Counting Subgraphs in Degenerate Graphs

Lior Gishboliner* Yevgeny Levanzov† Asaf Shapira‡

August 24, 2020

Abstract

We consider the problem of counting the number of copies of a fixed graph H within an input graph G. This is one of the most well-studied algorithmic graph problems, with many theoretical and practical applications. We focus on solving this problem when the input G has *bounded degeneracy*. This is a rich family of graphs, containing all graphs without a fixed minor (e.g. planar graphs), as well as graphs generated by various random processes (e.g. preferential attachment graphs). We say that H is *easy* if there is a linear-time algorithm for counting the number of copies of H in an input G of bounded degeneracy. A seminal result of Chiba and Nishizeki from '85 states that every H on at most 4 vertices is easy. Bera, Pashanasangi, and Seshadhri recently extended this to all H on 5 vertices, and further proved that for every $k > 5$ there is a k-vertex H which is not easy. They left open the natural problem of characterizing all easy graphs H.

Bressan has recently introduced a framework for counting subgraphs in degenerate graphs, from which one can extract a sufficient condition for a graph H to be easy. Here we show that this sufficient condition is also necessary, thus fully answering the Bera–Pashanasangi–Seshadhri problem. We further resolve two closely related problems; namely characterizing the graphs that are easy with respect to counting induced copies, and with respect to counting homomorphisms. Our proofs rely on several novel approaches for proving hardness results in the context of subgraph-counting.

1 Introduction

Subgraph counting refers to the algorithmic task of computing the number of copies (i.e., occurences) of a given graph H in an input graph G. Due to its fundamental nature, this problem has been studied extensively, both from a theoretical perspective and for practical applications. In practice, subgraph counts are widely used to analyze real-world graphs, such as graphs representing telecommunication networks, biological structures and social interactions. Consequently, subgraph counts feature prominently in studies of biological [25, 37, 38] and sociological [12, 24] networks, as well as in the network science literature in general [6, 27, 34, 40, 41, 42]. For example, [34] observed that networks coming from different areas of science (such as biochemistry, neurobiology, ecology, and engineering) have significantly different counts of small subgraphs. Such frequently occuring subgraphs are called *motifs*, and, quoting [34], "may uncover the basic building blocks of most networks". Needless to say, some real-world graphs can be very large – having billions of vertices – thus making it all the more desirable to have fast subgraph counting algorithms.

*School of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. Email: liorgis1@mail.tau.ac.il. Supported by ERC Starting Grant 633509.

†School of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. Email: yevgenyl@mail.tau.ac.il. Supported by ERC Starting Grant 633509.

‡School of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. Email: asafico@tau.ac.il. Supported in part by ISF Grant 1028/16 and ERC Starting Grant 633509.

In theoretical computer science, subgraph counting and detection[1] are fundamental and widely-studied problems. Much of the research focused on counting special kinds of graphs, such as cliques [20, 26, 36, 44]; cycles [2, 18, 26]; paths and matchings (and other graphs with bounded pathwidth) [7, 8]; graphs with bounded vertex-cover number [16, 17, 29, 46]. Many of the algorithms use fast matrix multiplication [2, 20, 26, 29, 36]. For example, the best known algorithm for counting k-cliques [36] runs in time $n^{\omega k/3+O(1)}$, where $\omega < 2.373$ is the matrix multiplication constant [45]. On the negative side, k-clique counting is the canonical #W[1]-hard problem, and it is thus unlikely that there exists an algorithm which solves this problem in time $f(k) \cdot n^{o(k)}$ (for any function f). We refer the reader to [9] for further references on both theoretical and practical aspects of subgraph counting.

Subgraph counts also play a fundamental role in extremal graph theory, where subgraph densities are the basic notion used for studying sequences of dense graphs [30]. In particular, knowing approximate subgraph counts of small graphs inside a given graph G allows one to decide whether G is quasirandom [15] or, more generally, whether G consists of a bounded number of quasirandom pieces with prescribed edge densities [31].

Given the importance of the subgraph counting problem on the one hand, and its hardness in general graphs on the other, it is natural to consider special classes of graphs which admit faster counting algorithms, while also being rich enough to include many of the real-world graphs mentioned above. One prime example of such a family of graph classes is classes having bounded degeneracy. Recall that a graph G is κ-*degenerate*[2] if there is an ordering $v_1, \dots, v_n$ of the vertices of G such that v_i has at most κ neighbours in $\{v_{i+1}, \dots, v_n\}$ (for each $1 \leq i \leq n$). We say that a class of graphs has *bounded degeneracy* if there is an integer κ such that all graphs in the class are κ-degenerate. With a slight abuse of terminology, we will refer to graphs belonging to such classes as having bounded-degeneracy or being $O(1)$-degenerate.

There are many examples of well-studied graph classes having bounded degeneracy. These include all minor-closed classes (including planar graphs and, more generally, graphs embeddable into a given surface), preferential attachment graphs [3], and bounded expansion graphs [35].

The first result on subgraph counting in bounded-degeneracy graphs is probably the classical result of Chiba and Nishizeki [14], who showed that in κ-degenerate graphs, one can count r-cliques in time $O(n\kappa^{r-2})$ (for each $r \geq 3$), and 4-cycles in time $O(n\kappa)$. Bera, Pashanasangi and Seshadhri [9] recently extended this result, by showing that the H-counting problem in bounded-degeneracy graphs can be solved in time $O(n)$ for every graph H on at most 5 vertices (here, the implicit constant in the big-O notation depends on the degeneracy κ). They further showed that under a certain widely-believed hardness assumption in fine-grained complexity, the problem of counting 6-cycles cannot be solved in linear time in bounded-degeneracy graphs. (They also proved a similar result for all longer cycles, with the exception of the cycle of length 8.) The hardness assumption used asserts that detecting a triangle in a (general) graph with m edges requires significantly more than $O(m)$ time. This assumption will also serve as the foundation for our complexity-theoretic hardness results, and we state it here as follows. We refer the reader to [1], where Conjecture 1.1 was first formulated, for a detailed overview of the conjecture and its relations to many other computational problems.

Conjecture 1.1 (Triangle Detection Conjecture [1])**.** *There exists $\gamma > 0$ such that in the word RAM model of $O(\log n)$ bits, any algorithm to decide whether an input graph with n vertices and m edges is triangle-free requires $\Omega(m^{1+\gamma})$ time in expectation.*

It is believed that the constant γ in Conjecture 1.1 could be as large as 1/3. The reason for this is that

[1] The H-detection problem is the problem of deciding whether an input graph contains a copy of H.

[2] We note that degeneracy is closely related to another well-studied graph parameter, namely arboricity, which is the minimum number of forests into which the edge-set of a graph can be partitioned. It is well-known that the arboricity of a κ-degenerate graph is between $(\kappa + 1)/2$ and κ.

the best known algorithm for triangle detection [2] runs in time $O\big(\min\big(n^{\omega}, m^{2\omega/(\omega+1)}\big)\big)$, where ω is the matrix multiplication constant. If $\omega = 2$ (which would be optimal), then the running time of this algorithm is $O\big(\min\big(n^2, m^{4/3}\big)\big)$.

Having established both positive and negative results, Bera, Pashanasangi and Seshadhri [9] asked whether one can characterize all graphs H for which the H-counting problem can be solved in linear time in bounded-degeneracy graphs. Before proceeding to our resolution of this question, let us introduce some notation. Let G, H be graphs. Following [30], we denote by $\hom(H, G)$ the number of homomorphisms from H to G, by $\mathrm{inj}(H, G)$ the number of injective homomorphisms (i.e., embeddings) from H to G, and by $\mathrm{ind}(H, G)$ the number of isomorphisms from H to (induced subgraphs of) G. See Section 3 for the definition of graph homomorphism. The number of automorphisms of H is denoted $\mathrm{aut}(H)$. It is well-known (and easy to see) that the number of unlabeled copies (resp. unlabeled induced copies) of H in G is $\mathrm{inj}(H, G)/\mathrm{aut}(H)$ (resp. $\mathrm{ind}(H, G)/\mathrm{aut}(H)$).

For a graph H, we denote by HOM-CNT_H the problem of computing $\hom(H, G)$ for a given input graph G. The problems INJ-CNT_H and IND-CNT_H are defined analogously. In what follows, we say that $\text{HOM-CNT}_H/\text{INJ-CNT}_H/\text{IND-CNT}_H$ is *easy* if it can be solved in expected time $f(\kappa, H)\cdot n$ in n-vertex graphs of degeneracy κ (for some function f); and otherwise we say that it is *hard*. We will usually avoid mentioning the function f, and just speak of running time $O(n)$ (or linear time), where the implicit constant in the big-O notation may (and will) depend on κ and H. We will make no effort to optimize this dependence. The following is the main open problem raised in [9].

Problem 1.2 ([9]). *Characterize the graphs H for which INJ-CNT_H is easy.*

In this paper, we completely resolve Problem 1.2 by giving a very clean characterization of the graphs H for which INJ-CNT_H is easy. We will also solve the related problems of characterizing the graphs H for which HOM-CNT_H is easy and the graphs H for which IND-CNT_H is easy. It will turn out to be more convenient to first deal with the problem of obtaining a characterization for HOM-CNT_H. This characterization, discussed in Section 1.2, constitutes the main result of this paper. In Section 1.3 we describe how the solution of Problem 1.2 regarding INJ-CNT_H can be derived from our result regarding HOM-CNT_H. Then, in Section 1.4, we describe how a characterization for IND-CNT_H can be derived from the one for INJ-CNT_H. This approach – of relating homomorphisms to copies and copies to induced copies – was pioneered in [16] and [13], and is based on the framework developed in [30]. Finally, in Section 1.5 we briefly consider subgraph-counting in *general* (i.e., not necessarily degenerate) graphs. Using the methods developed to prove (the hardness part of) Theorem 1, we show that for a graph H, $\hom(H, \cdot)$ can be computed in linear time in general graphs if and only if H is a forest (again, the "only-if" direction assumes Conjecture 1.1).

1.1 α-acyclic hypergraphs and Bressan's algorithm

Bressan [11] provided a dynamic programming algorithm for computing $\hom(H, G)$ in $O(1)$-degenerate graphs G. A special case of Bressan's result gives a sufficient condition for HOM-CNT_H to be easy. To state this condition, we first need some additional definitions. Let us begin with the well-known notion of hypergraph α-acyclicity, which plays a key role in this paper.

Definition 1.3 (α-acyclic hypergraph). *A hypergraph F is called α-*acyclic *if there exists a tree T whose vertices are the hyperedges of F, such that the following condition is satisfied: for all $e_1, e_2, e \in E(F) = V(T)$, if e is on the unique path in T between e_1 and e_2, then $e_1 \cap e_2 \subseteq e$.*

Hypergraph α-acyclicity was introduced by Beeri, Fagin, Maier, Mendelzon, Ullman and Yannakakis [4] in the early 1980s in connection with relational database schemes, and has subsequently been widely studied. For further information, we refer the reader to [5, 10]. Note that in the special case that F is a graph, α-acyclicity is equivalent to being a forest.

Next, one associates to each directed acyclic graph $\vec{H}$ a hypergraph which captures the *reachability structure* of $\vec{H}$. For a vertex u in a directed graph $\vec{H}$, we denote by $R(u)$ the set of vertices that are *reachable* from u (namely, the set of all $v \in V(\vec{H})$ such that there is a directed path from u to v). Note that $u \in R(u)$. Recall that any directed acyclic graph (DAG) has at least one source (i.e. vertex of in-degree 0).

Definition 1.4 (α-acyclic graphs and digraphs)**.** *Let $\vec{H}$ be a DAG, and let $u_1, u_2, \ldots, u_r \in V(\vec{H})$ be the source vertices of $\vec{H}$. We associate to $\vec{H}$ a hypergraph $F_{\vec{H}}$, as follows. The vertices of $F_{\vec{H}}$ are the vertices of $\vec{H}$, and the hyperedges of $F_{\vec{H}}$ are the sets $R(u_i)$ for $1 \le i \le r$. We say that $\vec{H}$ is α-acyclic if the hypergraph $F_{\vec{H}}$ is α-acyclic. Finally, we say that an undirected graph H is α-acyclic if every acyclic orientation $\vec{H}$ of H is α-acyclic.*

Directed acyclic graphs arise naturally in the context of counting subgraphs in degenerate graphs. Indeed, many of the known counting algorithms [14, 9, 11] begin by orienting the edges of the given κ-degenerate input graph G according to some *degeneracy ordering*, thus obtaining an acyclic directed graph $\vec{G}$ in which all out-degrees are at most κ. Then the task becomes to compute $\hom(\vec{H}, \vec{G})$ for every acyclic orientation $\vec{H}$ of H. Summing these directed homomorphism counts then gives $\hom(H, G)$.

The main result[3] of [11] implies that $\hom(H, G)$ can be computed in expected time $O(n)$ whenever H is α-acyclic. Our main result, described in the following section, states that this sufficient condition is also necessary, thus giving a complete characterization of the graphs H for which HOM-CNT$_H$ is easy. Before proceeding, let us note that Bressan's algorithm [11] actually runs in time $\tilde{O}(n)$ (rather than linear), with the polylog(n) term arising from the need to search and update a dictionary with $O(n)$ entries. If, however, one allows Las Vegas randomized algorithms (as is done in [9]), then one can reduce the time for searching and updating entries to $O(1)$ by using perfect hashing [23]. This results in an algorithm which runs in expected time $O(n)$ (with randomness only appearing in the generation of the hash table, which is then followed by a deterministic algorithm).

1.2 Main result: counting homomorphisms in linear time

Our main result in this paper is as follows.

Theorem 1 (Main result)**.** *Assuming Conjecture 1.1,* HOM-CNT$_H$ *is hard whenever H is not α-acyclic.*

As mentioned above, Theorem 1 shows that the sufficient condition (for HOM-CNT$_H$ being easy) supplied by Bressan's algorithm [11] is in fact necessary. Thus we obtain the following characterization:

Corollary 2. *Assuming Conjecture 1.1,* HOM-CNT$_H$ *is easy if and only if H is α-acyclic.*

Given Theorem 1, it is natural to ask if there is a cleaner description of the α-acyclic graphs. Actually, another reason for seeking such a clean description is that in order to prove Theorem 1, it would be desirable to know that graphs that are not α-acyclic have certain easy-to-describe obstructions. Luckily (and somewhat surprisingly), we have the following concise equivalent description of the α-acyclic graphs. Throughout the paper, C_k denotes the cycle of length k. For graphs H, H_0, we say that H is *induced H_0-free* if H contains no induced copy of H_0.

[3] It is worth noting that the result of [11] is more general: it gives an algorithm that computes $\hom(H, G)$ in expected time $O(n^{\tau(H)})$, where $\tau(H)$ is a certain "treewidth-type" graph parameter which equals 1 if and only if H is α-acyclic

Theorem 3. *An undirected graph H is α-acyclic if and only if H is induced C_k-free for every $k \geq 6$.*

By combining Theorems 1 and 3, we immediately see that $\textsc{hom-cnt}_H$ is easy whenever H is induced C_k-free for all $k \geq 6$. In particular, $\textsc{hom-cnt}_H$ is easy for every chordal[4] graph H.

Let us now discuss the ideas that go into the proofs of Theorems 1 and 3, starting with the latter. Let us now discuss the proof of Theorem 3. This proof relies on a useful characterization of α-acyclic hypergraphs given in [5] (and stated here as Theorem 7). This characterization asserts that a hypergraph is α-acyclic if and only if it does not contain two certain types of *obstructions*. These two types of obstructions generalize in different ways the notion of an induced cycle from graphs to hypergraphs. The main difficulty in the proof of Theorem 3 lies in translating these "hypergraph obstructions" into "digraph obstructions", i.e. recognizing the digraph structures whose reachability hypergraphs correspond to these obstructions.

We now move on to discuss the proof of Theorem 1. With Theorem 3 at hand, it is natural to first try and show that $\textsc{hom-cnt}_{C_k}$ is hard for all $k \geq 6$. This is indeed accomplished in the following lemma.

Lemma 1.5. *Assuming Conjecture 1.1,* $\textsc{hom-cnt}_{C_k}$ *is hard for every $k \geq 6$.*

Lemma 1.5 closely resembles to the hardness result of Bera, Pashanasangi and Seshadhri [9], who proved a similar statement for the problem $\textsc{inj-cnt}_{C_k}$ (with the exception of the case $k = 8$). Still, it turns out that proving hardness results for $\textsc{hom-cnt}$ is significantly more challenging[5] than proving analogous statements for $\textsc{inj-cnt}$. While the reduction we use to prove Lemma 1.5 is similar to the one used in [9], the proof of its correctness is considerably more involved and requires several new ideas. First, it involves a subtle application of Möbius inversion (see the discussion regarding (2) in Section 1.3) to a poset consisting of a carefully chosen collection of partitions of $V(C_k)$. The reason for considering only some (and not all) of the partitions of $V(C_k)$ is that this allows us to express C_k-counts in terms of homomorphism counts of C_k and of certain trees, thus avoiding the need to compute homomorphism counts of other cyclic graphs. Second, in order to compute this Möbius inversion, we need to solve $\textsc{hom-cnt}_H$ in linear time for all trees H. This is possible due to the "if" part of Corollary 2 (using Bressan's algorithm [11]). Hence, an unexpected aspect of the proof of Lemma 1.5 is that *an algorithmic result for a problem (namely, Corollary 2 for H being a tree) is used to prove a hardness result for the same problem.*

The remaining ingredient in the proof of Theorem 1 is the following lemma, which implies that if $\textsc{hom-cnt}_H$ is easy then so is $\textsc{hom-cnt}_{H'}$ for every induced subgraph H' of H. It is easy to see that the combination of Theorem 3 and Lemmas 1.5 and 1.6 implies Theorem 1.

Lemma 1.6. *Let H be a graph. If* $\textsc{hom-cnt}_H$ *is easy, then* $\textsc{hom-cnt}_{H'}$ *is easy for every induced subgraph H' of H.*

The proof of Lemma 1.6 uses an innovative application of a powerful technique developed recently in several works on homomorphism-counting, see [16, 13]. At the heart of this technique is the observation that for (non-isomorphic) graphs $H_1, \ldots, H_k$ and non-zero constants $c_1, \ldots, c_k$, the problem of computing the linear combination $c_1 \hom(H_1, \cdot) + \cdots + c_k \hom(H_k, \cdot)$ is as hard as computing $\hom(H_i, \cdot)$ for every $1 \leq i \leq k$. The proof of this fact (appearing in [16]) uses tensor products of graphs and a result of Erdős, Lovász and Spencer [21] (stated here as Lemma A.2) regarding linear independence of homomorphism counts. For completeness, we present this proof (adapted to the setting of input graphs of bounded degeneracy and stated here as Lemma 5.2) in the appendix. The use of linear combinations of homomorphism-counts plays a crucial role in many of the reductions presented in this paper.

Our proof of Lemma 1.6 proceeds by showing that for every graph G, one can (efficiently) construct a graph G' such that $\hom(H, G')$ equals a linear combination of the homomorphism counts of all induced

[4] Recall that a graph is *chordal* if it is induced C_k-free for every $k \geq 4$.

[5] Indeed, as we note after Lemma 1.7, if $\textsc{inj-cnt}_H$ is easy then so is $\textsc{hom-cnt}_H$ (for every graph H).

subgraphs of H in G. Unlike Lemmas 1.7 and 1.8, which are similar (both in their statements and their proofs) to results obtained in [16], Lemma 1.6 is (to the best of our knowledge) a new application of the homomorphism-linear-combination technique.

In the next two sections we take advantage of known relations between the various subgraph counts hom, inj, ind in order to derive analogues of Corollary 2 for the problems INJ-CNT and IND-CNT. A similar approach was taken in [16].

1.3 From homomorphisms to copies

In this section obtain a characterization of graphs H for which INJ-CNT_H is easy, thus resolving Problem 1.2. To state this characterization, we first need to introduce the notion of a *quotient graph*. For a graph H and a partition $P = \{U_1, \dots, U_k\}$ of $V(H)$, the *quotient graph* H/P is the graph on P in which, for every $1 \leq i, j \leq k$, there is an edge[6] between U_i and U_j if and only if there are $u_i \in U_i, u_j \in U_j$ such that $\{u_i, u_j\} \in E(H)$. The following is our main result for INJ-CNT_H.

Theorem 4. *Let $\mathcal{H}$ be the family of all graphs H such that every quotient graph of H is induced C_k-free for all $k \geq 6$. If $H \in \mathcal{H}$ then INJ-CNT_H is easy. Conversely, assuming Conjecture 1.1, if $H \notin \mathcal{H}$ then INJ-CNT_H is hard.*

To illustrate an application of Theorem 4, let us observe that every split graph belongs to the graph-family $\mathcal{H}$. Recall that a graph is *split* if its vertex-set can be partitioned into two parts: one spanning a clique and the other an independent set. It is easy to see that every quotient graph of a split graph is itself split, and that C_k is not split for any $k \geq 6$ (in fact, this is also true for $k = 4, 5$). Thus, Theorem 4 implies that INJ-CNT_H is easy to every split graph H. At the end of Section 6, we discuss in further detail the graph-family $\mathcal{H}$ appearing in Theorem 4. For now, let us note that this graph-family is hereditary (i.e., closed under taking induced subgraphs), and that among the forbidden induced subgraphs for $\mathcal{H}$ are the 6-edge path and 6-edge matching.

The reason for the appearance of quotient graphs in Theorem 4 is that they can be used to relate homomorphism counts to injective homomorphism counts [30, Section 5.2.3]. Indeed, it is well-known and easy to see (for a proof, see Fact 3.2), that

$$\hom(H, G) = \sum_{P} \text{inj}(H/P, G), \tag{1}$$

where P runs over all partitions of $V(H)$. Equation (1) can be thought of as a relation over the poset of all partitions of $V(H)$. As is well-known [30, Section 5.2.3], one can invert this relation using Möbius inversion, thus obtaining the following:

$$\text{inj}(H, G) = \sum_{P} \mu_{\text{part}}(P) \cdot \hom(H/P, G). \tag{2}$$

Here, μ_{part} is the Möbius function of the partition poset. For more details, see Section 3.

Equation (2) expresses $\text{inj}(H, G)$ as a linear combination of $\hom(H/P, G)$ (where P runs over all partitions of $V(H)$). As mentioned above, this means that computing $\text{inj}(H, G)$ is exactly as hard as computing $\hom(H/P, G)$ for all P. Thus we have the following:

[6] Note that if some U_i is not an independent set in H, then the vertex U_i has a loop in H/P. Such partitions P can be safely ignored in all of our arguments, since our input graphs G are always assumed to be simple, and hence $\hom(H/P, G) = 0$ if H/P has a loop.

Lemma 1.7. *Let H be a graph. Then* INJ-CNT$_H$ *is easy if and only if* HOM-CNT$_{H/P}$ *is easy for every partition P of $V(H)$.*

A similar result has appeared in [16]. It is easy to see that Lemma 1.7 and Theorem 1 together imply Theorem 4. A corollary of Lemma 1.7 is that INJ-CNT$_H$ is at least as hard as HOM-CNT$_H$.

1.4 From copies to induced copies

In this subsection we apply the results of the previous two subsections in order to obtain a characterization of the graphs H for which IND-CNT$_H$ is easy. Here and throughout the paper, a *supergraph* of H is any graph on $V(H)$ of which H is a subgraph. The following is our main result for IND-CNT$_H$.

Theorem 5. *Let $\mathcal{H}^*$ be the family of all graphs H such that H does not contain as an induced subgraph any (not necessarily induced) spanning subgraph of C_6. If $H \in \mathcal{H}^*$ then* IND-CNT$_H$ *is easy. Conversely, assuming Conjecture 1.1, if $H \notin \mathcal{H}^*$ then H is hard.*

We note that the graph-family $\mathcal{H}^*$ is quite rich. For example, it contains all complements of triangle-free graphs (i.e. graphs having no independent set of size 3). On the negative side, graphs $H \in \mathcal{H}^*$ must be free (among other things) of independent sets of size 6, induced paths of length 5 and induced matchings of size 3.

Theorem 5 is derived from a result analogous to Lemma 1.7, this time relating the problems IND-CNT and INJ-CNT. Again, this relation has been previously used in [16]. It is well-known and easy to see that the following holds for every pair of graphs G, H.

$$\mathrm{inj}(H,G) = \sum_{E \subseteq \binom{V(H)}{2} \setminus E(H)} \mathrm{ind}(H \cup E, G). \tag{3}$$

Here, $H \cup E$ is the graph obtained from H by adding to it all edges in E. Hence, $H \cup E$ runs over all supergraphs of H. The equation (3) can be thought of as a relation over the boolean poset of all subsets of $\binom{V(H)}{2} \setminus E(H)$. Just like (1), it is well-known that this relation can be inverted using Möbius inversion (which in this case boils down to the inclusion-exclusion principle, see [30, Section 5.2.3]). The resulting inverted relation is:

$$\mathrm{ind}(H,G) = \sum_{E \subseteq \binom{V(H)}{2} \setminus E(H)} (-1)^{|E|} \cdot \mathrm{inj}(H \cup E, G). \tag{4}$$

Equation (4) shows that computing $\mathrm{inj}(H', G)$ for every supergraph H' of H is sufficient for computing $\mathrm{ind}(H, G)$. The following lemma states that it is also necessary. It also gives a (seemingly weaker but in fact equivalent) necessary and sufficient condition in terms of computing $\mathrm{hom}(H', G)$ for every supergraph H' of H.

Lemma 1.8. *For every graph H, the following are equivalent.*

1. IND-CNT$_H$ *is easy.*
2. INJ-CNT$_{H'}$ *is easy for every supergraph H' of H.*
3. HOM-CNT$_{H'}$ *is easy for every supergraph H' of H.*

Given the analogy between (2) and 4, one could hope for a reduction between IND-CNT and INJ-CNT that does not go through HOM-CNT. We are not aware of such a reduction, however. Instead, we again exploit the homomorphism-linear-combination framework in order to reduce HOM-CNT$_{H'}$ (for all supergraphs H' of H) to IND-CNT$_H$. We then complete the picture by proving that solving HOM-CNT$_{H'}$ for all supergraphs H' of H allows one to solve INJ-CNT$_{H'}$ for all such H', which in turn allows one to solve IND-CNT$_H$ using (4). This step (i.e. proving the implication $3 \Rightarrow 2$) requires Lemma 1.6.

A corollary of Lemma 1.8 is that IND-CNT$_H$ is at least as hard as INJ-CNT$_H$ (which itself is at least as hard as HOM-CNT$_H$). At the end of Section 7, we give the simple derivation of Theorem 5 from Lemma 1.8 and Theorem 1.

We conclude by noting that the reductions used to prove Lemmas 1.6, 1.7 and 1.8 are more robust than is stated in those lemmas. Namely, these reductions pertain not only to algorithms running in time $O(n)$, but to larger running times as well. See Lemmas 5.1, 6.1 and 7.1 for the general statements. Moreover, all of our hardness results (including Theorem 1) actually give a lower bound of $\Omega(n^{1+\gamma})$ (and not just a superlinear bound) on the expected time necessary to solve the respective problems. This is because $\Omega(m^{1+\gamma})$ is the lower bound asserted by Conjecture 1.1. So we see that if $\gamma = 1/3$ (as is believed), then our hardness results would actually give a lower bound of $\Omega(n^{4/3})$ on the expected running time (see the remark following Conjecture 1.1).

1.5 Counting homomorphisms in linear time in general graphs

In this section we obtain a characterization of the graphs H such that $\hom(H,\cdot)$ can be computed in linear time in *general* (i.e., not necessarily degenerate) input graphs. Dalmau and Jonsson [19] have shown that the complexity of counting H-homomorphisms (in general graphs) is essentially controled by the tree-width $\mathrm{tw}(H)$ of H: it had been previously shown (see [22, Proposition 7]) that $\hom(H,\cdot)$ can be computed in time $O(n^{\mathrm{tw}(H)+1})$ in n-vertex graphs; and conversely, Dalmau and Jonsson [19] have shown that there is a function $f : \mathbb{N} \to \mathbb{N}$ with $f(t) \to \infty$ (as $t \to \infty$), such that $\hom(H,\cdot)$ cannot be computed in time $O(n^{f(\mathrm{tw}(H))})$ in n-vertex graphs (under the assumption that FPT does not equal #W[1]). Here we obtain a sharper[7] result for the special case of linear runtime; we show that $\hom(H,\cdot)$ can be computed in linear time *if and only if* $\mathrm{tw}(H) = 1$, namely H is a forest.

Theorem 6. *Assuming Conjecture 1.1,* $\hom(H,G)$ *can be computed in time* $O(|V(G)| + |E(G)|)$ *if and only if* H *is a forest.*

Paper organization The rest of the paper is organized as follows. Section 2 contains the proof of Theorem 3. In Section 3 we survey some definitions and known results which will be used in subsequent sections. Most of Section 3 is based on [30, Chapter 5]. Section 4 is devoted to proving Lemma 1.5. The proofs of Lemmas 1.6, 1.7 and 1.8 appear in Sections 5, 6 and 7, respectively. Section 7 also contains the proof of Theorem 5. Finally, in Section 8 we prove Theorem 6.

2 A Characterization of α-acyclic Graphs: Proof of Theorem 3

The key step in the proof of Theorem 3 is the following lemma. Recall that for a vertex u in a digraph, $R(u)$ denotes the set of all vertices reachable from u.

Lemma 2.1. *Let $\vec{H}$ be an acyclic orientation of an undirected graph H. Let $k \geq 3$, and assume that there exist distinct vertices $u_0, \dots, u_{k-1}, x_0, \dots, x_{k-1} \in V(\vec{H})$ such that for all $0 \leq i \leq k-1$, we have that*

[7] Comparing Theorem 6 to the result of [19], we note that a special case of the latter also uses a reduction from the triangle-counting problem; it shows that, assuming Conjecture 1.1, $\hom(H,\cdot)$ cannot be computed in time $O(|V(G)|)$ whenever H contains the 3×3 grid as a minor. Evidently, this condition does not capture all non-forest graphs H (cf. Theorem 6).

$x_i \in R(u_{i-1}) \cap R(u_i)$ and $x_i \notin R(u_j)$ for all $j \neq i, i-1$ (with indices taken modulo k). Then, H contains an induced copy of a cycle C_ℓ for some $\ell \geq 6$.

Proof. We say that a $2k$-tuple $(v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1})$ of distinct vertices of $\vec{H}$ is *good* if for every $0 \leq i \leq k-1$, $y_i \in R(v_i) \cap R(v_{i-1})$ and $y_i \notin R(v_j)$ for all $j \neq i, i-1$ (with indices taken modulo k). In other words, $(v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1})$ is good if and only if for every $0 \leq i \leq k-1$, there are directed paths from v_i to y_i and y_{i+1}, and there is no directed path from v_i to y_j for any $j \neq i, i+1$. By assumption, the tuple $(u_0, \ldots, u_{k-1}, x_0, \ldots, x_{k-1})$ is good, implying that the set of good $2k$-tuples is non-empty. Observe that for a good $2k$-tuple $(v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1})$ and $0 \leq i \leq k-1$, it holds that $R(v_i) \cap R(v_{i-1}) \cap \{y_0, \ldots, y_{k-1}\} = \{y_i\}$ (here we use the assumption that $k \geq 3$). This implies that $R(v_i) \cap R(v_{i-1}) \cap \{v_0, \ldots, v_{k-1}\} = \emptyset$, because if $v_j \in R(v_i) \cap R(v_{i-1})$ (for some $0 \leq j \leq k-1$), then $y_j, y_{j+1} \in R(v_j) \subseteq R(v_i) \cap R(v_{i-1})$, which we just ruled out. Similarly, the definition of a good $2k$-tuple implies that if $y_i, y_{i+1} \in R(v_j)$ (for some $0 \leq i, j \leq k-1$), then $j = i$ (again, we are using here the assumption that $k \geq 3$). This implies that there are no $0 \leq i, j \leq k-1$ such that $y_i, y_{i-1} \in R(y_j)$, since otherwise we would have $y_i, y_{i-1} \in R(y_j) \subseteq R(v_j) \cap R(v_{j-1})$, which is impossible since we cannot have both $j = i$ and $j - 1 = i$. We have thus established the following fact, which will be used several times.

Fact 2.2. *Let $(v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1})$ be a good tuple, let $z \in \{v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1}\}$ and let $0 \leq i \leq k-1$. If $z \in R(v_i) \cap R(v_{i-1})$ then $z = y_i$, and if $y_i, y_{i+1} \in R(z)$ then $z = v_i$.*

For vertices $a, b \in V(H) = V(\vec{H})$, denote by $\overrightarrow{\text{dist}}(a, b)$ the length of a shortest directed path from a to b (in $\vec{H}$). Now, fix a good $2k$-tuple $M = (v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1})$ which minimizes the sum

$$\sum_{i=0}^{k-1} \left(\overrightarrow{\text{dist}}(v_i, y_i) + \overrightarrow{\text{dist}}(v_{i-1}, y_i) \right). \tag{5}$$

Let us now fix specific shortest (directed) paths $P(v_i, y_j)$ from v_i to y_j for $0 \leq i \leq k-1$ and $j = i, i+1$ (as always, indices are taken modulo k). We will denote by $P\{v_i, y_j\}$ the underlying undirected path of $P(v_i, y_j)$. Let C be the (undirected) closed walk obtained by concatenating the paths

$$P\{y_0, v_0\}, P\{v_0, y_1\}, P\{y_1, v_1\}, \ldots, P\{v_{k-2}, y_{k-1}\}, P\{y_{k-1}, v_{k-1}\}, P\{v_{k-1}, y_0\}. \tag{6}$$

We now show that C is a simple cycle. We will then show that C is induced. While the proof idea is rather simple, the details are somewhat lengthy.

Two paths appearing consecutively (in a cyclic manner) in (6) will be called consecutive. In other words, the pairs of consecutive paths are $(P\{y_i, v_i\}, P\{v_i, y_{i+1}\})$ and $(P\{v_i, y_{i+1}\}, P\{y_{i+1}, v_{i+1}\})$ (for $0 \leq i \leq k-1$). If, by contradiction, C is not a simple cycle, then either there is a pair of non-consecutive paths which intersect, or a pair of consecutive paths which intersect outside of the endpoint they share. We now rule out each of these possibilities.

Case 1: We consider the intersection of $P(v_i, y_i)$ and $P(v_i, y_{i+1})$ for some $0 \leq i \leq k-1$. Assume that there exists a vertex $z \neq v_i$ such that $z \in P(v_i, y_i) \cap P(v_i, y_{i+1})$. Then $y_i, y_{i+1} \in R(z)$. By Fact 2.2, we have $z \notin \{v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1}\}$. Now, replacing v_i with z in the tuple M, we get a new $2k$-tuple of distinct vertices which is also good. Indeed, there are paths from z to y_i, y_{i+1}, and there is no path from z to y_j for all $j \neq i, i+1$, as otherwise we would have a path from v_i to y_j (via z), contradicting the assumption. Since $z \neq v_i$, the two new paths we get by replacing v_i with z are strictly shorter than the two original ones, which contradicts the minimality of (5).

Case 2: We consider the intersection of $P(v_{i-1}, y_i)$ and $P(v_i, y_i)$ for some $0 \leq i \leq k-1$. Assume that there exists a vertex $z \neq y_i$ such that $z \in P(v_{i-1}, y_i) \cap P(v_i, y_i)$. Then $z \in R(v_i) \cap R(v_{i-1})$. By Fact 2.2,

$z \notin \{v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1}\}$. Now, replacing y_i with z in the tuple M, we get a new $2k$-tuple of distinct vertices which is also good. Indeed, there are paths from v_{i-1}, v_i to z, and there is no path from v_j to z for any $j \neq i, i-1$, as otherwise we would have a path from v_j to y_i (via z), which contradicts the assumption. Since $z \neq y_i$, the two new paths we get by replacing y_i with z are strictly shorter than the two original ones, which contradicts the minimality of (5).

Case 3: We consider the intersection of $P(v_i, y_s)$ and $P(v_j, y_t)$ for some $i \neq j$ and $s \neq t$ (where $s \in \{i, i+1\}$ and $t \in \{j, j+1\}$). Assume that there exists a vertex z such that $z \in P(v_i, y_s) \cap P(v_j, y_t)$. In this case, as there are paths from v_i and v_j to y_s, y_t (via z), we must have $\{s, t\} = \{i, i+1\} = \{j, j+1\}$, which implies that $i = j$ (as $k \geq 3$), in contradiction to the assumption that $i \neq j$.

From Cases 1-3 it follows that the closed walk C is indeed a simple cycle in H. We now prove that C is an induced cycle. We first observe that for a path $P(v_i, y_j)$ with $j \in i, i+1$ and two vertices $z_1, z_2 \in P(v_i, y_j)$ such that z_2 comes after z_1 along the path, we cannot have the edge (z_2, z_1) as $\vec{H}$ is acyclic. In addition, we cannot have the edge (z_1, z_2), unless it is an edge of the path, as this would contradict the fact that $P(v_i, y_j)$ is a shortest path from v_i to y_j. Thus, the (undirected) path $P\{v_i, y_j\}$ is induced. We conclude that if C has a chord, then it must connect two distinct paths among the paths in (6). We now rule out the existence of such a chord by analyzing several cases.

Case 1: Consider an edge between the two paths $P(v_i, y_i)$ and $P(v_i, y_{i+1})$ for some $0 \leq i \leq k-1$. We assume, without loss of generality, that there is an edge $(z_1, z_2) \in E(\vec{H})$ such that $z_1 \in P(v_i, y_i)$, $z_2 \in P(v_i, y_{i+1})$, and $z_1, z_2 \neq v_i$. Then $y_i \in R(z_1)$ and $y_{i+1} \in R(z_2) \subseteq R(z_2)$. By Fact 2.2, $z_1 \notin \{v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1}\}$. Now, replacing v_i with z_1 in the tuple M, we get a new $2k$-tuple of distinct vertices which is also good. Indeed, there are paths from z_1 to y_i, y_{i+1}, and there is no path from z_1 to y_j for all $j \neq i, i+1$, as otherwise we would have a path from v_i to y_j (via z_1), which is impossible. Since $z_1, z_2 \neq v_i$, we have $\overrightarrow{\mathrm{dist}}(z_1, y_i) + \overrightarrow{\mathrm{dist}}(z_1, y_{i+1}) < \overrightarrow{\mathrm{dist}}(v_i, y_i) + \overrightarrow{\mathrm{dist}}(v_i, y_{i+1})$, which contradicts the minimality of (5).

Case 2: Consider an edge between the two paths $P(v_{i-1}, y_i)$ and $P(v_i, y_i)$ for some $0 \leq i \leq k-1$. We assume, without loss of generality, that there is an edge $(z_1, z_2) \in E(\vec{H})$ such that $z_1 \in P(v_{i-1}, y_i)$, $z_2 \in P(v_i, y_i)$, and $z_1, z_2 \neq y_i$. Then $z_2 \in R(v_i) \cap R(z_1) \subseteq R(v_i) \cap R(v_{i-1})$. By Fact 2.2, $z_2 \notin \{v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1}\}$. Now, replacing y_i with z_2 in the tuple M, we get a new $2k$-tuple of distinct vertices which is also good. Indeed, there are paths from v_{i-1}, v_i to z_2, and there is no path from v_j to z_2 for all $j \neq i, i-1$, as otherwise we would have a path from v_j to y_i (via z_2), which is impossible. Since $z_1, z_2 \neq y_i$, we have $\overrightarrow{\mathrm{dist}}(v_i, z_2) + \overrightarrow{\mathrm{dist}}(v_{i-1}, z_2) < \overrightarrow{\mathrm{dist}}(v_i, y_i) + \overrightarrow{\mathrm{dist}}(v_{i-1}, y_i)$, which contradicts the minimality of (5).

Case 3: Consider an edge between the two paths $P(v_i, y_s)$ and $P(v_j, y_t)$ for some $0 \leq i \neq j \leq k-1$ and $s \neq t$ (where $s \in \{i, i+1\}$ and $t \in \{j, j+1\}$). We assume, without loss of generality, that there is an edge $(z_1, z_2) \in E(\vec{H})$ such that $z_1 \in P(v_i, y_s)$ and $z_2 \in P(v_j, y_t)$. In this case, as there is a path from v_i to y_t (via z_1, z_2), we must have $\{s, t\} = \{i, i+1\}$. In addition, as $t \in \{j, j+1\}$, we either have that $s = i, t = i+1$, which implies $t = j$ (as $i \neq j$); or that $s = i+1, t = i$, which implies $t = j+1$ (as $i \neq j$).

We first consider the case when $s = i, t = i+1$ and $j = i+1$. In other words, we are considering the situation where there is an edge $(z_1, z_2) \in E(\vec{H})$ from $z_1 \in P(v_i, y_i)$ to $z_2 \in P(v_{i+1}, y_{i+1})$. Note that $y_i \in R(z_1)$ and $y_{i+1} \in R(z_2) \subseteq R(z_1)$. By Fact 3.2, either $z_1 = v_i$ or $z_1 \notin \{v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1}\}$. Similarly, note that $z_2 \in R(z_1) \cap R(v_{i+1}) \subseteq R(v_i) \cap R(v_{i+1})$, so by Fact 3.2 either $z_2 = y_{i+1}$ or $z_2 \notin \{v_0, \ldots, v_{k-1}, y_0, \ldots, y_{k-1}\}$. Furthermore, we cannot have both $z_1 = v_i$ and $z_2 = y_{i+1}$, because then z_1, z_2 are both contained in the path $P(v_i, y_{i+1})$, and we have already ruled out the possibility of such a chord.

Now, replacing v_i with z_1 and y_{i+1} with z_2 in the tuple M, we get a new $2k$-tuple of distinct vertices which is also good. Indeed, there are paths from z_1 to y_i, z_2, and there is no path from z_1 to y_j for all $j \neq i, i+1$, as otherwise we would have a path from v_i to y_j (via z_1), which contradicts the assumption.

Similarly, there are paths from z_1, v_{i+1} to z_2, and there is no path from v_j to z_2 for all $j \neq i, i+1$, as otherwise we would have a path from v_j to y_{i+1} (via z_2), which contradicts the assumption. Since either $z_1 \neq v_i$ or $z_2 \neq y_{i+1}$, we have $\overrightarrow{\text{dist}}(z_1, y_i) + \overrightarrow{\text{dist}}(z_1, z_2) + \overrightarrow{\text{dist}}(v_{i+1}, z_2) < \overrightarrow{\text{dist}}(v_i, y_i) + \overrightarrow{\text{dist}}(v_i, y_{i+1}) + \overrightarrow{\text{dist}}(v_{i+1}, y_{i+1})$, which contradicts the minimality of (5).

We now consider the (symmetrical) case when $s = i + 1, t = i$ and $j = i - 1$. In other words, we are considering the situation where there is an edge $(z_1, z_2) \in E(\vec{H})$ from $z_1 \in P(v_i, y_{i+1})$ to $z_2 \in P(v_{i-1}, y_i)$. Note that $y_{i+1} \in R(z_1)$ and $y_i \in R(z_2) \subseteq R(z_1)$. By Fact 3.2, either $z_1 = v_i$ or $z_1 \notin \{v_0, \dots, v_{k-1}, y_0, \dots, y_{k-1}\}$. Similarly, note that $z_2 \in R(z_1) \cap R(v_{i-1}) \subseteq R(v_i) \cap R(v_{i-1})$, so by Fact 3.2 either $z_2 = y_i$ or $z_2 \notin \{v_0, \dots, v_{k-1}, y_0, \dots, y_{k-1}\}$. Furthermore, we cannot have both $z_1 = v_i$ and $z_2 = y_i$, because then z_1, z_2 are both contained in the path $P(v_i, y_i)$, and we have already ruled out the possibility of such a chord.

Now, replacing v_i with z_1 and y_i with z_2 in the tuple M, we get a new $2k$-tuple of distinct vertices which is also good. Indeed, there are paths from z_1 to z_2, y_{i+1}, and there is no path from z_1 to y_j for all $j \neq i, i+1$, as otherwise we would have a path from v_i to y_j (via z_1), which contradicts the assumption. Similarly, there are paths from z_1, v_{i-1} to z_2, and there is no path from v_j to z_2 for all $j \neq i, i-1$, as otherwise we would have a path from v_j to y_i (via z_2), which contradicts the assumption. Since either $z_1 \neq v_i$ or $z_2 \neq y_i$, we have $\overrightarrow{\text{dist}}(z_1, y_{i+1}) + \overrightarrow{\text{dist}}(z_1, z_2) + \overrightarrow{\text{dist}}(v_{i-1}, z_2) < \overrightarrow{\text{dist}}(v_i, y_{i+1}) + \overrightarrow{\text{dist}}(v_i, y_i) + \overrightarrow{\text{dist}}(v_{i-1}, y_i)$, which contradicts the minimality of (5).

Cases 1-3 imply that C is an induced cycle. The length of C is evidently at least $2k \geq 6$. This completes the proof of the lemma. ■

To prove Theorem 3, we combine Lemma 2.1 with the following theorem, which gives a structural characterization of α-acyclic hypergraphs. The proof of this theorem can be found in [4, 5].

Theorem 7. *A hypergraph F is α-acyclic if and only if for every $k \geq 3$, there is* no *$S = \{x_0, x_1, \dots, x_{k-1}\} \subseteq V(F)$ such that one of the following conditions holds:*

1. *For every $0 \leq i \leq k-1$ there exists $e \in E(F)$ such that $e \cap S = \{x_i, x_{i+1}\}$, and there is no $e \in E(F)$ with $|e \cap S| \geq 2$ such that $e \cap S \neq \{x_i, x_{i+1}\}$ for all $0 \leq i \leq k-1$. (All indices are taken modulo k.)*

2. *For every $0 \leq i \leq k-1$ there exists $e \in E(F)$ such that $e \cap S = S \setminus \{x_i\}$, and there is no $e \in E(F)$ such that $S \subseteq e$.*

We are now ready to prove Theorem 3.

Proof of Theorem 3. We first prove the "only if" part of the theorem, or, more precisely, the contrapositive of this statement. Suppose that H contains an induced copy of C_ℓ for some $\ell \geq 6$, and let $C = (c_0, c_1, \dots, c_{\ell-1})$ be such a cycle. We now orient the edges of H as follows. The edges of C are oriented alternatingly along the cycle. More precisely, for an even $0 \leq i \leq \ell - 1$, we orient the edge $\{c_i, c_{i+1}\} \in E(H)$ from c_i to c_{i+1}, and for an odd $0 \leq i \leq \ell - 1$, we orient the edge $\{c_i, c_{i+1}\} \in E(H)$ from c_{i+1} to c_i (with indices taken modulo ℓ). Now, orient all edges between C and $V(H) \setminus C$ from C to $V(H) \setminus C$, and orient all edges in $V(H) \setminus C$ arbitrarily while avoiding the creation of a directed cycle. We denote the resulting orientation of H by $\vec{H}$. One can easily verify that $\vec{H}$ is acyclic. Indeed, by our construction, the induced subgraphs $\vec{H}[C]$ and $\vec{H}[V(H) \setminus C]$ do not contain directed cycles, and all the edges between C and $V(H) \setminus C$ go in the same direction. Next, we observe that for an even ℓ, all vertices of C with an even index are source vertices in $\vec{H}$, while for an odd ℓ, all vertices of C with an even index apart from c_0 are source vertices in $\vec{H}$. Let $S = \{c_j \,|\, j \text{ is odd}\}$, observing that $|S| \geq \lfloor \ell/2 \rfloor \geq 3$. We also observe that for an even i we have that $R(c_i) \cap S = \{c_{i-1}, c_{i+1}\}$, and for every source vertex u of $\vec{H}$, which is not in C we have that $R(u) \cap S = \emptyset$. Therefore, applying Theorem 7 with respect to S, we get that $\vec{H}$ is not α-acyclic. Thus, H has an acyclic orientation which is not α-acyclic, as required.

We now establish the "if" part of the theorem. Again, we will prove the contrapositive. Suppose that there exists an acyclic orientation $\vec{H}$ of H which is not α-acyclic. Let $F_{\vec{H}}$ be the hypergraph as in Definition 1.4. Then $F_{\vec{H}}$ is not α-acyclic. Hence, by Theorem 7, there exists $S = \{x_0, x_1, \dots, x_{k-1}\} \subseteq V(F_{\vec{H}})$ with $k \geq 3$ such that (at least) one of the conditions 1-2 in that theorem holds with respect to S. Our goal is to show that this implies the existence of an induced ℓ-cycle in H for some $\ell \geq 6$.

Assume first that S satisfies Condition 1 of Theorem 7. For each $0 \leq i \leq k-1$, let $e_i \in E(F_{\vec{H}})$ be such that $e_i \cap S = \{x_i, x_{i+1}\}$. By the definition of $F_{\vec{H}}$, for each $0 \leq i \leq k-1$ there is a source u_i of $\vec{H}$ such that $e_i = R(u_i)$. So for every $0 \leq i \leq k-1$ we have that $x_i \in R(u_{i-1}) \cap R(u_i)$ and $x_i \notin R(u_j)$ for all $j \neq i, i-1$. Moreover, $u_0, \dots, u_{k-1}, x_0, \dots, x_{k-1}$ are pairwise-distinct because $x_0, \dots, x_{k-1}$ are pairwise-distinct, $u_0, \dots, u_{k-1}$ are pairwise-distinct, and $u_0, \dots, u_{k-1}$ are sources of $\vec{H}$ while $x_0, \dots, x_{k-1}$ are not. Therefore, we can apply Lemma 2.1 and get that H contains an induced copy of C_ℓ for some $\ell \geq 6$, as required.

Now assume that S satisfies Condition 2 of Theorem 7. For each $0 \leq i \leq k-1$, let $e_i \in E(F_{\vec{H}})$ be such that $e_i \cap S = S \setminus \{x_i\}$. By the definition of $F_{\vec{H}}$, for each $0 \leq i \leq k-1$ there is a source u_i of $\vec{H}$ such that $e_i = R(u_i)$. Considering only $R(u_0), R(u_1), R(u_2)$, and setting $y_0 := x_1, y_1 := x_2, y_2 := x_0$, we have that $y_0 \in R(u_2) \cap R(u_0)$ but $y_0 \notin R(u_1)$, $y_1 \in R(u_0) \cap R(u_1)$ but $y_1 \notin R(u_2)$, and $y_2 \in R(u_1) \cap R(u_2)$ but $y_2 \notin R(u_0)$. Furthermore, just like in the previous case, $u_0, u_1, u_2, y_0, y_1, y_2$ are pairwise-distinct. Therefore, we can again apply Lemma 2.1 and get that H contains an induced copy of C_ℓ for some $\ell \geq 6$, as required. ■

3 Preliminaries: Homomorphism Counts and Möbius Inversion

For a graph F, we will denote by $v(F)$ and $e(F)$ the number of vertices and edges of F, respectively. Recall that for graphs G, H, a *homomorphism* from H to G is a function $\varphi : V(H) \to V(G)$ such that $\{\varphi(u), \varphi(v)\} \in E(G)$ for every $\{u, v\} \in E(H)$.

It will sometimes be convenient to consider the empty graph K_0. For every graph G, we define $\hom(K_0, G) = 1$. If G is non-empty, then we also define $\hom(G, K_0) = 0$.

We now survey some basic properties of partitions and quotient graphs. For a graph H, we denote by $\mathcal{P}(H)$ the set of all partitions of $V(H)$. Recall that a partition P is a *refinement* of a partition Q if every part of P is contained in some part of Q. Note that for every partition $P \in \mathcal{P}(H)$, there is a one-to-one correspondence between partitions of $V(H/P)$, on the one hand, and partitions $Q \in \mathcal{P}(H)$ such that P refines Q, on the other. Given $P, Q \in \mathcal{P}(H)$ such that P refines Q, we denote by Q/P the partition of $V(H/P)$ corresponding to Q. Namely, $Q/P = \{\{V \in P : V \subseteq U\} : U \in Q\}$. We will need the following simple observation.

Observation 3.1. *Let H be a graph and let P, Q be partitions of $V(H)$ such that P refines Q. Then $H/Q \cong (H/P)/(Q/P)$.*

Proof. It is easy to check that the map $U \mapsto \{V \in P : V \subseteq U\}$ (for $U \in Q$) from Q to Q/P is an isomorphism from H/Q to $(H/P)/(Q/P)$. ■

The family of all partitions of some fixed ground set admits a natural order relation: that of refinement. For partitions P, Q, we will write $P \leq Q$ to mean that P refines Q. The following simple fact generalizes (1). For completeness, we include a proof.

Fact 3.2. *Let H, G be graphs, and let P be a partition of $V(H)$. Then*

$$\hom(H/P, G) = \sum_{Q \in \mathcal{P}(H)\colon Q \geq P} \mathrm{inj}(H/Q, G).$$

Proof. Let P_0 denote the partition of $V(H)$ into singletons; so $H/P_0 \cong H$. Note that for $P = P_0$, the statement of Fact 3.2 reduces to (1). So let us start by proving (1). Let $\varphi : V(H) \to V(G)$ be a homomorphism, and write $\mathrm{Im}\varphi = \{x_1, \dots, x_k\}$. For each $1 \leq i \leq k$, put $U_i := \varphi^{-1}(x_i)$. Then $P := \{U_1, \dots, U_k\}$ is a partition of $V(H)$. Moreover, it is easy to check that the map $U_i \mapsto x_i$ ($1 \leq i \leq k$) is an injective homomorphism from H/P to G. So we see that each homomorphism from H to G corresponds to an injective homomorphism from some quotient graph of H to G. It is easy to verify that this correspondence is one-to-one. This establishes (1). Now, the general statement of Fact 3.2 follows from the combination of (1) and Observation 3.1:

$$\begin{aligned}
\hom(H/P, G) &= \sum_{Q' \in \mathcal{P}(H/P)} \mathrm{inj}((H/P)/Q', G) \\
&= \sum_{Q \in \mathcal{P}(H)\colon Q \geq P} \mathrm{inj}((H/P)/(Q/P), G) \\
&= \sum_{Q \in \mathcal{P}(H)\colon Q \geq P} \mathrm{inj}(H/Q, G).
\end{aligned}$$

■

We will need the general Möbius Inversion Theorem for posets (see [43]), which we state as follows.

Theorem 8 (Möbius Inversion Theorem)**.** *Let $(\mathcal{P}, \leq)$ be a poset. Then there exists a (unique) function $\mu : \mathcal{P}^2 \to \mathbb{Z}$ such that for every pair of functions $f, g : \mathcal{P} \to \mathbb{Z}$, the following are equivalent:*

1. *$g(P) = \sum_{Q \geq P} f(Q)$ for every $P \in \mathcal{P}$.*
2. *$f(P) = \sum_{Q \geq P} \mu(P, Q) g(Q)$ for every $P \in \mathcal{P}$.*

Furthermore, the function μ satisfies $\mu(P, P) = 1$ for each $P \in \mathcal{P}$, and

$$\sum_{P \leq R \leq Q} \mu(P, R) = 0 \tag{7}$$

for each pair $(P, Q) \in \mathcal{P}^2$ with $P < Q$.

Note that the the Möbius function μ of a poset can be computed effectively by using the recursive relation (7) or, equivalently, inverting the adjacency matrix of the poset. In any case, the posets we consider will be of constant size, depending only on the graph H.

In what follows, we will need to know the Möbius function of the partition poset[8]. Let H be a graph and let P_0 be the partition of $V(H)$ into singletons. For each $P \in \mathcal{P}(H)$, set $\mu_{\mathrm{part}}(P) := \mu(P_0, P)$. Note that μ_{part} is exactly the Möbius function appearing in (2). The value of μ_{part} is determined by the Frucht–Rota-–Schützenberger formula (see, e.g., [43, Chapter 25]), which states that

$$\mu_{\mathrm{part}}(P) = \mu(P_0, P) = (-1)^{v(H) - |P|} \cdot \prod_{U \in P} (|U| - 1)!. \tag{8}$$

[8]We will actually only need to know the sign of this Möbius function; the precise formula (8) is stated solely for completeness.

4 Hardness Results for HOM-CNT$_{C_k}$: Proof of Lemma 1.5

In this section we prove Lemma 1.5. Our reduction gives the following result:

Lemma 4.1. *For every $k \geq 4$ and for every graph F, one can construct in time $O(|V(F)|+|E(F)|)$ graphs G, G' such that $|V(G)|, |E(G)|, |V(G')|, |E(G')| = O(|V(F)| + |E(F)|)$, and such that knowing* $\hom(C_k, G)$ *and* $\hom(C_k, G')$ *allows one to decide whether F is triangle-free in expected time $O(|V(F)| + |E(F)|)$. Furthermore, if $k \geq 6$ then G and G' are 2-degenerate.*

It is easy to see that Lemma 4.1 implies Lemma 1.5. For $k = 4, 5$, Lemma 4.1 cannot produce graphs G, G' which are $O(1)$-degenerate, as C_4 and C_5 are α-acyclic (and so their homomorphisms can be counted in linear time). The reason for including the cases $k = 4, 5$ (in Lemma 4.1) is that these will be necessary in the proof of Theorem 6. We note that the statement of Lemma 4.1 remains true if we replace "expected time $O(|V(F)| + |E(F)|)$" with "deterministic time $\tilde{O}(|V(F)| + |E(F)|)$".

The proof of Lemma 4.1 is split into several cases, according to the parity of the cycle length k and its residue modulo 3. There are also some exceptional cases: those of $k = 4, 5, 7, 8$. As mentioned before, the proof relies on the use of Möbius inversion as well as on a special case of the main result of [11], stated below as Lemma 4.5. The cleanest variant of our argument appears in Section 4.1, which deals with the case where k is divisible by 3. In this case one can avoid some technical complications which arise for $k \equiv 1, 2 \pmod 3$. The case of an odd k is somewhat easier to handle than the general one, and does not require the use of Möbius inversion or Lemma 4.5. This case is resolved in Sections 4.2 (the case $k \geq 9$), 4.3 (the case $k = 7$) and 4.4 (the case $k = 5$). Section 4.5 handles the remaining "irregular" case $k = 4$. The cases $k \equiv 4, 2 \pmod 6$ are handled in Sections 4.6 and 4.7, respectively. The case $k \equiv 2 \pmod 6$, and $k = 8$ in particular, appears to be the hardest one, requiring a more involved use of Möbius inversion.

We note that in some cases, namely in Sections 4.1, 4.2, 4.3, 4.5 and 4.6, our reduction to HOM-CNT$_{C_k}$ is not just from the triangle-detection problem, but actually from the (evidently harder) triangle-counting problem. Furthermore, in the case where $k \geq 9$ is odd, our reduction shows that even *detecting* C_k-homomorphisms is at least as hard as detecting triangles. Finally, we note that in most cases (with the exception of $k = 4, 5, 8$), our reduction will actually use a single graph G (rather than two graphs G and G'); we decided to state Lemma 4.1 in the above form in order to capture all cases.

We will need the following three simple lemmas, whose proofs we postpone to the end of this section.

Lemma 4.2. *For every $k \geq 3$, there is no partition P of $V(C_k)$ such that $C_k/P \cong C_{k-1}$.*

Lemma 4.3. *Let $k \geq 4$ be even and let $\ell > k/2$ be odd. Then there is no partition P of $V(C_k)$ such that C_k/P contains exactly one odd cycle, and this cycle has length ℓ.*

Lemma 4.4. *Let $k \geq 5$. Denote by C'_{k-2} the graph obtained from the cycle C_{k-2} by adding a pendant vertex (i.e., a vertex of degree one adjacent to one of the vertices of the cycle). Then:*

1. *There are exactly k partitions P of $V(C_k)$ such that $C_k/P \cong C'_{k-2}$, and exactly k partitions P of $V(C_k)$ such that $C_k/P \cong C_{k-2}$.*

2. *For every partition P of $V(C_k)$ satisfying $C_k/P \cong C'_{k-2}$, there are exactly 2 partitions Q of $V(C_k)$ such that P refines Q and $C_k/Q \cong C_{k-2}$. Conversely, for every partition Q of $V(C_k)$ satisfying $C_k/Q \cong C_{k-2}$, there are exactly 2 partitions P of $V(C_k)$ which refine Q and satisfy $C_k/P \cong C'_{k-2}$*

The following lemma follows immediately by combining the main result of [11] (which is stated here as the "if" part of Corollary 2) and Theorem 3.

Lemma 4.5. *For every forest H, one can compute* $\hom(H, F)$ *in expected time $O(|V(F)| + |E(F)|)$.*

We are now ready to begin the proof of Lemma 4.1.

4.1 The case $k \equiv 0 \pmod 3$

Fix any graph F, and let $G = G(F)$ be the graph obtained from F by replacing each edge of F by a path of length ℓ. Note that G is 2-degenerate and that $|V(G)| = |V(F)| + (\ell-1)\cdot|E(F)| = O(|V(F)|+|E(F)|)$. It is easy to see that $\text{girth}(G) \geq 3\ell$ and that $\text{inj}(C_{3\ell}, G) = \ell \cdot \text{inj}(C_3, F)$ (so in particular, $\text{girth}(G) = 3\ell$ if and only if F contains a triangle). Hence, in order to decide whether or not F is triangle-free, it is enough to know $\text{inj}(C_{3\ell}, G)$.

Denote by P_0 the partition of $V(C_{3\ell})$ in which every part is a singleton; so $C_{3\ell}/P_0 \cong C_{3\ell}$. Let $\mathcal{P}$ be the set of all partitions P of $V(C_{3\ell})$ such that either $P = P_0$, or $C_{3\ell}/P$ is a (simple) forest[9]. We claim that for every $P \in \mathcal{P}$, it holds that

$$\text{hom}(C_{3\ell}/P, G) = \sum_{Q \in \mathcal{P}\colon Q \geq P} \text{inj}(C_{3\ell}/Q, G). \tag{9}$$

Fix any $P \in \mathcal{P}$. By Fact 3.2, we have

$$\text{hom}(C_{3\ell}/P, G) = \sum_{Q \in \mathcal{P}(C_{3\ell})\colon Q \geq P} \text{inj}(C_{3\ell}/Q, G),$$

where $\mathcal{P}(C_{3\ell})$ denotes the set of all partitions of $V(C_{3\ell})$. Thus, in order to prove (9) it suffices to show that if $Q \in \mathcal{P}(C_{3\ell}) \setminus \mathcal{P}$ then $\text{inj}(C_{3\ell}/Q, G) = 0$. So let $Q \in \mathcal{P}(C_{3\ell}) \setminus \mathcal{P}$. The definition of $\mathcal{P}$ implies that $C_{3\ell}/Q$ is neither $C_{3\ell}$ nor a forest. Hence, $C_{3\ell}/Q$ must contain a cycle of length shorter than 3ℓ (this cycle could be a loop). However, we have $\text{girth}(G) \geq 3\ell$, implying that there is no injective homomorphism from $C_{3\ell}/Q$ to G. We have thus established (9).

We now invert (9) using Theorem 8. Define $f, g : \mathcal{P} \to \mathbb{Z}$ by $g(P) = \text{hom}(C_{3\ell}/P, G)$, $f(P) = \text{inj}(C_{3\ell}/P, G)$. Then (9) states that Item 1 in Theorem 8 holds (for these f and g). It follows that Item 2 in Theorem 8 holds as well, namely that

$$\text{inj}(C_{3\ell}/P, G) = \sum_{Q \in \mathcal{P}\colon Q \geq P} \mu(P, Q) \cdot \text{hom}(C_{3\ell}/Q, G) \tag{10}$$

for all $P \in \mathcal{P}$. For $P = P_0$ (which is the minimum element of $\mathcal{P}$), (10) becomes:

$$\text{inj}(C_{3\ell}, G) = \sum_{Q \in \mathcal{P}} \mu(P_0, Q) \cdot \text{hom}(C_{3\ell}/Q, G). \tag{11}$$

By the definition of $\mathcal{P}$, the graph $C_{3\ell}/Q$ is a forest for each $Q \in \mathcal{P} \setminus \{P_0\}$. Therefore, for each $Q \in \mathcal{P} \setminus \{P_0\}$, one can compute $\text{hom}(C_{3\ell}/Q, G)$ in expected time $O(|V(G)| + |E(G)|) = O(|V(F)| + |E(F)|)$ by Lemma 4.5. Hence, knowing $\text{hom}(C_{3\ell}, G) = \text{hom}(C_{3\ell}/P_0, G)$ would allow one to compute $\text{inj}(C_{3\ell}, G)$ in expected time $O(|V(F)| + |E(F)|)$, using (11). This in turn allows one one to decide whether F contains a triangle, as $\text{inj}(C_{3\ell}, G) = \ell \cdot \text{inj}(C_3, F)$.

4.2 Odd $k \geq 9$

Let $k \geq 9$ be an odd integer. Having already dealt with the case $k \equiv 0 \pmod 3$ in Section 4.1, we assume, for convenience of presentation, that k is not divisible by 3. We note, however, that a similar reduction works in the case $k \equiv 0 \pmod 3$ as well. Write $k = 3\ell + r$, where $r \in \{1, 2\}$, and set $p := \ell + r - 2$, $q := \ell + r$. Fix any graph F, and let $G = G(F)$ be the graph obtained from F by replacing each

[9] Actually, since $C_{3\ell}$ is connected, so are all of its quotient graphs, so one could replace "forest" by "tree".

edge $\{x,y\}$ of F by a pair of internally-disjoint paths $P^p_{x,y}, P^q_{x,y}$, both having x,y as their endpoints, where $P^p_{x,y}$ is of length p and $P^q_{x,y}$ is of length q. It is easy to see that G is 2-degenerate and that $|V(G)| = |V(F)| + (p+q-2)\cdot|E(F)| = O(|V(F)| + |E(F)|)$. We will need the following claim.

Claim 4.6. *For every odd cycle C in G, there are $x_1,\dots,x_t \in V(F)$ such that $x_1,\dots,x_t,x_1$ is an odd cycle in F, and such that C is the concatenation of the paths $(P^{r_i}_{x_i,x_{i+1}} : 1 \le i \le t)$, where $r_1,\dots,r_t \in \{p,q\}$, and indices are taken modulo t.*

Proof. Let $x_1,\dots,x_t$ be the vertices of C which belong to $V(F)$, in the order they appear when traversing the cycle C. It is easy to see that $t \ge 2$. Note also that for each $1 \le i \le t$, the section of C between x_i and x_{i+1} is either $P^p_{x_i,x_{i+1}}$ or $P^q_{x_i,x_{i+1}}$. This implies that $\{x_i,x_{i+1}\} \in E(F)$ for every $1 \le i \le t$ (with indices taken modulo t).

We see that C is the concatenation of paths $P^{r_i}_{x_i,x_{i+1}}$ $(1 \le i \le t)$, where $r_i \in \{p,q\}$. It only remains to show that t is odd. This in particular will imply that $t \ge 3$, meaning that $x_1,\dots,x_t,x_1$ is a cycle in F. If, by contradiction, t is even, then, since $p \equiv q \pmod 2$, we would get

$$|C| = \sum_{i=1}^{t} e(P^{r_i}_{x_i,x_{i+1}}) \equiv t\cdot p \equiv 0 \pmod 2,$$

in contradiction to the fact that C is odd. Hence, t must be odd. This proves the claim. ■

We now show that there is a homomorphism from C_k to G if and only if F contains a triangle. This will establish the validity of our reduction. In one direction, it is easy to check that if F contains a triangle then G contains a cycle of length k (so, in particular, there is a homomorphism from C_k to G). Indeed, suppose that $x,y,z \in V(F)$ span a triangle in F. If $k \equiv 1 \pmod 3$ then a cycle of length k is formed by concatinating the paths $P^p_{x,y}, P^q_{y,z}, P^q_{z,x}$, as in this case $p + 2q = \ell - 1 + 2(\ell+1) = 3\ell + 1 = k$. And if $k \equiv 2 \pmod 3$ then a cycle of length k is formed by concatinating the paths $P^p_{x,y}, P^p_{y,z}, P^q_{z,x}$, as in this case $2p + q = 2\ell + \ell + 2 = 3\ell + 2 = k$.

Suppose now, in the other direction, that F is triangle-free. Then, by Claim 4.6, the shortest odd cycle in G has length at least $5p = 5(\ell + r - 2) = 5(3\ell + r) - 10\ell - 10 = 5k - 10\ell - 10 = 5k - 10\cdot\lfloor k/3\rfloor - 10$. It is easy to verify that for each odd $k \ge 9$ not divisible by 3, it holds that $5k - 10\cdot\lfloor k/3\rfloor - 10 > k$. Hence, all odd cycles in G have length larger than k. Since the homomorphic image of an odd cycle must itself contain an odd cycle, there can be no homomorphism from C_k to G, as required. This completes the proof.

4.3 The case $k = 7$

Here we use the following reduction: given a graph F, we replace each edge $\{x,y\}$ of F by two internally-disjoint paths, one of length 2 and the other of length 3. The resulting graph is denoted by $G = G(F)$. It is easy to see that G' is 2-degenerate, that $|V(G)| = |V(F)| + 3|E(F)| = O(|V(F)| + |E(F)|)$, that $\text{girth}(G) = 5$, that $\text{inj}(C_7, G) = 7\cdot\text{inj}(C_3, F)$, and that every subgraph of G on at most 7 vertices contains at most one cycle. It follows that if P is a partition of $V(C_7)$ such that $\text{inj}(C_7/P, G) > 0$, then C_7/P must be isomorphic to one of the graphs C_7, C'_5, C_5, where C'_5 is the graph obtained from C_5 by adding a pendant vertex. To see this, note first that C_7/P must contain an odd cycle for every partition P of $V(C_7)$. On the other hand, by the properties of G' stated above, if $\text{inj}(C_7/P, G) > 0$ then this odd cycle must be the unique cycle in C_7/P and have length either 5 or 7. It is easy to see that C_7, C'_5, C_5 are the only quotient graphs of C_7 satisfying these conditions.

By Item 1 of Lemma 4.4, there are 7 partitions P of $V(C_7)$ for which $C_7/P \cong C'_5$, and 7 such partitions for which $C_7/P \cong C_5$. Hence, (1) gives

$$\text{hom}(C_7, G) = \text{inj}(C_7, G) + 7\cdot\text{inj}(C'_5, G) + 7\cdot\text{inj}(C_5, G). \tag{12}$$

Observe that $\mathrm{inj}(C_5, G) = 10 \cdot e(F)$ and that

$$\mathrm{inj}(C_5', G) = \sum_{x \in V(F)} 2 \cdot 2 \cdot d_F(x) \cdot (d_F(x) - 1),$$

which means that both $\mathrm{inj}(C_5, G)$ and $\mathrm{inj}(C_5', G)$ can be computed in time $O(|V(F)| + |E(F)|)$, as they only depend on the degree sequence of F. So we see that knowing $\mathrm{hom}(C_7, G)$ would allow one to compute $\mathrm{inj}(C_7, G)$ in time $O(|V(F)| + |E(F)|)$ by using (12). But as $\mathrm{inj}(C_7, G) = 7 \cdot \mathrm{inj}(C_3, F)$, knowing $\mathrm{inj}(C_7, G)$ would allow one to decide whether or not F is triangle-free.

4.4 The case $k = 5$

Given a graph F, let $G' = G'(F)$ be the graph obtained by adding for each edge $\{x, y\} \in E(G)$, a path of length two between x and y, where the middle vertex of this path is new. (Note that, unlike in a subdivision, the edges of F are kept.) Then $|V(G')| = |V(F)| + |E(F)|$ and $|E(G')| = 3|e(G)|$. Let C_3' denote the graph obtained by adding a pendant vertex to the triangle C_3. It is easy to see that the only simple quotient graphs of C_5 are C_5, C_3', C_3. Therefore, by using (1) and Item 1 of Lemma 4.4, we conclude that $\mathrm{hom}(C_5, F_0) = \mathrm{inj}(C_5, F_0) + 5 \cdot \mathrm{inj}(C_3', F_0) + 5 \cdot \mathrm{inj}(C_3, F_0)$ holds for every graph F_0. It is also straightforward to check that $\mathrm{inj}(C_3, G') = \mathrm{inj}(C_3, G)$, $\mathrm{inj}(C_5, G') = \mathrm{inj}(C_5, G) + 3 \cdot \mathrm{inj}(C_3, G)$ and $\mathrm{inj}(C_3', G') = 2 \cdot \mathrm{inj}(C_3', G) + 2 \cdot \mathrm{inj}(C_3, G)$. It follows that

$$\mathrm{hom}(C_5, F) = \mathrm{inj}(C_5, F) + 5 \cdot \mathrm{inj}(C_3', F) + 5 \cdot \mathrm{inj}(C_3, F)$$

and

$$\mathrm{hom}(C_5, G') = \mathrm{inj}(C_5, G') + 5 \cdot \mathrm{inj}(C_3', G') + 5 \cdot \mathrm{inj}(C_3, G') = \mathrm{inj}(C_5, F) + 10 \cdot \mathrm{inj}(C_3', F) + 18 \cdot \mathrm{inj}(C_3, F).$$

Subtracting, we obtain $\mathrm{hom}(C_5, G') - \mathrm{hom}(C_5, F) = 5 \cdot \mathrm{inj}(C_3', F) + 13 \cdot \mathrm{inj}(C_3, F)$. Now, note that $5 \cdot \mathrm{inj}(C_3', F) + 13 \cdot \mathrm{inj}(C_3, F) > 0$ if and only if F contains a triangle. Thus, knowing the quantity $\mathrm{hom}(C_5, G') - \mathrm{hom}(C_5, F)$ would enable one to decide whether F is triangle-free. So we see that the assertion of Lemma 4.1 holds with $G := F$ and G' as defined above.

4.5 The case $k = 4$

We start by observing that given a graph F, knowing $\mathrm{hom}(C_4, F)$ allows us to find $\mathrm{inj}(C_4, F)$ in time $O(|V(F)| + |E(F)|)$. Indeed, the only simple quotient graphs of C_4 are C_4 itself, K_2 and $K_{1,2}$ (where K_2 is an edge and $K_{1,2}$ is a star with two leaves). Now, both $\mathrm{hom}(K_2, F) = 2|E(F)|$ and $\mathrm{hom}(K_{1,2}, F) = \sum_{v \in V(F)} d(v)^2$ depend only on the degree-sequence of F, and hence can be computed in time $O(|V(F)| + |E(F)|)$. Therefore, by (1), knowing $\mathrm{hom}(C_4, \cdot)$ allows one to find $\mathrm{inj}(C_4, \cdot)$ in time $O(|V(F)| + |E(F)|)$, as claimed.

Next, we use the same construction as in the case $k = 5$. Namely, given a graph F, we let G' be the graph obtained by adding for each edge $\{x, y\} \in E(F)$, a path of length two between x and y, where the middle vertex of this path is new. Then $|V(G')| = |V(F)| + |E(F)|$ and $|E(G')| = 3|E(F)|$. Furthermore, it is not hard to check that $\mathrm{inj}(C_4, G') = \mathrm{inj}(C_4, F) + 4 \cdot \mathrm{inj}(C_3, F)$. Hence, knowing $\mathrm{inj}(C_4, F)$ and $\mathrm{inj}(C_4, G')$ would allow one to solve for $\mathrm{inj}(C_3, F)$ and thus decide if F is triangle-free. So we see that the assertion of Lemma 4.1 holds with $G := F$ and G' as defined above.

4.6 The case $k \equiv 4 \pmod 6$

Here we consider the case that $k \equiv 4 \pmod 6$ and $k \geq 10$. Set $\ell := \lfloor k/3 \rfloor$; so $k = 3\ell + 1$. We use the following reduction: for a graph F, let $G = G(F)$ be the graph obtained from F by replacing each edge $\{x, y\}$ of F by a pair of internally-disjoint paths, both having x, y as their endpoints, one of length ℓ, denoted $P^{\ell}_{x,y}$, and the other of length $\ell + 1$, denoted $P^{\ell+1}_{x,y}$. It is easy to see that G is 2-degenerate, that $|V(G)| = |V(F)| + (2\ell - 1) \cdot |E(F)| = O(|V(F)| + |E(F)|)$, that $\text{inj}(C_{3\ell+1}, G) = (3\ell + 1) \cdot \text{inj}(C_3, F)$, and that every cycle in G either has length at least 3ℓ, or has length exactly $2\ell + 1$. Note also that every subgraph of G on at most $k = 3\ell + 1$ vertices contains at most one cycle.

Let $\mathcal{P}$ be the set of all partitions P of $V(C_k)$ such that either $C_k/P \cong C_k$ (i.e., P, is the partition of $V(C_k)$ into singletons) or C_k/P is a (simple) forest. We claim that for every $P \in \mathcal{P}$, it holds that

$$\text{hom}(C_k/P, G) = \sum_{Q \in \mathcal{P}\colon Q \geq P} \text{inj}(C_k/Q, G). \tag{13}$$

Fix any $P \in \mathcal{P}$. By Fact 3.2, we have

$$\text{hom}(C_k/P, G) = \sum_{Q \in \mathcal{P}(C_k)\colon Q \geq P} \text{inj}(C_k/Q, G),$$

where $\mathcal{P}(C_k)$ denotes the set of all partitions of $V(C_k)$. Thus, in order to prove (13), it suffices to show that if $Q \in \mathcal{P}(C_k) \setminus \mathcal{P}$ then $\text{inj}(C_k/Q, G) = 0$. So let $Q \in \mathcal{P}(C_k) \setminus \mathcal{P}$, and suppose by contradiction that $\text{inj}(C_k/Q, G) > 0$. By the definition of $\mathcal{P}$, C_k/Q contains a cycle of length less than k. This cycle must have length either $2\ell + 1$ or $3\ell = k - 1$, since the length of any cycle in G is either $2\ell + 1$ or at least 3ℓ. Moreover, C_k/Q must contain exactly one cycle, since every subgraph of G of order at most k contains at most one cycle. By Lemma 4.2, C_k/Q cannot be isomorphic to C_{k-1}. By Lemma 4.3, C_k/Q cannot contain a cycle of length $2\ell + 1$, since $2\ell + 1$ is odd and $2\ell + 1 > (3\ell + 1)/2 = k/2$. We have thus arrived at a contradiction. This proves (13).

Denote by P_0 the partition of $V(C_k)$ in which every part is a singleton; so $C_k/P_0 \cong C_k$. By combining (13) with Theorem 8, we obtain

$$\text{inj}(C_k, G) = \text{inj}(C_k/P_0, G) = \sum_{Q \in \mathcal{P}} \mu(P_0, Q) \cdot \text{hom}(C_k/Q, G). \tag{14}$$

By the definition of $\mathcal{P}$, the graph C_k/Q is a forest for each $Q \in \mathcal{P} \setminus \{P_0\}$. It follows that for each $Q \in \mathcal{P} \setminus \{P_0\}$, one can compute $\text{hom}(C_k/Q, G)$ in expected time $O(|V(G)| + |E(G)|) = O(|V(F)| + |E(F)|)$ by Lemma 4.5. Hence, knowing $\text{hom}(C_k, G) = \text{hom}(C_k/P_0, G)$ allows one to compute $\text{inj}(C_k, G)$ in expected time $O(|V(F)| + |E(F)|)$ using (14). This in turn allows one to decide whether or not F is triangle-free, since $\text{inj}(C_k, G) = k \cdot \text{inj}(C_3, F)$.

4.7 The case $k \equiv 2 \pmod 6$

Proof. Let $k \geq 8$ be such that $k \equiv 2 \pmod 6$. Set $\ell := \lfloor k/3 \rfloor$; so $k = 3\ell + 2$. We use the same reduction as in Section 4.1, but the analysis is somewhat more involved. For a graph F, let $G = G(F)$ be the graph obtained from F by replacing each edge $\{x, y\}$ of F by a path of length ℓ. It is easy to see that G is 2-degenerate, that $|V(G)| = |V(F)| + (\ell - 1) \cdot |E(F)| = O(|V(F)| + |E(F)|)$, that $\text{girth}(G) \geq 3\ell$, and that if F is triangle-free then in fact $\text{girth}(G) \geq 4\ell$. Note also that every subgraph of G on at most $k = 3\ell + 2$ vertices contains at most one cycle.

As in Lemma 4.4, we denote by C'_{k-2} the graph obtained from C_{k-2} by adding a pendant vertex. Let $\mathcal{P}$ be the set of all partitions P of $V(C_k)$ such that either C_k/P is isomorphic to one of the graphs C_k, C'_{k-2}, C_{k-2} or C_k/P is a (simple) forest. We claim that for every $P \in \mathcal{P}$, it holds that

$$\hom(C_k/P, G) = \sum_{Q \in \mathcal{P}\colon Q \geq P} \mathrm{inj}(C_k/Q, G). \tag{15}$$

Fix any $P \in \mathcal{P}$. As in previous proofs, we first note that by Fact 3.2, for every $P \in \mathcal{P}$ we have

$$\hom(C_k/P, G) = \sum_{Q \in \mathcal{P}(C_k)\colon Q \geq P} \mathrm{inj}(C_k/Q, G),$$

where $\mathcal{P}(C_k)$ denotes the set of all partitions of $V(C_k)$. So in order to prove (15), we only need to show that $\mathrm{inj}(C_k/Q, G) = 0$ for each $Q \in \mathcal{P}(C_k) \setminus \mathcal{P}$. Let $Q \in \mathcal{P}(C_k) \setminus \mathcal{P}$. By the definition of $\mathcal{P}$, either C_k/Q contains more than one cycle, or it contains a cycle of length (strictly) smaller than $k-2$. To see this, recall first that, according to Lemma 4.2, C_k/Q is not isomorphic to C_{k-1}. Hence, if C_k/Q contains exactly one cycle, which is of length at least $k-2$, then C_k/Q must be isomorphic to one of the graphs C_k, C'_{k-2}, C_{k-2}. But this is impossible as $Q \notin \mathcal{P}$. Now, recall that G contains neither a cycle of length smaller than $k-2 = 3\ell$, nor a subgraph on at most k vertices having more than one cycle. So we see that $\mathrm{inj}(C_k/Q, G) = 0$ for each $Q \in \mathcal{P}(C_k) \setminus \mathcal{P}$, as required.

We have thus proved (15). By inverting (15) using Theorem 8, we get that

$$\mathrm{inj}(C_k/P, G) = \sum_{Q \in \mathcal{P}\colon Q \geq P} \mu(P,Q) \cdot \hom(C_k/Q, G) \tag{16}$$

for each $P \in \mathcal{P}$. Let P_0 denote the partition of $V(C_k)$ into singletons (so $C_k/P_0 \cong C_k$), let $\mathcal{P}_1$ be the set of all $P \in \mathcal{P}$ such that $C_k/P \cong C'_{k-2}$, let $\mathcal{P}_2$ be the set of all $P \in \mathcal{P}$ such that $C_k/P \cong C_{k-2}$, and put $\mathcal{P}' = \mathcal{P} \setminus (\{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2)$. By the definition of $\mathcal{P}$, the graph C_k/P is a forest for every $P \in \mathcal{P}'$. By Item 1 of Lemma 4.4, we have $|\mathcal{P}_1| = |\mathcal{P}_2| = k$. By Item 2 of Lemma 4.4, for each $P \in \mathcal{P}_1$ there are exactly two $Q \in \mathcal{P}_2$ such that $Q \geq P$, and for every $Q \in \mathcal{P}_2$ there are exactly two $P \in \mathcal{P}_1$ such that $P \leq Q$. By summing (16) over all $P \in \{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2$ we obtain

$$\begin{aligned}
&\mathrm{inj}(C_k, G) + k \cdot \mathrm{inj}(C'_{k-2}, G) + k \cdot \mathrm{inj}(C_{k-2}, G) = \sum_{P \in \{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2} \mathrm{inj}(C_k/P, G) = \\
&\sum_{P \in \{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2} \; \sum_{Q \in \mathcal{P}\colon Q \geq P} \mu(P,Q) \cdot \hom(C_k/Q, G) = \\
&\sum_{Q \in \mathcal{P}} \; \sum_{\substack{P \in \{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2\colon \\ P \leq Q}} \mu(P,Q) \cdot \hom(C_k/Q, G) = \\
&\sum_{Q \in \mathcal{P}'} c_Q \cdot \hom(C_k/Q, G) + \sum_{\substack{P,Q \in \{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2\colon \\ P \leq Q}} \mu(P,Q) \cdot \hom(C_k/Q, G),
\end{aligned} \tag{17}$$

where $(c_Q : Q \in \mathcal{P}')$ are (explicit) integers whose values will not be important for our argument. In the last equality above, we split the sum over $Q \in \mathcal{P}$ into two parts: $Q \in \mathcal{P}'$ and $Q \in \mathcal{P} \setminus \mathcal{P}' = \{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2$.

Let S denote the last sum appearing in (17). We claim that $S = \hom(C_k, G)$. To this end, first note that for $Q = P_0$, the only $P \in \mathcal{P}$ satisfying $P \leq Q$ is $P = P_0 = Q$. By Theorem 8, $\mu(P_0, P_0) = 1$. Next, observe that for each $Q \in \mathcal{P}_1$, there are exactly 2 partitions $P \in \mathcal{P}$ satisfying $P \leq Q$, namely P_0 and Q itself. Both of these partitions are in $\{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2$. By (7), for every $Q \in \mathcal{P}_1$ it holds that $\mu(P_0, Q) + \mu(Q, Q) = 0$. Finally, observe that for every $Q \in \mathcal{P}_2$, there are exactly 4 partitions $P \in \mathcal{P}$ satisfying $P \leq Q$; these are P_0, two partitions belonging to $\mathcal{P}_1$, which we denote by R_1 and R_2, and Q itself. All of these partitions are in $\{P_0\} \cup \mathcal{P}_1 \cup \mathcal{P}_2$. By (7), it holds that $\mu(P_0, Q) + \mu(R_1, Q) + \mu(R_2, Q) + \mu(Q, Q) = 0$. This shows that $S = \hom(C_k, G)$, as required. Now, plugging this equality into (17), we obtain:

$$\mathrm{inj}(C_k, G) + k \cdot \mathrm{inj}(C'_{k-2}, G) + k \cdot \mathrm{inj}(C_{k-2}, G) = \sum_{Q \in \mathcal{P}'} c_Q \cdot \hom(C_k/Q, G) + \hom(C_k, G). \tag{18}$$

For each $Q \in \mathcal{P}'$, the homomorphism number $\hom(C_k/Q, G)$ can be computed in expected time $O(|V(G)| + |E(G)|) = O(|V(F)| + |E(F)|)$, since C_k/Q is a forest (see Lemma 4.5). Hence, one can compute the sum on the right-hand side of (18) in expected time $O(|V(F)| + |E(F)|)$. It follows that knowing $\hom(C_k, G)$ allows one to compute $h := \mathrm{inj}(C_k, G) + k \cdot \mathrm{inj}(C'_{k-2}, G) + k \cdot \mathrm{inj}(C_{k-2}, G)$ in expected time $O(|V(F)| + |E(F)|)$. It is easy to see that $\mathrm{inj}(C'_{k-2}, G) + \mathrm{inj}(C_{k-2}, G) > 0$ if and only if F contains a triangle, and that if $k > 8$ then $\mathrm{inj}(C_k, G) = 0$, since the length of every cycle in G is a multiple of $\ell = \lfloor k/3 \rfloor$, while $k = 3\ell + 2$ is not a multiple of ℓ if $k > 8$. So we see that for $k > 8$, knowing whether $h > 0$ allows one to decide whether F is triangle-free. This completes the proof in the case $k > 8$.

It remains to handle the case $k = 8$. The exception in this case is that $\mathrm{inj}(C_8, G)$ is not necessarily 0, as $\mathrm{inj}(C_8, G) = 2 \cdot \mathrm{inj}(C_4, F)$. To overcome this difficulty, we use an additional construction, as follows. Let $G' = G'(F)$ be the graph obtained from F by replacing each edge $\{x, y\} \in E(F)$ with two internally-disjoint paths: one of length 2 and the other of length 3. It is easy to see that G' is 2-degenerate, that $|V(G')| = |V(F)| + 3|E(F)| = O(|V(F)| + |E(F)|)$, that $\mathrm{girth}(G') = 5$, and that every subgraph of G' on at most 8 vertices contains at most one cycle. By combining these last two facts with Lemma 4.3, we conclude that every $Q \in \mathcal{P}(C_8) \setminus \mathcal{P}$ satisfies $\mathrm{inj}(C_8/Q, G') = 0$ (due to similarity to previous arguments, we omit the details). From this it follows that (15) also holds for the graph G', namely that

$$\hom(C_8/P, G') = \sum_{Q \in \mathcal{P} \colon Q \geq P} \mathrm{inj}(C_k/Q, G')$$

for every $P \in \mathcal{P}$. By repeating the above steps used to derive (18), we obtain a similar equality for G', namely that

$$\mathrm{inj}(C_8, G') + 8 \cdot \mathrm{inj}(C'_6, G') + 8 \cdot \mathrm{inj}(C_6, G') = \sum_{Q \in \mathcal{P}'} c_Q \cdot \hom(C_8/Q, G') + \hom(C_8, G'). \tag{19}$$

Observe that

$$\begin{aligned} &\mathrm{inj}(C_6, G) = \mathrm{inj}(C_6, G') = 2 \cdot \mathrm{inj}(C_3, F), \;\; \mathrm{inj}(C_8, G) = 2 \cdot \mathrm{inj}(C_4, F), \\ &\mathrm{inj}(C_8, G') = 2 \cdot \mathrm{inj}(C_4, F) + 8 \cdot \mathrm{inj}(C_3, F). \end{aligned} \tag{20}$$

We claim that additionally, one has

$$\mathrm{inj}(C'_6, G') = 2 \cdot \mathrm{inj}(C'_6, G) + 2 \cdot \mathrm{inj}(C_3, F). \tag{21}$$

To prove (21), it will be convenient to consider *unlabeled* copies; for graphs H_1, H_2, let us denote by $\mathrm{cop}(H_1, H_2)$ the number of unlabeled copies of H_1 in H_2; so $\mathrm{cop}(H_1, H_2) = \mathrm{inj}(H_1, H_2)/\mathrm{aut}(H_1)$, where $\mathrm{aut}(H_1)$ is the number of automorphisms of H_1. It is easy to see that $\mathrm{cop}(C_6, G) = \mathrm{cop}(C_6, G') = \mathrm{cop}(C_3, F)$. Now, fix any triangle $\{x, y, z\}$ in F, and consider the copy of C_6 corresponding to $\{x, y, z\}$ in the graphs G, G'. In G, this copy of C_6 can be extended to a copy of C'_6 in exactly

$(d_F(x)-2)+(d_F(y)-2)+(d_F(z)-2)=d_F(x)+d_F(y)+d_F(z)-6$ ways, for in order to add a pendant vertex, one must select some $w \in \{x,y,z\}$ and some neighbour $v \in N_F(w)\setminus\{x,y,z\}$ of w which is outside of the triangle $\{x,y,z\}$, and choose as the pendant vertex the middle vertex of the 2-edge path which replaces the edge $\{v,w\}$. In G', on the other hand, there are two types of copies of C_6' which extend the given copy of C_6. One type is obtained by selecting, as before, vertices $w \in \{x,y,z\}$ and $v \in N_F(w)\setminus\{x,y,z\}$, and then selecting the pendant vertex to be a vertex adjacent to w on one of the 2 internally-disjoint paths between w and v which replace the edge $\{v,w\}$; there are $2\cdot(d_F(x)+d_F(y)+d_F(z)-6)$ copies of C_6' of this type. The second type is obtained by choosing as the pendant vertex one of the internal vertices of the 3-edge paths between x, y and z; there are exactly 6 copies of C_6' of this type. Summing over all triangles $\{x,y,z\}$ of F, we see that

$$\mathrm{cop}(C_6',G') = 2\cdot\sum_{\{x,y,z\}}(d_F(x)+d_F(y)+d_F(z)-6)+6\cdot\mathrm{cop}(C_3,F) = 2\cdot\mathrm{cop}(C_6',G)+\mathrm{inj}(C_3,F).$$

By multiplying the above equation by $\mathrm{aut}(C_6')=2$, we get (21).

Finally, we plug (20) and (21) into (18) and (19) to obtain

$$\begin{aligned}2\cdot\mathrm{inj}(C_4,F)+8\cdot\mathrm{inj}(C_6',G)+16\cdot\mathrm{inj}(C_3,F) &= \mathrm{inj}(C_8,G)+8\cdot\mathrm{inj}(C_6',G)+8\cdot\mathrm{inj}(C_6,G)\\ &= \sum_{Q\in\mathcal{P}'} c_Q\cdot\mathrm{hom}(C_8/Q,G)+\mathrm{hom}(C_8,G),\end{aligned}\tag{22}$$

and

$$\begin{aligned}&2\cdot\mathrm{inj}(C_4,F)+16\cdot\mathrm{inj}(C_6',G)+40\cdot\mathrm{inj}(C_3,F) =\\ &2\cdot\mathrm{inj}(C_4,F)+8\cdot\mathrm{inj}(C_3,F)+8\cdot(2\cdot\mathrm{inj}(C_6',G)+2\cdot\mathrm{inj}(C_3,F))+16\cdot\mathrm{inj}(C_3,F) =\\ &\mathrm{inj}(C_8,G')+8\cdot\mathrm{inj}(C_6',G')+8\cdot\mathrm{inj}(C_6,G') = \sum_{Q\in\mathcal{P}'} c_Q'\cdot\mathrm{hom}(C_8/Q,G')+\mathrm{hom}(C_8,G'),\end{aligned}\tag{23}$$

respectively. By subtracting (22) from (23), we obtain

$$8\cdot\mathrm{inj}(C_6',G)+24\cdot\mathrm{inj}(C_3,F) = S'+\mathrm{hom}(C_8,G')-\mathrm{hom}(C_8,G),$$

where

$$S' := \sum_{Q\in\mathcal{P}'} c_Q\cdot(\mathrm{hom}(C_8/Q,G')-\mathrm{hom}(C_8/Q,G)).$$

As before, S' can be computed in expected time $O(|V(F)|+|E(F)|)$ because C_8/Q is a forest for each $Q \in \mathcal{P}'$ (see Lemma 4.5). Thus, knowing $\mathrm{hom}(C_8,G)$ and $\mathrm{hom}(C_8,G')$ allows one to compute $h := 8\cdot\mathrm{inj}(C_6',G)+24\cdot\mathrm{inj}(C_3,F)$ in expected time $O(|V(F)|+|E(F)|)$. But since $h>0$ if and only if F contains a triangle, knowing h allows one to decide whether or not F is triangle-free, as required. ■

Proof of Lemma 4.2. Suppose by contradiction that there is a partition $P=\{U_1,\dots,U_{k-1}\}$ of $V(C_k)$ such that C_k/P is the cycle $U_1,\dots,U_{k-1},U_1$. Clearly, one of the parts U_i has 2 elements, and all other parts have 1. Suppose without loss of generality that $|U_1|=2$, and write $U_i=\{u_i\}$ for $2\le i\le k-1$. Then $u_2,\dots,u_{k-1}$ is a path in C_k because $U_2,\dots,U_{k-1}$ is a path in C_k/P. This implies that the two vertices in U_1 are adjacent along the cycle C_k, which in turn means that U_1 has a loop in C_k/P. Hence, C_k/P is not isomorphic to C_{k-1} (since it contains a loop), a contradiction. ■

Proof of Lemma 4.3. Suppose, by contradiction, that there exists a partition P of C_k as in the statement of the claim, and let $U_1, \dots, U_\ell \in P$ be such that $U_1, \dots, U_\ell, U_1$ is the unique cycle in C_k/P. Since $\ell > k/2$, there must be $1 \leq i \leq \ell$ such that $|U_i| = 1$. Suppose, without loss of generality, that $|U_1| = 1$, and write $U_1 = \{u\}$. Let v, w be the neighbours of u in the cycle C_k. Since U_2, U_ℓ are both adjacent to U_1 in C_k/P, it must be the case that $|U_2 \cap \{v, w\}| = 1$ and $|U_\ell \cap \{v, w\}| = 1$. Since $U_1, \dots, U_\ell, U_1$ is the unique odd cycle in C_k/P, deleting U_1 from C_k/P leaves a bipartite graph. This means that in the graph $(C_k/P) \setminus \{U_1\}$, all walks between U_2 and U_ℓ have the same parity. Note that $U_2, U_3, \dots, U_\ell$ is a path of *odd* length $\ell - 2$ between U_2 and U_ℓ. On the other hand, since there is a path of length $k-2$ between v and w which avoids u (in the cycle C_k), there must be a (not necessarily simple) walk of *even* length $k-2$ between U_2 and U_ℓ which avoids U_1. We have thus arrived at a contradiction. ∎

Proof of Lemma 4.4. Let $x_1, \dots, x_k$ be the vertices of C_k (appearing in this order when traversing the cycle). It is easy to see that both items in the lemma follow from the following assertion: for a partition P of $V(C_k)$, $C_k/P \cong C'_{k-2}$ holds if and only if P is of the form $\{\{x_i, x_{i+2}\}\} \cup \{\{x_j\} : j \in [k] \setminus \{i, i+2\}\}$ for some $1 \leq i \leq k$, and $C_k/P \cong C_{k-2}$ holds if and only if P is of the form $\{\{x_i, x_{i+2}\}, \{x_{i+1}, x_{i+3}\}\} \cup \{\{x_j\} : j \in [k] \setminus \{i, i+1, i+2, i+3\}\}$ for some $1 \leq i \leq k$. (All indices are taken modulo k.) Let us now prove this assertion.

Assume first that $C_k/P \cong C'_{k-2}$. Then P consists of one part of size 2 and $k-2$ parts of size 1 (since $|V(C'_{k-2})| = k-1$). Let $U = \{x, y\} \in P$ be the part of P of size 2. Then x, y are not adjacent in C_k, because otherwise C_k/P would have a loop. Furthermore, x, y must have a common neighbour in C_k, because otherwise U would have degree 4 in C_k/P, which is impossible as $C_k/P \cong C'_{k-2}$. Thus, we must have that $\{x, y\} = \{x_i, x_{i+2}\}$ for some $1 \leq i \leq k$, as required.

Suppose now that $C_k/P \cong C_{k-2}$. Since $|P| = k-2$, the "type" of P must be either $(3, 1, \dots, 1)$, meaning that P has one part of size 3 and $k-3$ parts of size 1, or $(2, 2, 1, \dots, 1)$, meaning that P has two parts of size 2 and $k-4$ parts of size 1. The former case is impossible, since if P has type $(3, 1, \dots, 1)$ then the part $U \in P$ of size 3 must either have a loop or degree at least 3 in C_k/P, contradicting the assumption that $C_k/P \cong C_{k-2}$. So suppose that P has type $(2, 2, 1, \dots, 1)$, and let $U = \{x, y\}$ and $V = \{z, w\}$ be the parts of P of size 2. Then $\{x, y\}, \{z, w\} \notin E(C_k)$, as otherwise C_k/P would have loops. Furthermore, considering first the pair x, y, we see that x, y must have a common neighbour in C_k, because otherwise the degree of U in C_k would be at least 3. If the common neighbour u of x, y was neither z nor w, then u would have degree 1 in C_k/P, a contradiction. Thus, either z or w is a common neighbour of x, y. Symmetrically, either x or y is a common neighbour of z, w. It follows that $\{U, V\} = \{\{x_i, x_{i+2}\}, \{x_{i+1}, x_{i+3}\}\}$ for some $1 \leq i \leq k$, as required. ∎

5 Solvability of HOM-CNT$_H$ is Hereditary: Proof of Lemma 1.6

In this section we prove Lemma 1.6. We will actually prove the following more general statement.

Lemma 5.1. *For every graph H there is $k = k(H)$ such that the following holds. For every graph G there are graphs $G_1, \dots, G_k$, computable in time $O(|V(G)| + |E(G)|)$, such that $|V(G_i)| = O(|V(G)|)$ and $|E(G_i)| = O(|E(G)|)$ for every $i = 1, \dots, k$, and such that knowing $\hom(H, G_1), \dots, \hom(H, G_k)$ allows one to find $\hom(H', G)$ for all induced subgraphs H' of H in time $O(1)$. Furthermore, if G is $O(1)$-degenerate, then so are $G_1, \dots, G_k$.*

It is easy to see that Lemma 5.1 implies Lemma 1.6.

We now state an important lemma concerned with computing *linear combinations of homomorphism counts*. As mentioned in the introduction, many of the reductions presented in this paper — including the proof of Lemma 5.1 — rely on this lemma.

Lemma 5.2. *Let $H_1, \dots, H_k$ be pairwise non-isomorphic graphs and let $c_1, \dots, c_k$ be non-zero constant. For every graph G there are graphs $G_1, \dots, G_k$, computable in time $O(|V(G)|+|E(G)|)$, such that $|V(G_i)| = O(|V(G)|)$ and $|E(G_i)| = O(|E(G)|)$ for every $i = 1, \dots, k$, and such that knowing $b_j := c_1 \cdot \hom(H_1, G_j) + \dots + c_k \cdot \hom(H_k, G_j)$ for every $j = 1, \dots, k$ allows one to find $\hom(H_1, G), \dots, \hom(H_k, G)$ in time $O(1)$. Furthermore, if G is $O(1)$-degenerate, then so are $G_1, \dots, G_k$.*

Lemma 5.2 is implicit in [16]. For completeness, we give its proof in the appendix.

Proof of Lemma 1.6. Let H be a graph on h vertices. For a graph F, we denote by $F + K_h$ the graph obtained from F by adding a clique of size h and connecting it to $V(F)$ with a complete bipartite graph. We start by showing that for every graph F, it holds that

$$\hom(H, F + K_h) = \sum_{U \subseteq V(H)} \hom(H[U], F) \cdot \hom(H[V(H) \setminus U], K_h). \tag{24}$$

To see that (24) holds, let us assign to each function $\varphi : V(H) \to V(F + K_h)$ the set $U = U(\varphi) := \varphi^{-1}(V(F)) \subseteq V(H)$. Our definition of the graph $F + K_h$ guarantees that a function $\varphi : V(H) \to V(F + K_h)$ is a homomorphism if and only if $\varphi|_{U(\varphi)}$ is a homomorphism from $H[U]$ to F and $\varphi|_{V(H)\setminus U(\varphi)}$ is a homomorphism from $H[V(H) \setminus U(\varphi)]$ to K_h. By summing over all possible values of U, we get (24). Note that $U(\varphi)$ may be empty (in case $\mathrm{Im}(\varphi) \subseteq K_h$); in this case $\hom(H[U], F) = 1$ (as $H[U]$ is the empty graph).

Let $H_1, H_2, \dots, H_k$ be an enumeration of all induced subgraphs of H (including the empty one), up to isomorphism (that is, $H_1, \dots, H_k$ are pairwise non-isomorphic). For each $1 \leq i \leq k$, set

$$c_i := \sum_{\substack{U \subseteq V(H): \\ H[U] \cong H_i}} \hom(H[V(H) \setminus U], K_h).$$

Note that $c_1, \dots, c_k$ depend only on H (and not on the "host graph" G). With this notation, we can rewrite (24) as follows:

$$\hom(H, F + K_h) = \sum_{i=1}^{k} c_i \cdot \hom(H_i, F). \tag{25}$$

Note that for each $1 \leq i \leq k$ we have $c_i > 0$, since there is some $U \subseteq V(H)$ for which $H[U] \cong H_i$ (by our choice of $H_1, \dots, H_k$), and for this U it clearly holds that $\hom(H[V(H) \setminus U], K_h) > 0$. In particular, $c_1, \dots, c_k$ are non-zero.

Now let G be a graph. Apply Lemma 5.2 (to the graph G), and let $G'_1, \dots, G'_k$ be the graphs given by that lemma. For each $1 \leq i \leq k$, set $G_i := G'_i + K_h$, noting that $|V(G_i)| = |V(G'_i)| + h = O(|V(G)| + |E(G)|)$ and $|E(G_i)| = |E(G'_i)| + |V(G'_i)| \cdot h + \binom{h}{2} = O(|V(G)| + |E(G)|)$, where in both cases the last equality is guaranteed by Lemma 5.2. Furthermore, if G is $O(1)$-degenerate then so is G'_i for every $1 \leq i \leq k$ (by Lemma 5.2), and hence so is G_i for every $1 \leq i \leq k$ (indeed, if a graph F is κ-degenerate then $F + K_h$ is $(\kappa + h)$-degenerate). Crucially, observe that knowing $\hom(H, G_1), \dots, \hom(H, G_k)$ allows one to compute $c_1 \cdot \hom(H_1, G'_j) + \dots + c_k \cdot \hom(H_k, G'_j)$ for every $1 \leq j \leq k$ (by (25)), which in turn allows one to find $\hom(H_1, G), \dots, \hom(H_k, G)$ in time $O(1)$ (by our choice of $G'_1, \dots, G'_k$ via Lemma 5.2). This completes the proof, as every induced subgraph of H is isomorphic to one of the graphs $H_1, \dots, H_k$. ■

6 Counting Copies: Proof of Lemma 1.7

We start with the following more general lemma, from which Lemma 1.7 will easily follow.

Lemma 6.1. *For every graph H there is $k = k(H)$ such that the following holds. For every graph G there are graphs $G_1, \dots, G_k$, computable in time $O(|V(G)| + |E(G)|)$, such that $|V(G_i)| = O(|V(G)|)$ and $|E(G_i)| = O(|E(G)|)$ for every $i = 1, \dots, k$, and such that knowing $\text{inj}(H, G_1), \dots, \text{inj}(H, G_k)$ allows one to find $\text{hom}(H/P, G)$ for all partitions P of $V(H)$ in time $O(1)$. Furthermore, if G is $O(1)$-degenerate, then so are $G_1, \dots, G_k$.*

Proof of Lemma 6.1. The lemma follows easily by combining (2) and Lemma 5.2. Let $H_1, \dots, H_k$ be an enumeration of all quotient graphs of H, up to isomorphism. That is, $H_1, \dots, H_k$ are pairwise non-isomorphic and $\{H_1, \dots, H_k\} = \{H/P : P \in \mathcal{P}(H)\}$, where, as before, $\mathcal{P}(H)$ denotes the set of all partitions of $V(H)$.) Let G be a graph. By using (2) and "combining like terms", we get that

$$\text{inj}(H, G) = \sum_{P \in \mathcal{P}(H)} \mu_{\text{part}}(P) \cdot \text{hom}(H/P, G) = \sum_{i=1}^{k} \left(\sum_{\substack{P \in \mathcal{P}(H): \\ H/P \cong H_i}} \mu_{\text{part}}(P) \right) \cdot \text{hom}(H_i, G). \tag{26}$$

From (8) we know that $\mu_{\text{part}}(P) \neq 0$ for all $P \in \mathcal{P}(H)$, and that the sign of $\mu_{\text{part}}(P)$ depends only on the number of parts in P. In particular, if $P, Q \in \mathcal{P}(H)$ are such that $H/P \cong H/Q$, then $\mu_{\text{part}}(P)$ and $\mu_{\text{part}}(Q)$ have the same sign. Now, setting

$$c_i := \sum_{\substack{P \in \mathcal{P}(H): \\ H/P \cong H_i}} \mu_{\text{part}}(P),$$

we see that c_i is non-zero, since the summands in the above sum cannot cancel each other. With this notation, (26) becomes

$$\text{inj}(H, G) = \sum_{i=1}^{k} c_i \cdot \text{hom}(H_i, G). \tag{27}$$

Now the lemma immediately follows from (27) and Lemma 5.2, as every quotient graph of H is isomorphic to one of the graphs $H_1, \dots, H_k$. ∎

Proof of Lemma 1.7. The "if" part of the lemma follows from (2), and the "only-if" part of the lemma follows from Lemma 6.1. ∎

What can be said of the graph-family $\mathcal{H}$ appearing in Theorem 4? It turns out that $\mathcal{H}$ is hereditary and, furthermore, can be (explicitly) characterized by a *finite* collection of minimal forbidden induced subgraphs. In fact, every such minimal forbidden induced subgraph has at most 14 vertices. Evidently, a collection of forbidden induced subgraphs for $\mathcal{H}$ is the collection of all graphs F such that for some $k \geq 6$, the cycle C_k is (isomorphic to) a quotient graph of F. However, not all such graphs are minimal (with respect to not being in $\mathcal{H}$). Indeed, it is easy to see that for each $k \geq 5$, C_{k-2} is a quotient graph of C_k. This implies that if C_k is a quotient graph of a graph H, then so is C_{k-2}. Thus, it is enough to forbid graphs which have a quotient isomorphic to C_6 or C_7. Now, note that if a graph F has C_k as a quotient and is minimal with this property (with respect to containment), then F has at most $2k$ vertices. Hence, every minimal forbidden induced subgraph for $\mathcal{H}$ has at most 14 vertices.

7 Counting Induced Copies: Proof of Theorem 5 and Lemma 1.8

Here we prove Lemma 1.8. We again start with a more general lemma.

Lemma 7.1. *For every graph H there is $k = k(H)$ such that the following holds. For every graph G there are graphs $G_1, \dots, G_k$, computable in time $O(|V(G)| + |E(G)|)$, such that $|V(G_i)| = O(|V(G)|)$ and $|E(G_i)| = O(|E(G)|)$ for every $i = 1, \dots, k$, and such that the following holds:*

1. *Knowing* $\text{ind}(H, G_1), \dots, \text{ind}(H, G_k)$ *allows one to find* $\text{hom}(H', G)$ *for all supergraphs H' of H in time $O(1)$.*

2. *Knowing* $\text{hom}(H', G_i)$ *for every supergraph H' of H and every $1 \leq i \leq k$, allows one to find* $\text{inj}(H', G)$ *for all supergraphs H' of H in time $O(1)$.*

Furthermore, if G is $O(1)$-degenerate, then so are $G_1, \dots, G_k$.

Proof. We will show that there are $G_1, \dots, G_k$ which satisfy the assertion of each of the items 1-2 separately. One can then take the union of these two families of graphs to obtain the desired graphs $G_1, \dots, G_k$ for which both items hold.

Starting with Item 2, we begin by proving the following preliminary claim: for every supergraph H' of H and every partition P of $V(H)$, there is a supergraph H'' of H such that H'/P is an induced subgraph of H''. Indeed, fixing H' and P as above, we define H'' as follows: for each $\{U, V\} \in E(H'/P)$ (so $U, V \in P$), add to H all edges between U and V; the resulting graph is H''. It is easy to see that H'/P is indeed an induced subgraph of H'', as required.

Now let G be a graph. By combining the above claim with Lemma 5.1 (which we apply to all supergraphs H' of H at once), we see that there are graphs $G_1, \dots, G_k$ with $|V(G_i)| = O(|V(G)|)$ and $|E(G_i)| = O(|E(G)|)$ for every $i = 1, \dots, k$, such that knowing $\text{hom}(H', G_i)$ for every supergraph H' of H and every $1 \leq i \leq k$, allows one to find $\text{hom}(H'/P, G)$ for all supergraphs H' of H and all partitions P of $V(H)$ in time $O(1)$. With this information at hand, one can use (1) to compute $\text{inj}(H', G)$ for all supergraphs H' of H in time $O(1)$, as required.

We now move on the establish Item 1. For convenience, we denote by $\mathcal{E}$ the set of all subsets of $\binom{V(H)}{2} \setminus E(H)$, and by $\mathcal{P}$ the set of all partitions of $V(H)$. We start by observing that for every graph F,

$$
\begin{aligned}
\text{ind}(H, F) &= \sum_{E \in \mathcal{E}} (-1)^{|E|} \cdot \text{inj}(H \cup E, F) \\
&= \sum_{E \in \mathcal{E}} \sum_{P \in \mathcal{P}} (-1)^{|E|} \cdot \mu_{\text{part}}(P) \cdot \text{hom}((H \cup E)/P, F),
\end{aligned}
\tag{28}
$$

where the first equality uses (4) and the second equality uses (2). For an (unlabeled) graph H', put

$$
c_{H'} := \sum_{\substack{E \in \mathcal{E},\, P \in \mathcal{P}: \\ (H \cup E)/P \cong H'}} (-1)^{|E|} \cdot \mu_{\text{part}}(P). \tag{29}
$$

With this notation, (28) becomes

$$
\text{ind}(H, F) = \sum_{H' \colon c_{H'} \neq 0} c_{H'} \cdot \text{hom}(H', F).
$$

Let $H_1, \dots, H_k$ be an enumeration of all graphs H' such that $c_{H'} \neq 0$, up to isomorphism. In other words, $H_1, \dots, H_k$ are pairwise non-isomorphic and $\{H_1, \dots, H_k\} = \{H' : c_{H'} \neq 0\}$. For each $1 \leq i \leq k$, we put $c_i := c_{H_i}$. Finally, we can write

$$\mathrm{ind}(H, F) = \sum_{i=1}^{k} c_i \cdot \mathrm{hom}(H_i, F). \tag{30}$$

Crucially, observe that every supergraph of H is isomorphic to one of the graphs $H_1, \dots, H_k$. To see this, let H' be a (representative of the isomorphism class of a) supergraph of H. Clearly, if $E \in \mathcal{E}$ and $P \in \mathcal{P}$ are such that $(H \cup E)/P \cong H'$, then P is the partition of $V(H)$ into singletons, and $|E| = |E(H')| - |E(H)|$. Hence, all summands on the right-hand side of (29) have the same sign, implying that $c_{H'} \neq 0$. This in turn implies that H' is isomorphic to one of the graphs $H_1, \dots, H_k$, as required. Now Item 1 of the lemma immediately follows from (30) and Lemma 5.2. ■

Proof of Lemma 1.8. The implication $2 \Rightarrow 1$ follows from (4), the implication $3 \Rightarrow 2$ follows from Item 2 of Lemma 7.1, and the implication $1 \Rightarrow 3$ follows from Item 1 of Lemma 7.1. ■

Proof of Theorem 5. The "conversely"-part of the theorem follows immediately from Lemma 1.8 and Theorem 1. Let now $H \in \mathcal{H}^*$. If every supergraph of H is induced C_k-free for all $k \geq 6$, then the assertion of the theorem again follows from Lemma 1.8 and Theorem 1. Suppose then, by contradiction, that some supergraph H' of H contains an induced copy of C_k for some $k \geq 6$. It is easy to see that by adding a chord of this k-cycle, we obtain a supergraph H'' of H which contains an induced copy of C_6. This means that H contains an induced copy of some (not necessarily induced) spanning subgraph of C_6, in contradiction to the assumption that $H \in \mathcal{H}^*$. ■

8 Proof of Theorem 6

The "if" part of Theorem 6 follows from the following proposition.

Proposition 8.1. *Let H be a forest. Then for every graph G,* $\mathrm{hom}(H, G)$ *can be computed in expected time $O(|V(G)| + |E(G)|)$.*

Proof. First, observe that if H is a disconnected graph with connected components $H_1, \dots, H_k$, then $\mathrm{hom}(H, G) = \prod_{i=1}^{k} \mathrm{hom}(H_i, G)$ for every graph G. Thus, to prove the proposition, it suffices to consider the case that H is a tree.

Suppose then that H is a tree. To compute $\mathrm{hom}(H, G)$, we use dynamic programming to compute, for each subtree H' of H, $v \in V(H')$ and $x \in V(G)$, the number $N(H', v, x)$ of homomorphisms φ from H' to G such that $\varphi(v) = x$. To compute $N(H', v, x)$ for given H' and v (simultaneously for all $x \in V(G)$), we consider two cases according to whether or not v is a leaf of H'. Suppose first that v is a leaf of H', let u be the only neighbour of v in H', and set $H'' := H' - v$. Now, for each $x \in V(G)$, compute $N(H', v, x) = \sum_{y : \{x,y\} \in E(G)} N(H'', u, y)$ (where the sum is over all neighbours y of x). Suppose now that v is not a leaf, let $C_1, \dots, C_k$ be the connected components of the forest $H' - v$, and set $H'_i := H'[C_i \cup \{v\}]$. Now, for each $x \in V(G)$, compute $N(H', v, x) = \prod_{i=1}^{k} N(H'_i, v, x)$. Thus, in both cases, one can compute $N(H', v, x)$ for all $x \in V(G)$ by relying on counts for smaller trees which have already been computed and stored. By using perfect hashing [23], one can construct in expected linear time a hash table which allows writing and reading entries in time $O(1)$. So overall, this algorithm runs in (expected) time $O(|V(G)| + |E(G)|)$, as required. ■

Proof of Theorem 6. The "if" part of the theorem follows from Proposition 8.1. The "only if" part of the theorem follows from the combination of Lemmas 4.1 and 5.1 (as any non-forest graph evidently contains an induced cycle). ■

Acknowledgments We would like to thank an anonymous referee for kindly informing us that our methods can be used to derive Theorem 6.

References

[1] A. Abboud and V. Vassilevska Williams, Popular conjectures imply strong lower bounds for dynamic problems. In Proc. 55th Annual IEEE Symposium on Foundations of Computer Science, 2014.

[2] N. Alon, R. Yuster and U. Zwick, Finding and counting given length cycles, Algorithmica, 17(3):209-223, 1997.

[3] A. L. Barabási and R. Albert, Emergence of scaling in random networks, Science, 286(5439), 509-512, 1999.

[4] C. Beeri, R. Fagin, D. Maier, A. Mendelzon, J. Ullman and M. Yannakakis, Properties of acyclic database schemes. In Proceedings of the thirteenth annual ACM symposium on Theory of computing, pp. 355-362, 1981.

[5] C. Beeri, R. Fagin, D. Maier and M. Yannakakis, On the desirability of acyclic database schemes. Journal of the ACM (JACM), 30(3), pp. 479-513, 1983.

[6] A. Benson, D. F. Gleich and J. Leskovec, Higher-order organization of complex networks. Science, 353(6295):163-166, 2016.

[7] A. Björklund, T. Husfeldt, P. Kaski and M. Koivisto, Counting paths and packings in halves. In Proceedings of the 17th Annual European Symposium on Algorithms (ESA), 578-586, 2009.

[8] A. Björklund, P. Kaski and Ł. Kowalik, Counting thin subgraphs via packings faster than meet-in-the-middle time, ACM Transactions on Algorithms (TALG), 13(4), 1-26, 2017.

[9] S. K. Bera, N. Pashanasangi and C. Seshadhri, Linear Time Subgraph Counting, Graph Degeneracy, and the Chasm at Size Six. In Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS 2020), 38:1-38:20.

[10] J. Brault-Baron. Hypergraph acyclicity revisited. ACM Computing Surveys (CSUR), 49(3), pp. 1-26, 2016.

[11] M. Bressan, Faster subgraph counting in sparse graphs, arXiv preprint arXiv:1805.02089 (2018).

[12] R. Burt, Structural holes and good ideas, American Journal of Sociology, 110(2):349-399, 2004.

[13] H. Chen and S. Mengel. Counting answers to existential positive queries: A complexity classification. In Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 315-326, 2016.

[14] N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. SIAM Journal on computing, 14(1), pp. 210-223, 1985.

[15] F. R. K. Chung, R. L. Graham and R. M. Wilson, Quasi-random graphs, Combinatorica 9(1989), 345-362.

[16] R. Curticapean, H. Dell and D. Marx, Homomorphisms are a good basis for counting small subgraphs. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pages 210-223, 2017.

[17] R. Curticapean and D. Marx, Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts, In Proc. 55th Annual IEEE Symposium on Foundations of Computer Science, 130-139, 2014.

[18] M. Dalirrooyfard, T. D. Vuong and V. Vassilevska Williams, Graph pattern detection: Hardness for all induced patterns and faster non-induced cycles. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pp. 1167-1178, 2019.

[19] V. Dalmau and P. Jonsson, The complexity of counting homomorphisms seen from the other side. Theoretical Computer Science, 329(1-3), 315-323, 2004.

[20] F. Eisenbrand and F. Grandoni, On the complexity of fixed parameter clique and dominating set, Theoretical Computer Science, 326(1-3):57-67, 2004.

[21] P. Erdős, L. Lovász and J. Spencer, Strong independence of graphcopy functions, Graph Theory and Related Topics, 165-172, 1979.

[22] J. Flum and M. Grohe, The parameterized complexity of counting problems, in: Proc. 43rd IEEE Symposium on Foundations of Computer Science, 2002, pp. 538–547.

[23] M. L. Fredman, J. Komlós, and E. Szemerédi, Storing a sparse table with $O(1)$ worst case access time, Journal of the ACM (JACM), 31(3), pp. 538-544, 1984.

[24] P. Holland and S. Leinhardt, A method for detecting structure in sociometric data, American Journal of Sociology, 76:492-513, 1970.

[25] F. Hormozdiari, P. Berenbrink, N. Prulj and S. Cenk Sahinalp, Not all scale-free networks are born equal: The role of the seed graph in ppi network evolution, PLoS Computational Biology, 118, 2007.

[26] A. Itai and M. Rodeh, Finding a minimum circuit in a graph, SIAM Journal on Computing, 7(4):413-423, 1978.

[27] A. Khan, N. Li, X. Yan, Z. Guan, S. Chakraborty and S. Tao, Neighborhood based fast graph search in large networks. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, 2011.

[28] T. Kloks, D. Kratsch and H. Müller, Finding and counting small induced subgraphs efficiently, Information Processing Letters, 74(3):115-121, 2000.

[29] M. Kowaluk, A. Lingas and E. M. Lundell, Counting and detecting small subgraphs via equations, SIAM Journal on Discrete Mathematics, 27(2):892-909, 2013.

[30] L. Lovász, **Large networks and graph limits**, 2012, Providence: American Mathematical Society.

[31] L. Lovász and V. T. Sós, Generalized quasirandom graphs, J. Combin. Theory B 98 (2008), 146-163.

[32] D. W. Matula and L. L. Beck, Smallest-last ordering and clustering and graph coloring algorithms, Journal of the ACM (JACM), 30(3), pp. 417-427, 1983.

[33] A. McGregor, S. Vorotnikova and H. T. Vu, Better algorithms for counting triangles in data streams. In Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 401-411, 2016.

[34] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon, Network motifs: simple building blocks of complex networks, Science, 298(5594), 824-827, 2002.

[35] J. Nešetřil and P. O. De Mendez, **Sparsity: graphs, structures, and algorithms** (Vol. 28), Springer Science & Business Media, 2012.

[36] J. Nešetřil and S. Poljak, On the complexity of the subgraph problem, Commentationes Mathematicae Universitatis Carolinae, 26(2):415-419, 1985.

[37] N. Przulj, Biological network comparison using graphlet degree distribution, Bioinformatics, 23(2):177-183, 2007.

[38] N. Przulj, D. G. Corneil and I. Jurisica, Modeling interactome: scale-free or geometric?, Bioinformatics, 20(18):3508-3515, 2004.

[39] S. Rudich and A. Wigderson, **Computational complexity theory**, (Vol. 10), American Mathematical Soc., 2004.

[40] A. E. Sariyuce, C. Seshadhri, A. Pinar and U. V. Catalyurek, Finding the hierarchy of dense subgraphs using nucleus decompositions. In Proceedings, International World Wide Web Conference (WWW), 927-937, 2015.

[41] C. E. Tsourakakis, The k-clique densest subgraph problem. In Proceedings, International World Wide Web Conference (WWW), 1122-1132, 2015.

[42] J. Ugander, L. Backstrom and J. M. Kleinberg, Subgraph frequencies: mapping the empirical and extremal geography of large graph collections. In Proceedings, International World Wide Web Conference (WWW), 1307-1318, 2013.

[43] J. H. Van Lint and R. M. Wilson, **A course in combinatorics**, 2001, Cambridge university press.

[44] V. Vassilevska Williams, Efficient algorithms for clique problems, Information Processing Letters, 109(4):254-257, 2009.

[45] V. Vassilevska Williams, Multiplying matrices faster than Coppersmith-Winograd. In Proceedings of the 44th Symposium on Theory of Computing Conference (STOC), 887-898, 2012.

[46] V. Vassilevska Williams and R. Williams, Finding, minimizing, and counting weighted subgraphs. In Proc. 41st Annual ACM Symposium on the Theory of Computing, 455-464, 2009.

A Proof of Lemma 5.2

We start by recalling the definition of the tensor product of graphs. The tensor product of graphs G_1 and G_2, denoted $G_1 \times G_2$ has vertex set $V(G_1 \times G_2) = V(G_1) \times V(G_2)$ and edge-set

$$E(G_1 \times G_2) = \{\{(x_1, x_2), (y_1, y_2)\} : \{x_1, y_1\} \in E(G_1) \text{ and } \{x_2, y_2\} \in E(G_2)\}.$$

A key property of the tensor product is that the parameter $\hom(H, \cdot)$ is multiplicative with respect to it (for any graph H). That is, for every pair of graphs G_1, G_2, it holds that

$$\hom(H, G_1 \times G_2) = \hom(H, G_1) \cdot \hom(H, G_2). \tag{31}$$

To see that (31) holds, simply observe that for functions $\varphi_i : V(H) \to V(G_i)$ (where $1 \leq i \leq 2$), the function $v \mapsto (\varphi_1(v), \varphi_2(v))$ is a homomorphism from H to $G_1 \times G_2$ if and only if φ_i is a homomorphism from H to G_i for each $1 \leq i \leq 2$. In what follows, we will use the following (trivial) observation regarding tensor products and degeneracy.

Observation A.1. *Let F, G be graphs. If G is κ-degenerate, then $F \times G$ is $(v(F) \cdot \kappa)$-degenerate.*

Proof. It is easy to see that for each $x \in V(F)$ and $y \in V(G)$, the degree of (x, y) in $F \times G$ is $d_{F \times G}((x, y)) = d_F(x) \cdot d_G(y) < v(F) \cdot d_G(y)$. It follows that every subgraph of $F \times G$ contains a vertex of degree at most $v(F) \cdot \kappa$ (since G is κ-degenerate). ■

We now state a lemma of Erdős, Lovász and Spencer [21] (see also Proposition 5.44(b) in [30]), which will play a crucial role in the proof of Lemma 5.2.

Lemma A.2 ([30]). *Let $H_1, \dots, H_k$ be pairwise non-isomorphic graphs, and let $c_1, \dots, c_k \neq 0$ be non-zero constants. Then there exist graphs $F_1, \dots, F_k$ such that the $k \times k$ matrix $M_{i,j} = c_j \cdot \hom(H_j, F_i)$, $1 \leq i, j \leq k$, is invertible.*

Finally, we are ready to prove Lemma 5.2.

Proof of Lemma 5.2. Let G be a graph. By Lemma A.2, there are graphs $F_1, \dots, F_k$ such that the $k \times k$ matrix $M_{i,j} := c_j \cdot \hom(H_j, F_i)$ $(1 \leq i, j \leq k)$ is invertible. For each $1 \leq i \leq k$, we set $G_i := F_i \times G$ and $b_i := c_1 \cdot \hom(H_1, G_i) + \dots + c_k \cdot \hom(H_k, G_i)$, observing that

$$b_i = \sum_{j=1}^{k} c_j \cdot \hom(H_j, F_i \times G) = \sum_{j=1}^{k} c_j \cdot \hom(H_j, F_i) \cdot \hom(H_j, G) = \sum_{j=1}^{k} M_{i,j} \cdot \hom(H_j, G). \tag{32}$$

In the third equality above, we used (31). We will treat (32) $(1 \leq i \leq k)$ as a system of linear equations, where $\hom(H_1, G), \dots, \hom(H_k, G)$ are the variables, M is the matrix of the system, and $b_1, \dots, b_k$ are the constant terms. Since M is invertible (as guaranteed by our choice of $F_1, \dots, F_k$), knowing $b_1, \dots, b_k$ indeed enables us to find $\hom(H_1, G), \dots, \hom(H_k, G)$ in time $O(1)$, as required. To complete the proof, we note that $|V(G_i)| = |V(G)| \cdot |V(F_i)| = O(|V(G)|)$ and $|E(G_i)| \leq |E(G)| \cdot |E(F_i)|^2 = O(|E(G)|)$ for every $1 \leq i \leq k$, and that if G is $O(1)$-degenerate then so are $G_1, \dots, G_k$ by Observation A.1. ■