Lokshtanov [17], could be adapted to solve 3-Coloring in time $n^{\mathcal{O}(\log^3 n)}$. They also presented an arguably simpler algorithm with running time $n^{\mathcal{O}(\log^2 n)}$.

The complexity of the Hom(H) and LHom(H) problems in F-free graphs received a lot less attention [14, 25]. On the negative side, Piecyk and Rzążewski [32], showed that if F is connected and is not a subdivided claw, then for every non-bi-arc H, the LHom(H) problem remains NP-complete in F-free graphs and cannot be solved in subexponential time, assuming the ETH.

There are several results about the complexity of LHom(H) in P_t-free graphs. First, Chudnovsky et al. [3] showed that for $k \in \{5, 7, 9\} \cup [10; \infty)$, the LHom($C_k$) problem can be solved in polynomial time for P_9-free graphs. Very recently, Chudnovsky et al. [4] studied some further generalization of the homomorphism problem in subclasses of P_6-free graphs. Furthermore, the already mentioned $2^{\mathcal{O}(\sqrt{n \log n})}$-time algorithm by Groenland et al. [20] actually works for LHom(H) for a large family of graphs H: the requirement is that H does not contain two vertices with two common neighbors. Even more generally, the algorithm can solve a *weighted homomorphism problem*, where, in addition to lists, we allow vertex- and edge-weights. Later, Okrasa and Rzążewski [31] proved that the weighted homomorphism problem cannot be solved in P_t-free graphs in subexponential time, whenever the target graph has two vertices with two common neighbors. However, for some of the hardness reductions it was essential to exploit the existence of vertex- and edge-weights and thus they cannot be translated to the arguably more natural LHom(H) problem.

Our results. In this paper we investigate the fine-grained complexity of LHom(H) in F-free graphs, where F is a subdivided claw. Recall that these are the only connected forbidden graphs for which we can hope for the existence of subexponential-time algorithms.

First, we define the family of *predacious* graphs, and show that they precisely correspond to "hard" cases of LHom(H) in P_t-free graphs. More specifically, we prove the following theorem.

▶ **Theorem 1.** *Let H be a fixed graph.*

a) *If H is not predacious, then for every t, the* LHom*(H) problem can be solved in time $n^{\mathcal{O}(\log^2 n)}$ in n-vertex P_t-free graphs.*

b) *If H is predacious, then there exists t, such that the* LHom*(H) problem cannot be solved in time $2^{o(n)}$ in n-vertex P_t-free graphs, unless the ETH fails.*

The definition of predacious graphs is based on the decomposition theorem by Okrasa et al. [28] that is particularly useful for solving the LHom(H) problem. Using this theorem, each graph H can be decomposed into a family of induced subgraphs, called *factors*. Now, a graph H is predacious, if it has a factor that is simultaneously non-bi-arc and contains a *predator*: two vertices a_1, a_2 with two common neighbors b_1, b_2, such that a_1 and a_2 have incomparable neighborhoods and b_1 and b_2 have incomparable neighborhoods. Note that a predator is a refinement of the essential structure in the dichotomy for the weighted homomophism problem [20, 31].

The proof of Theorem 1 a) builds on the already mentioned decomposition of target graphs by Okrasa et al. [28] and on the recent quasi-polynomial-time algorithm for 3-Coloring P_t-free graphs [33]. The hardness counterpart is proven in two steps. First, we consider a special case that H is bipartite and "undecomposable" (the exact meaning of this is given in Section 2). Okrasa et al. [28] analyzed the structure of such graphs and showed that it is rich enough to build a number of useful gadgets. We use them as building blocks of gadgets required in our hardness reduction. Then, we lift this hardness result to general predacious graphs H, using the idea of *associated bipartite graphs* [13].

Next, we turn our attention to the case that F is an arbitrary subdivided claw. We obtain the dichotomy in two important special cases: that H is irreflexive, and that H is reflexive. Recall that these cases correspond to the first two steps of the complexity dichotomy for LHom(H) [11, 12].

As a warm-up, let us discuss the case that H is irreflexive and F is the simplest subdivided claw, i.e., the claw $S_{1,1,1}$. Recall that 3-Coloring is NP-complete in line graphs [23], which are in particular claw-free. Since the reduction yields an ETH lower bound, we obtain that if H contains a simple triangle, then LHom(H) cannot be solved in subexponential time in claw-free graphs.

So let us consider the case that H is triangle-free. We note that there is no homomorphism $K_3 \to H$, so if the instance graph contains a triangle, we can immediately report a no-instance. On the other hand, $\{S_{1,1,1}, K_3\}$-free graphs are just collections of disjoint paths and cycles, where the problem can be solved in polynomial time using dynamic programming. We generalize this simple classification to the case if F is an arbitrary subdivided claw as follows.

▶ **Theorem 2.** *Let H be a fixed irreflexive graph.*

a) *If H is non-predacious and triangle-free, then for every a, b, c, the* LHom*(H) problem can be solved in time $2^{\mathcal{O}(n^{8/9} \log n)}$ in n-vertex $S_{a,b,c}$-free graphs.*

b) *If H is predacious or contains a triangle, then there exist a, b, c, such that the* LHom*(H) problem cannot be solved in time $2^{o(n)}$ in n-vertex $S_{a,b,c}$-free graphs, unless the ETH fails.*

The algorithm from Theorem 2 a) is based on the existence of the so-called *extended strip decomposition* [6]. A similar approach was used by Chudnovsky et al. [5] to obtain a QPTAS and a subexponential-time algorithm for the Max Independent Set problem in $S_{a,b,c}$-free graphs. However, the decomposition itself is not structured enough to be useful for coloring problems, such as LHom(H). We proceed as follows. First, similarly as before, we restrict ourselves to instances that are $\{S_{a,b,c}, K_3\}$-free. We analyze the structure of such graphs G and show that they admit an extended strip decomposition with a very simple structure.

Very roughly speaking, we can find a "small" set $X \subseteq V(G)$, such that for each connected component C of $G - X$, the vertices of C can be partitioned into "small" sets called *atoms*, that can be arranged in a path-like or cycle-like manner. We exhaustively guess the coloring of X (which is fine, as X is small). For each atom we solve the problem recursively. Finally, we use the path-like or cycle-like arrangement of atoms to combine partial results using dynamic programming, similarly as we did for $\{S_{1,1,1}, K_3\}$-free graphs.

Let us point out that the assumption that H is irreflexive and triangle-free is only used to ensure that the instance is triangle-free. For such instances we can solve $\textsc{LHom}(H)$ in subexponential time for *every* non-predacious graph H.

The hardness counterpart of Theorem 2 is simple. If H is predacious, then we are done by Theorem 1 b), as every P_t-free graph is also $S_{t,t,t}$-free. On the other hand, if H contains a simple triangle, then the problem is hard even in claw-free graphs, as mentioned before.

Finally, we show that for reflexive H the only "easy" cases are bi-arc graphs.

▶ **Theorem 3.** *For every fixed reflexive non-bi-arc graph H, there exist a, b, c, such that the $\textsc{LHom}(H)$ problem cannot be solved in time $2^{o(n)}$ in n-vertex $S_{a,b,c}$-free graphs, unless the ETH fails.*

Unfortunately, we were not able to provide the full complexity dichotomy for $S_{a,b,c}$-free graphs. We conjecture that the distinction between "easy" and "hard" cases is as follows.

▶ **Conjecture 4.** *Assume the ETH. Let H be a non-bi-arc graph. Then for all a, b, c, the $\textsc{LHom}(H)$ problem can be solved in time $2^{o(n)}$ in n-vertex $S_{a,b,c}$-free graphs if and only if none of the following conditions is satisfied:*

a) *H is predacious,*

b) *H contains a simple triangle,*

c) *has a factor that is not bi-arc and contains two incomparable vertices with loops.*

Full version of the paper. The proofs of some statements, marked with (♣), are omitted or just sketched. Complete proofs can be found in the full version of the paper [30].

2 Notation and preliminaries

For a positive integer n, by $[n]$ we denote the set $\{1, 2, \ldots, n\}$. For a set X and integer k, by 2^X we denote the family of all subsets of X and by $\binom{X}{k}$ (resp. $\binom{X}{\leqslant k}$) we denote the family of all subsets of X with exactly (resp. at most) k elements.

For two sets $X, Y \subseteq V(G)$, we say that X is *complete* to Y if every vertex from X is adjacent to every vertex from Y. For $v \in V(G)$, by $N_G(v)$ we denote the set of neighbors of v and by $N_G[v]$ we denote the set $N_G(v) \cup \{v\}$. Note that if v has a loop, then $v \in N_G(v)$, so $N_G(v) = N_G[v]$. We omit the subscript and write $N(v)$ and $N[v]$, respectively, if G is clear from the context.

We say that two vertices u, v of G are *incomparable* if $N(u) \not\subseteq N(v)$ and $N(v) \not\subseteq N(u)$. We say that a set S of vertices is *incomparable* if its elements are pairwise incomparable. Let H be a graph and suppose that there are two distinct vertices a, b of H, such that $N_H(a) \subseteq N_H(b)$. We observe that in any homomorphism to H, if some vertex is mapped to a, we can safely remap it to b. Thus, if for some instance (G, L) of the $\textsc{LHom}(H)$ problem and for some $v \in V(G)$ the list $L(v)$ contains a and b as above, then we can safely remove a from $L(v)$. Thus, without loss of generality, we can always assume that in any instance of $\textsc{LHom}(H)$ each list is an incomparable set in H.

For a graph H, by H^* we denote the bipartite graph with vertex set $\{a', a'' \mid a \in V(H)\}$ and edge set $\{a'b'' \mid ab \in E(H)\}$. We observe that H^* is connected if and only if H is connected and non-bipartite. Moreover, for bipartite H, the graph H^* consists of two disjoint copies of H. Feder et al. [13] proved that H is a bi-arc graph if and only H^* is a bi-arc graph. As H^* is bipartite, we can equivalently say that H is a bi-arc graph if and only if the complement of H^* is a circular-arc graph.

▶ **Definition 5** (Predator). *A* predator *is a tuple (a_1, a_2, b_1, b_2) of vertices, such that $a_1 \neq a_2, b_1 \neq b_2$, and $\{a_1, a_2\}$ and $\{b_1, b_2\}$ are incomparable sets, complete to each other.*

Figure 1 shows some examples of predators. Let us point out that the leftmost structure in Figure 1 is the only predator, which can be bipartite. It will play a special role in our hardness proofs; we call it an *incomparable* C_4. Observe that (a_1, a_2, b_1, b_2) is a predator in H, for some $a_1, a_2, b_1, b_2 \in V(H)$, if and only if $(a'_1, a'_2, b''_1, b''_2)$ is an incomparable C_4 in H^*. This implies the following observation.

▶ **Observation 6.** *A graph H contains a predator if and only if H^* contains an incomparable C_4.*

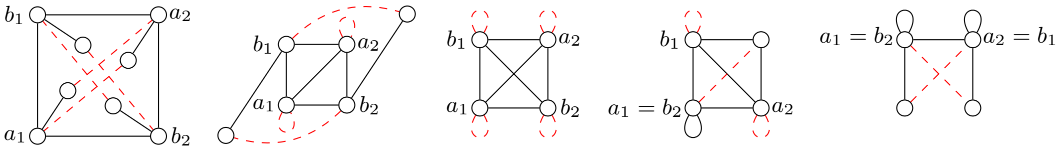

Figure 1 Examples of predators (a_1, a_2, b_1, b_2) and their neighbors. Red dashed lines denote the edges that cannot exist. The edges that are not drawn are possible, but not necessary.

We say that H is a *strong split graph* if $V(H)$ can be partitioned into two sets, P and B, such that $H[P]$ is a reflexive clique and B is independent.

For a bipartite graph H with bipartition classes X, Y, a *bipartite decomposition* is a partition of $V(H)$ into an ordered triple of sets (D, N, R), such that (i) N is non-empty and separates D and R, (ii) $|D \cap X| \geqslant 2$ or $|D \cap Y| \geqslant 2$, (iii) $(D \cup N) \cap X$ is complete to $N \cap Y$ and $(D \cup N) \cap Y$ is complete to $N \cap X$. We say that H is *undecomposable* if it admits no bipartite decomposition.

▶ **Theorem 7** (Okrasa et al. [28, 29]). *Let H be a graph. In time $|V(H)|^{\mathcal{O}(1)}$ we can construct a family $\mathcal{H}$ of $\mathcal{O}(|V(H)|)$ connected graphs, called* factors *of H, such that:*

(1) *H is a bi-arc graph if and only if every $H' \in \mathcal{H}$ is a bi-arc graph,*

(2) *for each $H' \in \mathcal{H}$, the graph H'^* is an induced subgraph of H^* and:*

  a. *H' is a bi-arc graph, or*

  b. *H' a strong split graph and has an induced subgraph H'', which is not a bi-arc graph and is an induced subgraph of H, or*

  c. *$(H')^*$ is undecomposable,*

(3) *for every instance (G, L) of* LHom*(H), the following implication holds:*
If there exists a non-decreasing, convex function $f : \mathbb{N} \to \mathbb{R}$, such that for every $H' \in \mathcal{H}$, for every induced subgraph G' of G, and for every H'-lists L' on G', we can decide whether $(G', L') \to H'$ in time $f(|V(G')|)$, then we can solve the instance (G, L) in time $\mathcal{O}\left(|V(H)| f(n) + n^2 \cdot |V(H)|^3\right)$.

Now we are ready to define the class of predacious graphs.

▶ **Definition 8** (Predacious graphs). *Let H be a graph and let $\mathcal{H}$ be the family of factors of H. We say that H is* predacious *if there exists $H' \in \mathcal{H}$ that is not a bi-arc graph and contains a predator.*

3 P_t-free graphs

3.1 Quasi-polynomial-time algorithm

We observe that to obtain Theorem 1 a), it is sufficient to prove the following.

▶ **Theorem 9.** *Let H be a fixed graph that does not contain a predator. Then for every t, the* LHom(H) *problem can be solved in time $n^{\mathcal{O}(\log^2 n)}$ in n-vertex P_t-free graphs.*

Indeed, suppose we have proven Theorem 9 and consider a non-predacious graph H, let $\mathcal{H}$ be the family of its factors given by Theorem 7. Since H is non-predacious, every $H' \in \mathcal{H}$ is either a bi-arc graph, or does not contain a predator. Thus, for each H' we can solve the LHom(H') problem in P_t-free graphs in polynomial time (in the first case) or in time $n^{\mathcal{O}(\log^2 n)}$, using Theorem 9 (in the second case). Now Theorem 1 a) follows from Theorem 7 (3).

Before we proceed to the proof of Theorem 9, let us show one crucial property of graphs H.

▶ **Observation 10.** *Let H be a graph which does not contain a predator. For any incomparable sets $X, Y \subseteq V(H)$, each of size at least 2, there exist $x \in X$ and $y \in Y$ such that $xy \notin E(H)$.*

Proof. For contradiction, suppose that there are two incomparable sets X, Y, each of size at least 2, which are complete to each other. Let x_1, x_2 be distinct elements from X, and y_1, y_2 be distinct elements from Y. Then (x_1, x_2, y_1, y_2) is a predator. ◀

So let us now prove Theorem 9. The algorithm follows the algorithm for 3-Coloring by Pilipczuk et al. [33], which is in turn inspired by the work of Gartland and Lokshtanov [17].

Sketch of proof of Theorem 9. Let (G, L) be an instance of LHom(H), such that graph G is P_t-free. We start with a preprocessing phase, in which we exhaustively perform the following steps. (1) If for some $v \in V(G)$ it holds that $L(v) = \emptyset$, then we terminate and report a no-instance. (2) If for some $v \in V(G)$, the list $L(v)$ contains two vertices $x, y \in V(H)$, such that $N_H(x) \subseteq N_H(y)$, then we remove x from $L(v)$. (3) If for some edge $uv \in E(G)$, and some $x \in L(u)$, the vertex x is non-adjacent in H to every $y \in L(v)$, then we remove x from $L(u)$. (4) If for some $v \in V(G)$ we have $|L(v)| = 1$, we remove v from G. Note that by the previous step the lists of neighbors of v contain only neighbors of the vertex in $L(v)$. (5) We enumerate all $S \in \binom{V(G)}{\leqslant t}$ and all possible H-colorings of $(G[S], L)$. If for some $v \in V(G)$ and some $x \in L(v)$, for some $S \in \binom{V(G)}{\leqslant t}$ such that $v \in S$ there is no $h : (G[S], L) \to H$ such that $h(v) = x$, we remove x from $L(v)$.

We will continue calling the current instance (G, L), let n be its number of vertices of G. The instance satisfies the following properties.

(P1) For every $v \in V(G)$, the set $L(v)$ is incomparable and has at least two elements.

(P2) For every $v \in V(G)$, every $S \in \binom{V(G)}{\leqslant t}$, such that $v \in S$, and every $x \in L(v)$, there exists $h : (G[S], L) \to H$ which maps v to x.

Now let us describe the algorithm. If $n \leqslant 1$, then we report a yes-instance; recall that by property (P1) each list is non-empty. If the instance G is disconnected, we call the algorithm for each connected component independently. If none of the above cases occurs, we perform branching. We will carefully choose a *branching pair* (v, x), where $v \in V(G)$ and $x \in L(v)$, and branch into two possibilities. In the first one, called the *successful* branch, we call the algorithm recursively with the list of v set to $\{x\}$. In the second branch we call the algorithm with x removed from $L(v)$. We report a yes-instance if at least one of the branches reports a yes-instance.

Now let us discuss how we select a branching pair. For each $\{u, u'\} \in \binom{V(G)}{2}$ we define the *bucket* $\mathcal{B}_{u,u'}$. The elements of $\mathcal{B}_{u,u'}$ are all possible pairs (P, h), where P is an induced u-u'-path and h is a list homomorphism from (P, L) to H. We will refer to pairs (P, h) as *colored paths*.

Note that since G is P_t-free, the total size of all buckets is $\mathcal{O}(n^t)$ and they can be enumerated in polynomial time. Furthermore, by property (P2), we know that $\mathcal{B}_{u,u'}$ is non-empty if and only if u and u' are in the same connected component of G. Even more, if w belongs to an induced u-u'-path P, and $x \in L(w)$, then $\mathcal{B}_{u,u'}$ contains a colored path (P, h), such that $h(w) = x$.

Define

$$\delta := \frac{1}{2^{|V(H)|+1} \cdot t} \qquad \text{and} \qquad \varepsilon := \frac{1}{2^{|V(H)|+1} \cdot |V(H)|^t \cdot t} = \frac{\delta}{|V(H)|^t}.$$

▷ Claim 11. If G is a connected P_t-free graph, then there is a pair (v, x), where $v \in V(G)$ and $x \in L(v)$, with the following property. There is a set $Q \subseteq \binom{V(G)}{2}$ of size at least $\delta \cdot \binom{n}{2}$, such that for every $\{u, u'\} \in Q$ there is a subset $\mathcal{P}_{u,u'} \subseteq \mathcal{B}_{u,u'}$ of size at least $\varepsilon \cdot |\mathcal{B}_{u,u'}|$, such that for every $(P, h) \in \mathcal{P}_{u,u'}$, there is $w_P \in V(P) \cap N[v]$, such that $h(w_P) \notin N_H(x)$.

Proof. For $\{u, u'\} \in \binom{V(G)}{2}$, let $\theta(u, u')$ denote the number of induced u-u'-paths in G. By [33, Lemma 5], there is a vertex $v \in V(G)$, such that for at least $\frac{1}{2t}\binom{n}{2}$ pairs $\{u, u'\} \in \binom{V(G)}{2}$ and for at least $\frac{1}{2t}\,\theta(u, u')$ induced u-u'-paths P, the set $N[v]$ intersects $V(P)$. Since the number of distinct H-lists is at most $2^{|V(H)|}$, we observe that by the pigeonhole principle there is a list $L' \subseteq V(H)$ and a subset $Q \subseteq \binom{V(H)}{2}$ of size at least $\frac{1}{2^{|V(H)|+1} \cdot t}\binom{n}{2} = \delta \cdot \binom{n}{2}$, such that for every $\{u, u'\} \in Q$ there exists a set $\mathcal{P}_{u,u'}$ of at least $\delta \cdot \theta(u, u')$ induced u-u'-paths, with the property that for every $P \in \mathcal{P}_{u,u'}$ there exists $w_P \in N[v] \cap V(P)$, such that $L(w_P) = L'$.

By property (P1) we know that each of $L(v)$ and L' is an incomparable set with at least two elements. Thus by Observation 10 there are $x \in L(v)$ and $y \in L'$, which are non-adjacent in H.

Let us argue that the pair (v, x) satisfies the desired conditions. Fix some $\{u, u'\} \in Q$. As every induced u-u' path has at most $t-1$ elements, we have that $|\mathcal{B}_{u,u'}| \leqslant |V(H)|^t \cdot \theta(u, u')$. On the other hand, by property (P2) for every $P \in \mathcal{P}_{u,u'}$ there exists a homomorphism $h : (P, L) \to H$ such that $h(w_P) = y \notin N_H(x)$. So, summing up, we obtain that the number of such pairs $(P, h) \in \mathcal{B}_{u,u'}$ is at least $|\mathcal{P}_{u,u'}| \geqslant \delta \cdot \theta(u, u') \geqslant \frac{\delta}{|V(H)|^t} \cdot |\mathcal{B}_{u,u'}| = \varepsilon \cdot |\mathcal{B}_{u,u'}|$. ◁

Consider the successful branch for the branching pair (v, x) given by Claim 11. For some $\{u, u'\} \in Q$, let (P, h) be a colored path in $\mathcal{P}_{u,u'}$, and let w_P be as in the claim. Consider the preprocessing phase of the current call. If $w_P = v$, then w_P is removed from the graph, so (P, h) will no longer appear in the bucket of $\{u, u'\}$. Similarly, if $w_P \neq v$, then we remove $h(w_P)$ from $L(w_P)$, so (P, h) will not appear in the bucket of $\{u, u'\}$. Thus when we branch using the pair (v, x), in the successful branch we remove an ε-fraction of elements in a δ-fraction of buckets. This gives the quasi-polynomial running time, we refer to the full version of the paper for a detailed complexity analysis (♣). ◀

3.2 Hardness results for P_t-free graphs

Let H be a predacious graph and let $\mathcal{H}$ be the family of factors of H. Since H is predacious, there is some non-bi-arc factor $H' \in \mathcal{H}$, which contains a predator. By Theorem 7 (2) there are two possible cases:

Case A. H' is a strong split graph as in Theorem 7 (2b) (every such graph H' contains a predator, but we will not use it explicitly), and

Case B. $(H')^*$ is an undecomposable induced subgraph of H^*.

Case A: Strong split target graphs. We show that for strong split graphs H' the $\text{LHom}(H')$ problem remains hard even if the instance is a split graph, i.e., its vertex set can be partitioned into a clique and an independent set. Equivalently, split graphs are $\{C_4, C_5, 2P_2\}$-free graphs.

▶ **Theorem 12.** *Let H' be a fixed non-bi-arc strong split graph. Then the $\text{LHom}(H')$ problem cannot be solved in time $2^{o(n)}$ in n-vertex split graphs, unless the ETH fails.*

Proof. Let P be the set of vertices in H' that have loops, and let B be the set of vertices of H' without loops. Consider an instance (G, L) of $\text{LHom}(H')$. Recall that without loss of generality we can assume that each list $L(v)$ is an incomparable set. As for every $p \in P$ and $b \in B$ it holds that $N_{H'}(b) \subseteq N_{H'}(p)$, no vertex in G has both a vertex from P and a vertex from B in its list. Since every list is non-empty, we can partition the vertex set of $V(G)$ into two sets:

$$X := \{v \in V(G) \mid L(v) \cap P \neq \emptyset\} \quad \text{and} \quad Y := \{v \in V(G) \mid L(v) \cap B \neq \emptyset\}.$$

Furthermore, as B is independent, we can assume that Y is independent; otherwise (G, L) is a no-instance. Let G' be obtained from G by adding all edges with both endvertices in X (except for loops). It is straightforward to verify that $(G, L) \to H'$ if and only if $(G', L) \to H'$. ◀

Now we can show the main result of this subsection.

Proof of Theorem 1 b) in Case A. Let H be as in Case A and let H', H'' be as in Theorem 7 (2b). Since H'' is an induced subgraph of H', it is also a strong split graph, so by Theorem 12 we know that $\text{LHom}(H'')$ admits no subexponential-time algorithm in split graphs. As H'' is an induced subgraph of H, every instance of $\text{LHom}(H'')$ is also an instance of $\text{LHom}(H)$, and we are done. ◀

Case B: Target graphs with the associated bipartite graph undecomposable. First we consider bipartite, undecomposable, non-bi-arc graphs H, which contain a predator. Recall that the only bipartite predator is an incomparable C_4. We will prove the following.

▶ **Theorem 13.** *Let H be a fixed, bipartite, non-bi-arc, undecomposable graph, which contains an incomparable C_4. Then there exists t, such that $\text{LHom}(H)$ cannot be solved in time $2^{o(n)}$ in n-vertex P_t-free graphs, unless the ETH fails.*

Before we proceed to the proof of Theorem 13, we need to introduce some tools which we will need. For a pair of vertices (a, b) of $V(H)$, an $\text{OR}_3(a, b)$-*gadget* is an instance (F, L) of $\text{LHom}(H)$ with *interface vertices* $o_1, o_2, o_3 \in V(F)$, such that $L(o_1) = L(o_2) = L(o_3) = \{a, b\}$, and

$$\{f(o_1)f(o_2)f(o_3) \mid f : (F, L) \to H\} = \{aaa, aab, aba, baa, abb, bab, bba\}.$$

For an incomparable set of vertices S, such that $|S| \geqslant 2$, a $\text{NEQ}(S)$-*gadget* is an instance (F, L) of $\text{LHom}(H)$ with *interface vertices* $s_1, s_2 \in V(F)$, such that $L(s_1) = L(s_2) = S$, and

$$\{f(s_1)f(s_2) \mid f : (F, L) \to H\} = \{uv \mid u, v \in S, u \neq v\}.$$

The following structural result is proven by Okrasa et al. [29, Lemma 19 and Corollary 20].

▶ **Lemma 14** (Okrasa et al. [29]). *Let H be a connected, bipartite, non-bi-arc, undecomposable graph with bipartition classes X and Y. Then there exist two incomparable sets of vertices $\{\alpha, \beta\} \subseteq X$ and $\{\alpha', \beta'\} \subseteq Y$, such that $\alpha\alpha', \beta\beta' \in E(H)$, $\alpha\beta', \beta\alpha' \notin E(H)$, and the following conditions hold.*

(1) *For any incomparable two-element set $\{a, b\} \subseteq V(H)$, and for any $\{\gamma, \delta\} \in \{\{\alpha, \beta\}, \{\alpha', \beta'\}\}$, such that $\{a, b, \gamma, \delta\}$ is contained in one bipartition class, there exist a path $D^{\gamma/\delta}_{a/b}$ with endvertices x, y and H-lists L, such that $L(x) = \{a, b\}$, $L(y) = \{\gamma, \delta\}$, and:*
- **(D1)** *there is a list homomorphism $h_a : (D^{\gamma/\delta}_{a/b}, L) \to H$, such that $h_a(x) = a$ and $h_a(y) = \gamma$,*
- **(D2)** *there is a list homomorphism $h_b : (D^{\gamma/\delta}_{a/b}, L) \to H$, such that $h_b(x) = b$ and $h_b(y) = \delta$,*
- **(D3)** *there is no list homomorphism $h : (D^{\gamma/\delta}_{a/b}, L) \to H$, such that $h(x) = a$ and $h(y) = \delta$.*

(2) *There exist an $\mathrm{OR}_3(\alpha, \beta)$-gadget and an $\mathrm{OR}_3(\alpha', \beta')$-gadget.*

▶ **Lemma 15** ([29]). *Let H be a connected, bipartite, non-bi-arc, undecomposable graph, let $S \subseteq V(H)$ be an incomparable set contained in one bipartition class of H. Then there exists a $\mathrm{NEQ}(S)$-gadget.*

We use Lemma 14 and Lemma 15 to construct the so-called *occurrence gadget*.

▶ **Lemma 16.** *Let H be a connected, bipartite, non-bi-arc and undecomposable graph, and let $\{a, b\}$, γ, δ be as in Lemma 14 (1). Then there is a $\mathrm{Var}(a, b)$-gadget (G, L) with interface vertices v, t, f, such that $L(v) = \{a, b\}$, $L(t) = L(f) = \{\gamma, \delta\}$, and:*

(1) *for any homomorphism $h : (G, L) \to H$, if $h(v) = a$, then $h(t) = \gamma$ and $h(f) = \delta$,*

(2) *for any homomorphism $h : (G, L) \to H$, if $h(v) = b$, then $h(t) = \delta$ and $h(f) = \gamma$.*

Proof. We use Lemma 14 to construct gadgets $(D^{\gamma/\delta}_{a/b}, L)$ and $(D^{\gamma/\delta}_{b/a}, L)$ with endvertices, respectively, $x_1, y_1 \in V(D^{\gamma/\delta}_{a/b})$ and $x_2, y_2 \in V(D^{\gamma/\delta}_{b/a})$. We then use Lemma 15 for $S = \{\gamma, \delta\}$ to construct a $\mathrm{NEQ}(S)$-gadget (F, L) with interface vertices $s_1, s_2 \in V(F)$.

We identify vertices x_1 and x_2 into a single vertex v. We identify vertices y_1 and s_1 into a single vertex t, and we identify vertices y_2 and s_2 into a single vertex f, see Figure 2 (top left). ◀

We proceed to the proof of Theorem 13.

Proof of Theorem 13. Let (a_1, a_2, b_1, b_2) be an incomparable C_4 in H. Let X and Y be the bipartition classes of H, so that $a_1, a_2 \in X$ and $b_1, b_2 \in Y$.

We reduce from 3-Sat. Consider a formula Φ of 3-Sat with variables $x_1, \ldots, x_N$ and clauses $C_1, \ldots, C_M$. We can assume that each clause has exactly three literals. We construct an instance (G_Φ, L) of LHom(H) as follows. We introduce a biclique with partite sets $V := \{v_1, \ldots, v_N\}$ and $U := \{u_1, \ldots, u_{3M}\}$. Vertices in V correspond to the variables of Φ, while vertices in U correspond to literals in Φ, i.e., the occurrences of the variables in clauses. For a clause C_i, by U_i we denote the three-element subset of vertices of U corresponding to the literals of C_i. For every $j \in [N]$ we set $L(v_j) := \{a_1, a_2\}$ and for every $i \in [3M]$ we set $L(u_i) := \{b_1, b_2\}$.

Mapping the vertex v_j to a_1 (a_2, resp.) will correspond to making the variable v_j true (false, resp.). Similarly, we will interpret u_j being mapped to b_1 (b_2, resp.) as setting the corresponding literal true (false, resp.). So we need to ensure that (i) the coloring of vertices in V is consistent with the coloring of vertices in U, and (ii) for each clause C_i, at least one vertex in U_i is mapped to b_1.

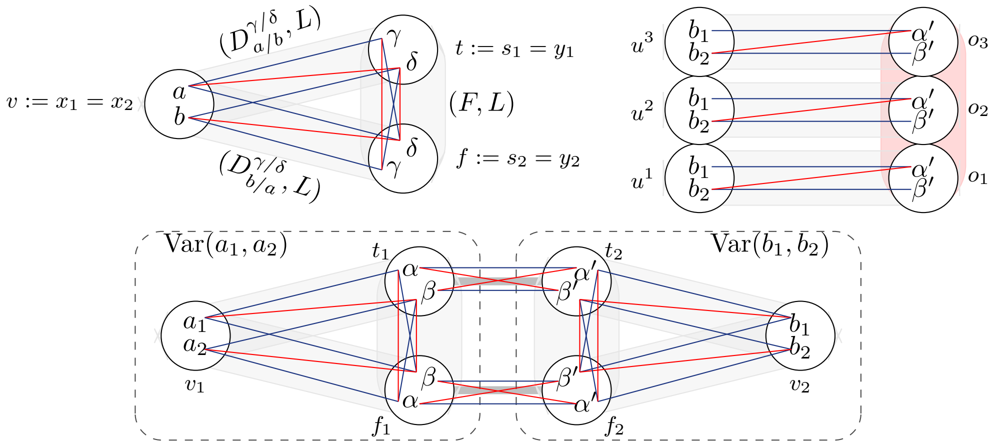

Figure 2 A schematic view of a $\mathrm{Var}(a, b)$-gadget (top left), an $\mathrm{OR}_3(b_1, b_2)$-gadget (top right) and a positive occurrence gadget (bottom). On every picture, the blue lines indicate that there exists an H-coloring of the respective part of the graph, which assigns chosen values to white vertices, and the red ones indicate that there is no such H-coloring. The red area indicates an $\mathrm{OR}_3(\alpha', \beta')$-gadget with interface vertices o_1, o_2, o_3.

To ensure property (i), we will introduce two types of *occurrence gadgets*. We use Lemma 16 to construct two variable gadgets $\mathrm{Var}(a_1, a_2)$ and $\mathrm{Var}(b_1, b_2)$ and add an edge between their t-vertices and another one between f-vertices. This way we obtain a *positive occurrence gadget*, see Figure 2 (bottom). A *negative occurrence gadget* is obtained from a positive occurrence gadget by adding a copy of a $\mathrm{NEQ}(\{b_1, b_2\})$-gadget, constructed by Lemma 15, with interface vertices s_1, s_2, and identifying s_1 with v_2. The occurrence gadgets have two special vertices: a *variable vertex* v_1, and a *literal vertex*, which is v_2 for the positive occurrence gadget, and s_2 for the negative occurrence gadget. Consider a vertex $u_i \in U$, which corresponds to an occurrence of a variable x_j, and thus to the vertex v_j. If u_i corresponds to a positive (resp., negative) literal, we introduce a positive (resp., negative) occurrence gadget, and identify v_j with its variable vertex and u_i with its literal vertex. One can readily verify that the constructed gadgets can indeed be used to ensure property (i).

Consider a set $U_i = \{u^1, u^2, u^3\}$, corresponding to the literals of some clause C_i. We observe that in order to ensure property (ii), we need to construct an $\mathrm{OR}_3(b_1, b_2)$-gadget, whose interface vertices are precisely u^1, u^2, and u^3. We call Lemma 14 to construct an $\mathrm{OR}_3(\alpha', \beta')$-gadget with interface vertices o_1, o_2, o_3 and three copies of the graph $D^{\beta'/\alpha'}_{b_2/b_1}$. For $s \in \{1, 2, 3\}$, we identify one endvertex of the s-th copy of $D^{\beta'/\alpha'}_{b_2/b_1}$ (the one with the list $\{b_1, b_2\}$) with u^s, and the other endvertex (the one with the list $\{\alpha', \beta'\}$) with o_s, see Figure 2 (top right). Again, it is straightforward to verify that the constructed subgraph is indeed an $\mathrm{OR}_3(b_1, b_2)$-gadget with interface vertices u^1, u^2, u^3.

The discussion above implies that $(G_\Phi, L) \to H$ if and only if Φ is satisfiable. Let t' be the maximum of the numbers of vertices in the negative occurrence gadget and in the $\mathrm{OR}_3(b_1, b_2)$-gadget and define $t := t' + 4$. By a simple case analysis it can be verified that G_Φ is P_t-free (♣). ◀

Finally, we can prove Theorem 1 b) in Case B.

Proof of Theorem 1 b) in Case B. For contradiction, suppose that there exists a graph H, satisfying the assumptions, and for every t there is an algorithm A_t, which solves every P_t-free instance of $\text{LHom}(H)$ in subexponential time. Let H' be a factor of H as in the assumptions of Case B and observe that H'^* satisfies the assumptions of Theorem 13. Let t be given by Theorem 13 for H'^*.

Let (G, L') be an instance of $\text{LHom}(H'^*)$ constructed as in the proof of Theorem 13. Since H'^* is an induced subgraph of H^*, (G, L') is also an instance of $\text{LHom}(H^*)$. Note that G is bipartite and P_t-free, and no list intersects both bipartition classes of H^*. Define $L : V(G) \to 2^{V(H)}$ as follows: $L(v) := \{a \mid \{a', a''\} \cap L(v) \neq \emptyset\}$. It is straightforward to verify that $(G, L') \to H^*$ if and only if $(G, L) \to H$ [29, Proposition 43]. Thus we can use A_t to decide if $(G, L) \to H$ or, equivalently, if $(G, L') \to H'^*$, in subexponential time. By Theorem 13 this contradicts the ETH. ◀

4 $S_{a,b,c}$-free graphs

4.1 Subexponential-time algorithm for $\{S_{a,b,c}, K_3\}$-free graphs

To describe the algorithm, we first need to introduce the notion of an *extended strip decomposition* [6, 5]. For a graph G, by $T(G)$ we denote the set of all triangles in G, i.e., three-element sets $\{x, y, z\}$ of pairwise adjacent vertices. We will denote a triangle $\{x, y, z\}$ shortly by xyz.

Let G be a simple graph. An *extended strip decomposition* (D, η) of G consists of:

- a simple graph D and a function $\eta : V(D) \cup E(D) \cup T(D) \to 2^{V(G)}$,
- for each $xy \in E(D)$, subsets $\eta(xy, x), \eta(xy, y) \subseteq \eta(xy)$,

which satisfy the following properties:

1. $\{\eta(o) \mid o \in V(D) \cup E(D) \cup T(D)\}$ is a partition of $V(G)$,
2. for every $x \in V(D)$ and every distinct $y, z \in N_D(x)$, the set $\eta(xy, x)$ is complete to $\eta(xz, x)$,
3. every $uv \in E(G)$ is contained in one of the sets $\eta(o)$ for $o \in V(D) \cup E(D) \cup T(D)$ or:
   - $u \in \eta(xy, x), v \in \eta(xz, x)$ for some $x \in V(D)$ and $y, z \in N_D(x)$, or
   - $u \in \eta(xy, x), v \in \eta(x)$ for some $xy \in E(D)$, or
   - $u \in \eta(xyz)$ and $v \in \eta(xy, x) \cap \eta(xy, y)$ for some $xyz \in T(D)$.

We will sometimes refer to elements of $V(D) \cup E(D) \cup T(D)$ as *objects* of D.

The following subsets of $V(G)$ are called *atoms* of a decomposition (D, η): (1) for an object $o \in V(D) \cup T(D)$, the set $\eta(o)$, (2) for $xy \in E(D)$, the set $\eta(xy) - (\eta(xy, x) \cup \eta(xy, y))$, the set $\eta(x) \cup \eta(xy) - \eta(xy, y)$, and the set $\eta(x) \cup \eta(y) \cup \eta(xy) \cup \bigcup_{xyz \in T(D)} \eta(xyz)$.

The following theorem is the main combinatorial tool used in our algorithm.

▶ **Theorem 17** (♣). *Let $t \geqslant 4$, $\sigma \in (0, \frac{1}{100t})$, and let G be a connected $(S_{t,t,t}, K_3)$-free graph on n vertices with $\Delta(G) < \sigma^8 \cdot n$. Then there exists $X \subseteq V(G)$ and an extended strip decomposition (D, η) of $G - X$ with each atom of size at most α, such that:*

(1) *$\alpha \leqslant (1 - \sigma^7)n$ and $|X| \leqslant \sigma(n - \alpha)$,*
(2) *$\eta(xyz) = \emptyset$ for every $xyz \in T(D)$,*
(3) *D is a simple graph with maximum degree at most 2,*
(4) *if for some edge xy of D we have $\eta(xy, x) = \emptyset$, then x is of degree 1 in D.*

Sketch of proof. By a result of Chudnovsky et al. [5, Lemma 6.5] there is $X \subseteq V(G)$ and an extended strip decomposition (D', η') of $G - X$, satisfying (1). We aim to modify (D', η') in order to obtain a decomposition with the desired structure. We will still denote the extended strip decomposition obtained after each step of modification as (D', η').

To ensure properties (2), (3), and (4), we use the fact that G is triangle-free. Let us start with (2) and consider a triangle xyz in D'. Define $A_{xy} := \eta(xy, x) \cap \eta(xy, y)$. Sets A_{yz} and A_{xz} are defined in an analogous way. Recall that the neighborhood of $\eta(xyz)$ is contained in $A_{xy} \cup A_{yz} \cup A_{xz}$. However, these three sets are complete to each other, so at least one of them must be empty. It turns out that we can "absorb" $\eta(xyz)$ into $\eta(o)$, where o is either a vertex or an edge of xyz, without violating the properties of an extended strip decomposition and increasing the maximum atom size.

Now let us discuss property (3). Suppose D' has a vertex x with at least three neighbors, say y, y', y''. As the sets $\eta(xy, x)$, $\eta(xy, x)$, $\eta(xy'', x)$ are pairwise complete to each other, at least one of them, say $\eta(xy, x)$, must be empty. We introduce a new vertex x' to D', add the edge $x'y$ with $\eta(x'y) := \eta(xy)$ and $\eta(x'y, y) := \eta(xy, y)$, and remove xy from D'. Observe that the degree of x was reduced by 1. We repeat this step exhaustively. Property (4) is ensured in a similar way. ◀

▶ **Theorem 18 (♣).** *Let H be a connected graph with no predator. Then for every $a, b, c \geqslant 0$, the* LHom*(H) problem can be solved in time $2^{\mathcal{O}(n^{8/9} \log n)}$ in n-vertex $\{S_{a,b,c}, K_3\}$-free graphs.*

Sketch of proof. We assume that n is large, as otherwise we solve the problem exhaustively. We will present a recursive algorithm. Let $F(n)$ be the running time bound on instances with n vertices.

Similarly as we did in Section 3.1, we can ensure that every list is an incomparable set of size at least two and for every $uv \in E(G)$ and every $a \in L(v)$ there exists $b \in L(u)$ such that $ab \in E(H)$.

First, suppose that exists a vertex $v \in V(G)$ such that $\deg_G(v) \geqslant n^{1/9}$. This implies that there exists a list L' assigned to at least $\ell := n^{1/9}/2^{|V(H)|}$ neighbors of v. By Observation 10 there exist $a \in L(v)$ and $b \in L'$ such that $ab \notin E(H)$. We branch on assigning a to v; either we remove a from $L(v)$ or color v with a and remove b from the lists of all neighbors of v. Since in the second branch at least ℓ lists are shortened, we obtain that the complexity in this case is $F(n) = 2^{\mathcal{O}(n^{8/9} \log n)}$.

Now suppose that the maximum degree of G is smaller than $n^{1/9}$. Theorem 17 called for $t := \max(a, b, c, 4)$ and $\sigma := n^{1-/9}$ yields $X \subseteq V(G)$ and an extended strip decomposition (D, η) of $G - X$, satisfying the conditions stated in the statement. Let α be the maximum size of an atom of (D, η). If $x \in V(D)$ has two neighbors y and z, then, by Theorem 17 (4), $\eta(xy, x) \neq \emptyset$ and $\eta(xz, x) \neq \emptyset$. As $\eta(xy, x)$ is complete to $\eta(xz, x)$ and the maximum degree of G is at most $n^{1/9}$, we observe that $|\eta(xy, x)| < n^{1/9}$ and $|\eta(xz, x)| < n^{1/9}$.

We proceed as follows. We exhaustively guess the H-coloring of vertices of X; there are at most $|V(H)|^{|X|}$ possibilities. In each branch we need to decide if the H-coloring of X can be extended to all vertices of G. For an edge $uv \in E(G)$, such that $u \in X$ and $v \notin X$, we remove from $L(v)$ every non-neighbor of the color of u. Now the problem is reduced to solving the instance of LHom(H) on each component G' of $G - X$ independently. We observe that $V(G') \subseteq \bigcup_{o \in V(D') \cup E(D') \cup T(D')} \eta(o)$ for some connected component D' of D. Recall that D' is a path or a cycle.

▷ Claim 19 (♣). We can solve the instance (G', L) of LHom(H) in time $|V(H)|^{4n^{1/9}} \cdot F(\alpha) \cdot n^{\mathcal{O}(1)}$.

Sketch of Proof. Suppose D' is a path with consecutive vertices $x_1, \ldots, x_m$. If $m \leqslant 2$, then $|V(G')| \leqslant \alpha$ and we solve the problem recursively in time $F(\alpha)$. Otherwise, for every edge $x_i x_{i+1}$, except for $x_1 x_2$ and $x_{m-1} x_m$, we enumerate all pairs (f, g), such that $f : (G[\eta(x_i x_{i+1}, x_i)], L) \to H$ and $g : (G[\eta(x_i x_{i+1}, x_{i+1})], L) \to H$. Now for each such pair we verify if the partial H-coloring given by f and g can be extended to an H-coloring of $G([\eta(x_ix_{i+1})], L)$. We can do it in time $F(\alpha)$ by recursively solving an appropriate instance of LHom(H). Similarly, for each vertex x_i, except for x_1, x_m, we enumerate all list H-colorings of $\eta(x_{i-1}x_i, x_i)$ and $\eta(x_ix_{i+1}, x_i)$ and recursively check if they can be extended to a homomorphism $(G[\eta(x_i)], L) \to H$. To deal with the extremities of D', for each coloring of $\eta(x_1x_2, x_2)$ (resp. $\eta(x_{m-1}x_m, x_{m-1})$) we test if it can be extended to an H-coloring of $(G[\eta(x_1) \cup \eta(x_1x_2)], L)$ (resp. $(G[\eta(x_m) \cup \eta(x_{m-1}x_m)], L)$); note that each of these instances has at most α vertices. Then we use dynamic programming to decide if $(G', L) \to H$.

Suppose now that D' is a cycle $x_1, x_2, \ldots, x_m$. We guess the coloring of $\eta(x_1x_2, x_1)$, adjust the lists of its neighbors, and remove $\eta(x_1x_2, x_1)$ from G'. We modify D' by introducing a vertex x_1' and the edge $x_1'x_2$ to D', and removing the edge x_1x_2. Sets η are modified as in the proof of Claim 19. After the modification D' is a path and we continue as in the previous case. ◁

So let us now estimate the total running time in case that the maximum degree of G is at most $n^{1/9}$. Recall that we exhaustively guess the coloring of $|X|$ and then, for every connected component of $G - X$, we try to extend it, using Claim 19. Thus the overall complexity $F(n)$ in the considered case is described by $F(n) \leqslant |V(H)|^{|X|} \cdot |V(H)|^{4n^{1/9}} \cdot F(\alpha) \cdot n^{\mathcal{O}(1)}$. Applying $|X| \leqslant n^{-1/9}(n - \alpha)$ and $1 \leqslant \alpha \leqslant n - n^{2/9}$, and the inductive assumption, we conclude that $F(n) = 2^{\mathcal{O}(n^{8/9} \log n)}$. ◀

Combining Theorem 18 with Theorem 7, we immediately obtain the following corollary.

▶ **Corollary 20.** *Let H be a non-predacious graph. Then for every $a, b, c \geqslant 0$, the LHom(H) problem can be solved in time $2^{\mathcal{O}(n^{8/9} \log n)}$ in n-vertex $\{S_{a,b,c}, K_3\}$-free graphs.*

Now Theorem 2 a) follows from Corollary 20. Since H is irreflexive and triangle-free, if G is not triangle-free, we report a no-instance. In the other case, we use the algorithm from Corollary 20.

4.2 Hardness results

▶ **Theorem 3 b).** *Let H be a graph, which is predacious or contains a simple triangle. Then there is t, such that LHom(H) cannot be solved in time $2^{o(n)}$ in n-vertex $S_{t,t,t}$-free graphs, unless the ETH fails.*

Proof. The first case of the theorem follows directly from Theorem 1 b), as P_t-free graphs are $S_{t,t,t}$-free. The second case follows from the hardness of 3-Coloring in line graphs [23]. ◀

▶ **Theorem 21.** *Let H be a connected non-bi-arc graph such that H^* is undecomposable and there are three distinct vertices u_1, u_2, u_3 of H with loops, such that $S = \{u_1, u_2, u_3\}$ is incomparable. Then there is t, such that LHom(H) cannot be solved in time $2^{o(n)}$ in $S_{t,t,t}$-free graphs, unless the ETH fails.*

Proof. Let G be an instance of 3-Coloring with $V(G) = \{v_1, v_2, \ldots, v_N\}$. We construct an instance (G', L) of LHom(H) such that G is 3-colorable if and only if $(G', L) \to H$. First, for every $i \in [N]$ we introduce to G' a graph K^i, which is a complete graph with the vertex set $V(K^i) := \{x_{ij} \mid v_j \in N_G(v_i)\}$. Intuitively, the vertex x_{ij} represents the connection of v_i and v_j from the point of view of v_i. We set $L(x_{ij}) := S$ for all relevant i, j. Now, for each edge v_iv_j of G, we introduce a copy of the NEQ(S)-gadget given by Lemma 15, and identify its two interface vertices with x_{ij} and x_{ji}, respectively. Suppose for now that we can ensure the following property.

($\star$) For each $i \in [N]$ and each $f : (K^i, L) \to H$, all vertices of K^i are mapped to the same element of S, and for each $u \in S$ there is $f : (K^i, L) \to H$ that maps all vertices of K^i to u.

With the property above at hand, we can interpret the mapping of vertices in K^i as coloring v_i with one of three possible colors. The properties of the NEQ(S)-gadget imply that G is 3-colorable if and only if the constructed graph admits a list homomorphism to H.

Now let us argue how to ensure property ($\star$). For each $i \in [N]$ we add an independent set Q^i and make it complete to K^i. The size of Q^i and the lists of its vertices depend on the structure of H.

For $\{\ell, \ell', \ell''\} = [3]$, a *private neighbor* of $u_\ell \in S \subseteq V(H)$ is a vertex $w_\ell \in N(u_\ell) - (N(u_{\ell'}) \cup N(u_{\ell''}))$. Note that if u_ℓ does not have a private neighbor, then, since S is incomparable, there exist $w_{\ell\ell'} \in N(u_\ell) \cap N(u_{\ell'}) - N(u_{\ell''})$ and $w_{\ell\ell''} \in N(u_\ell) \cap N(u_{\ell''}) - N(u_{\ell'})$.

We consider three cases. If for each $\ell \in [3]$, the vertex u_ℓ has a private neighbor, then $Q^i := \{q^i\}$, and $L(q^i) := \{w_1, w_2, w_3\}$. Otherwise, if there are exactly two vertices in S which have private neighbors, say u_2 and u_3, we set $Q^i := \{q^i, r^i\}$, $L(q^i) := \{w_{12}, w_2, w_3\}$ and $L(r^i) := \{w_{13}, w_2, w_3\}$. Last, if there is at most one vertex in S which has private neighbors, say u_3, we set $Q^i := \{q^i, r^i, s^i\}$ and $L(q^i) := \{w_{12}, w_{13}\}$, $L(r^i) := \{w_{12}, w_{23}\}$, and $L(s^i) := \{w_{13}, w_{23}\}$. It is straightforward to verify that in each of the above cases the property ($\star$) holds.

That completes the construction of (G', L). By the reasoning above we observe that $(G', L) \to H$ if and only if G is 3-colorable. Let $t \geqslant 2$ be the number of vertices in the NEQ(S)-gadget given by Lemma 15. A straightforward analysis of the structure of G' implies that G' is $S_{t,t,t,}$-free (♣). ◀

With Theorem 21 at hand, we can prove Theorem 3.

Proof of Theorem 3. Let H' be a vertex-minimal induced non-bi-arc subgraph of H. Feder and Hell [11] proved that H' (i) is an induced cycle with at least four vertices, or (ii) consists of an independent set $\{x, y, z\}$ and three paths, each joining two vertices from $\{x, y, z\}$ and avoiding the neighborhood of the third one. The minimality of H' implies that H'^* is undecomposable (see e.g. [9]). Now observe that H' contains an incomparable set of size 3: in case (i) we can take any three vertices of H', and in case (ii) this set is $\{x, y, z\}$. Thus the claim follows from Theorem 21. ◀

5 Conclusion

Recall that while for P_t-free graphs, in Theorem 1 we were able to fully characterize the "easy" and "hard" cases of LHom(H), for the case of $S_{a,b,c}$-free graphs we obtained a full dichotomy only for irreflexive (Theorem 2) and for reflexive (Theorem 3) graphs H. In order to complete the dichotomy, we need to consider graphs H that are neither irreflexive nor reflexive. Some hardness results for such graphs follow already from Theorem 3 b) and Theorem 21. We were also able to obtain a few more ad-hoc hardness results, which we do not present here. All our results seem to support the following conjecture.

▶ **Conjecture 4.** *Assume the ETH. Let H be a non-bi-arc graph. Then for all a, b, c, the LHom(H) problem can be solved in time $2^{o(n)}$ in n-vertex $S_{a,b,c}$-free graphs if and only if none of the following conditions is satisfied:*

a) *H is predacious,*

b) *H contains a simple triangle,*

c) *has a factor that is not bi-arc and contains two incomparable vertices with loops.*

References

1. Flavia Bonomo, Maria Chudnovsky, Peter Maceli, Oliver Schaudt, Maya Stein, and Mingxian Zhong. Three-coloring and list three-coloring of graphs without induced paths on seven vertices. *Combinatorica*, 38(4):779–801, 2018. doi:10.1007/s00493-017-3553-8.
2. Andrei A. Bulatov. H-Coloring dichotomy revisited. *Theor. Comput. Sci.*, 349(1):31–39, 2005. doi:10.1016/j.tcs.2005.09.028.
3. Maria Chudnovsky, Shenwei Huang, Pawel Rzążewski, Sophie Spirkl, and Mingxian Zhong. Complexity of C_k-coloring in hereditary classes of graphs. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 31:1–31:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.31.
4. Maria Chudnovsky, Jason King, Michał Pilipczuk, Paweł Rzążewski, and Sophie Spirkl. Finding large H-colorable subgraphs in hereditary graph classes. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 35:1–35:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.35.
5. Maria Chudnovsky, Marcin Pilipczuk, Michal Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the maximum weight independent set problem in H-free graphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2260–2278. SIAM, 2020. doi:10.1137/1.9781611975994.139.
6. Maria Chudnovsky and Paul D. Seymour. The three-in-a-tree problem. *Combinatorica*, 30(4):387–417, 2010. doi:10.1007/s00493-010-2334-4.
7. Víctor Dalmau, László Egri, Pavol Hell, Benoit Larose, and Arash Rafiey. Descriptive complexity of list H-coloring problems in logspace: A refined dichotomy. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 487–498. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.52.
8. László Egri, Pavol Hell, Benoit Larose, and Arash Rafiey. Space complexity of list H-colouring: a dichotomy. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 349–365. SIAM, 2014. doi:10.1137/1.9781611973402.26.
9. László Egri, Dániel Marx, and Paweł Rzążewski. Finding list homomorphisms from bounded-treewidth graphs to reflexive graphs: a complete complexity characterization. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, volume 96 of *LIPIcs*, pages 27:1–27:15. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.STACS.2018.27.
10. Thomas Emden-Weinert, Stefan Hougardy, and Bernd Kreuter. Uniquely colourable graphs and the hardness of colouring graphs of large girth. *Comb. Probab. Comput.*, 7(4):375–386, 1998. URL: http://journals.cambridge.org/action/displayAbstract?aid=46667.
11. Tomás Feder and Pavol Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B*, 72(2):236–250, 1998. doi:10.1006/jctb.1997.1812.
12. Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999. doi:10.1007/s004939970003.
13. Tomás Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory*, 42(1):61–80, 2003. doi:10.1002/jgt.10073.
14. Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms of graphs with bounded degrees. *Discrete Mathematics*, 307(3-5):386–392, 2007. doi:10.1016/j.disc.2005.09.030.
15. Tomás Feder, Pavol Hell, Sulamita Klein, and Rajeev Motwani. List partitions. *SIAM J. Discrete Math.*, 16(3):449–478, 2003. URL: http://epubs.siam.org/sam-bin/dbq/article/38405.

16 Tomás Feder, Pavol Hell, David G. Schell, and Juraj Stacho. Dichotomy for tree-structured trigraph list homomorphism problems. *Discrete Applied Mathematics*, 159(12):1217–1224, 2011. doi:10.1016/j.dam.2011.04.005.

17 Peter Gartland and Daniel Lokshtanov. Independent set on P_k-free graphs in quasi-polynomial time. *CoRR*, abs/2005.00690, 2020. To appear in FOCS 2020 Proc. arXiv:2005.00690.

18 Petr A. Golovach, Matthew Johnson, Daniël Paulusma, and Jian Song. A survey on the computational complexity of coloring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84(4):331–363, 2017. doi:10.1002/jgt.22028.

19 Petr A. Golovach, Daniël Paulusma, and Jian Song. Closing complexity gaps for coloring problems on H-free graphs. *Inf. Comput.*, 237:204–214, 2014. doi:10.1016/j.ic.2014.02.004.

20 Carla Groenland, Karolina Okrasa, Paweł Rzążewski, Alex Scott, Paul Seymour, and Sophie Spirkl. H-colouring P_t-free graphs in subexponential time. *Discrete Applied Mathematics*, 267:184–189, 2019.

21 Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990. doi:10.1016/0095-8956(90)90132-J.

22 Chính T. Hoàng, Marcin Kaminski, Vadim V. Lozin, Joe Sawada, and Xiao Shu. Deciding k-colorability of P_5-free graphs in polynomial time. *Algorithmica*, 57(1):74–81, 2010. doi:10.1007/s00453-008-9197-8.

23 Ian Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981. doi:10.1137/0210055.

24 Shenwei Huang. Improved complexity results on k-coloring P_t-free graphs. *Eur. J. Comb.*, 51:336–346, 2016. doi:10.1016/j.ejc.2015.06.005.

25 Marcin Kamiński and Anna Pstrucha. Certifying coloring algorithms for graphs without long induced paths. *Discret. Appl. Math.*, 261:258–267, 2019. doi:10.1016/j.dam.2018.09.031.

26 Gábor Kun and Mario Szegedy. A new line of attack on the dichotomy conjecture. *Eur. J. Comb.*, 52:338–367, 2016. doi:10.1016/j.ejc.2015.07.011.

27 Daniel Leven and Zvi Galil. NP-completeness of finding the chromatic index of regular graphs. *J. Algorithms*, 4(1):35–44, 1983. doi:10.1016/0196-6774(83)90032-9.

28 Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 74:1–74:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.74.

29 Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. *CoRR*, abs/2006.11155, 2020. arXiv:2006.11155.

30 Karolina Okrasa and Paweł Rzążewski. Complexity of the list homomorphism problem in hereditary graph classes. *CoRR*, abs/2010.03393, 2020. arXiv:2010.03393.

31 Karolina Okrasa and Paweł Rzążewski. Subexponential algorithms for variants of the homomorphism problem in string graphs. *J. Comput. Syst. Sci.*, 109:126–144, 2020. doi:10.1016/j.jcss.2019.12.004.

32 Marta Piecyk and Paweł Rzążewski. Fine-grained complexity of the list homomorphism problem: feedback vertex set and cutwidth. *CoRR*, abs/2009.11642, 2020. Extended abstract available in STACS 2021 Proc. arXiv:2009.11642.

33 Michał Pilipczuk, Marcin Pilipczuk, and Paweł Rzążewski. Quasi-polynomial-time algorithm for Independent Set in P_t-free graphs via shrinking the space of induced paths. In Valerie King and Hung Viet Le, editors, *Proceedings of the Fourth SIAM Symposium on Simplicity in Algorithms (SOSA), Alexandria, Virginia, USA, January 11-12, 2021 (Virtual conference)*, pages 204–209. SIAM, 2021. doi:10.1137/1.9781611976496.23.

34 Sophie Spirkl, Maria Chudnovsky, and Mingxian Zhong. Four-coloring P_6-free graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1239–1256. SIAM, 2019. doi:10.1137/1.9781611975482.76.