

Parameterized Complexity of INDUCED H -MATCHING on Claw-Free Graphs*

Danny Hermelin[†] Matthias Mnich[‡] Erik Jan van Leeuwen[§]

Abstract

The INDUCED H -MATCHING problem asks to find k disjoint, induced subgraphs isomorphic to H in a given graph G such that there are no edges between vertices of different subgraphs. This problem generalizes the classical INDEPENDENT SET and INDUCED MATCHING problems, among several other problems. We show that INDUCED H -MATCHING is fixed-parameter tractable in k on claw-free graphs when H is a fixed connected graph of constant size, and even admits a polynomial kernel when H is a clique. Both results rely on a new, strong algorithmic structure theorem for claw-free graphs. To show the fixed-parameter tractability of the problem, we additionally apply the color-coding technique in a novel way. Complementing the above two positive results, we prove the W[1]-hardness of INDUCED H -MATCHING for graphs excluding $K_{1,4}$ as an induced subgraph. In particular, we show that INDEPENDENT SET is W[1]-hard on $K_{1,4}$ -free graphs.

1 Introduction

A graph is *claw-free* if no vertex in the graph has three pairwise nonadjacent neighbors, i.e. if it does not contain a copy of $K_{1,3}$ as an induced subgraph. The class of claw-free graphs contains several well-studied graph classes such as line graphs, unit interval graphs, de Bruijn graphs, the complements of triangle-free graphs, and graphs of several polyhedra and polytopes. Consequently, claw-free graphs have attracted much interest, and are by now the subject of numerous research papers (see e.g. the survey [9]).

Our understanding of claw-free graphs and their structure was greatly extended with the recently developed theory of Chudnovsky and Seymour.

*This work is partially supported by the ERC StG project PAAI no. 259515.

[†]Max Planck Institute for Informatics, Saarbrücken, Germany, hermelin@mpi-inf.mpg.de

[‡]Cluster of Excellence “Multimodal Computing and Interaction”, Saarland University, Saarbrücken, Germany, mmnich@mmci.uni-saarland.de

[§]Dept. Computer, Control, and Management Engineering “Antonio Ruberti”, Sapienza University of Rome, Italy, E.J.van.Leeuwen@dis.uniroma1.it

This highly technical and detailed claw-free structure theory is contained in a sequence of seven papers called Claw-free graphs I–VII (see [4] for an accessible survey), and culminates in several variants of *claw-free decomposition theorems*. Each of these decomposition theorems shows that every connected claw-free graph can be constructed by starting with a collection of “basic claw-free graphs”, and then gluing these basic graphs together in some controlled manner.

Recently, Chudnovsky and Seymour’s claw-free structure theory has been deployed for algorithmic purposes. For example, it was used to develop approximation algorithms for CHROMATIC NUMBER on quasi-line graphs [3, 18]. Faenza et al. [8] gave an algorithm to find a decomposition of claw-free graphs, and used this algorithm to improve on the fastest previously known algorithm for INDEPENDENT SET on claw-free graphs. Their decomposition, although similar in nature, does not rely on Chudnovsky and Seymour’s structure theorem. Independently, the authors of this paper, along with Woeginger, gave an algorithm for the Chudnovsky-Seymour decomposition theorem, and used this to show that DOMINATING SET is fixed-parameter tractable on claw-free graphs, and admits a polynomial kernel as well [16]. (The fixed-parameter tractability result was also shown independently by Cygan et al. [5] using different methods.) Recently, Golovach et al. [14] used the algorithmic decomposition theorem in [16] to give a parameterized algorithm for the INDUCED DISJOINT PATHS problem on claw-free graphs.

In this paper we extend this line of research by considering the INDUCED H -MATCHING problem on claw-free graphs. This problem generalizes several important and well-studied problems, such as INDEPENDENT SET and INDUCED MATCHING. It can be defined as follows. We are given a claw-free graph G and an integer k , and the goal is to determine whether there is an induced occurrence of $k \cdot H$ in G ; that is, whether there is a set of induced subgraphs $M = \{H_1, \dots, H_k\}$ in G , each isomorphic to H , that are pairwise vertex-disjoint and satisfy that $(u, v) \notin E(G)$ for any pair of vertices $u \in V(H_i)$ and $v \in V(H_j)$ with $i \neq j$. As discussed further below, the study of this problem among other things requires a significantly stronger decomposition theorem for claw-free graphs compared to those developed previously.

The INDUCED H -MATCHING problem is closely related to the equally well-studied H -MATCHING problem. In H -MATCHING, the goal is to find induced subgraphs H_1, \dots, H_k isomorphic to H that are pairwise vertex-disjoint. Observe that when H has minimum degree at least 3, H -MATCHING reduces to INDUCED H -MATCHING by subdividing all the edges of G and H . On the other hand, on line graphs (a subclass of claw-free graphs), a reduction exists in the other direction. Recall that the *line graph* $L(G)$ of a graph G consists of a vertex v_e for each $e \in E(G)$ and there is an edge (v_e, v_f) in $L(G)$ if and only if e and f are incident to the same vertex in G . The graph G is known as the *pre-image* of $L(G)$. It can be easily seen that there is a

bijection between the set of induced H -matchings on $L(G)$ and the set of H -matchings on G . Hence INDUCED H -MATCHING on line graphs inherits all known complexity results on H -MATCHING on general graphs. In particular, this implies that INDUCED H -MATCHING is NP-complete on line graphs if H contains a connected component with at least one edge [12, 19, 21].

In the context of parameterized complexity, the H -MATCHING problem has received significant attention. In particular, the problem is known to be fixed-parameter tractable by the size of the matching k [11, 20] and even has a polynomial kernel (when H is fixed) [23]. Recently, tight lower bounds on the kernel size for specific graphs H were obtained in [6, 17]. Note again that these results immediately carry over to INDUCED H -MATCHING on line graphs.

In general graphs, the INDUCED H -MATCHING problem is W[1]-hard for any clique H on $h \geq 1$ vertices when parameterized by the matching size k [7, 24]. Marx [22] showed that the INDUCED H -MATCHING for a graph H on a single vertex (i.e. the INDEPENDENT SET problem) is W[1]-hard on $K_{1,5}$ -free graphs. Another related result by Cameron and Hell [2] shows that on certain graph classes the problem can be reduced to an instance of INDEPENDENT SET on a graph in that same graph class, provided that the set of all occurrences of H are given. Their results, however, do not apply to claw-free graphs.

Our Results. The main result of this paper is that INDUCED H -MATCHING is fixed-parameter tractable on claw-free graphs when parameterized by k for fixed connected graphs H of constant size. It is important to note that requiring H to be of constant size is essential, since the problem is W[1]-hard with respect to $k + |V(H)|$ even for line graphs and co-bipartite graphs [13, 14]. In the special case that H is a fixed clique, we also show that the problem admits a polynomial kernel. In contrast, we prove that the problem becomes W[1]-hard on $K_{1,4}$ -free graphs. These results both complement and tighten the above-mentioned known hardness results on the problem.

Techniques. The main difference between this work and previous works on algorithms using the Chudnosky-Seymour structure theorem is that the so-called *twins* and *W-joins* seem hard to get rid of in INDUCED H -MATCHING. Hence we require a stronger version of the previously proposed decomposition theorems, one that does not require that such structures are not present in the given claw-free graph. We provide such a decomposition theorem in Section 2, and show that the decomposition it implies can be computed in polynomial time.

The new decomposition theorem that we develop has the advantage that it is simpler to state. In particular, it decomposes the input claw-free graph

into only two graph classes. Yet, for one of these graph classes, namely the class of fuzzy circular-arc graphs (a superclass of proper circular-arc graphs), the INDUCED H -MATCHING problem was not previously known to be polynomial-time solvable. We show that in fact it is, and in the appendix we exhibit stronger results on circular-arc graphs and give evidence why these results are tight.

After obtaining our refined decomposition theorem, we solve INDUCED H -MATCHING on claw-free graphs by applying the color-coding technique [1] in a novel way. To give some intuition behind this approach, we recall that to solve INDUCED H -MATCHING on line graphs G and H , we need to find pairwise vertex-disjoint induced subgraphs H_1, \dots, H_k in the pre-image of G such that H_i is isomorphic to the pre-image of H . In particular, we can find these H_i in our structural decomposition for G . However, if G and H are claw-free, there is no notion of pre-image and therefore no immediate relation between H and the H_i that we would want to find. Instead, we show that it is sufficient to find, in a coordinated manner and using color-coding, isomorphic copies of H itself. Thus, in some sense, color-coding allows us to reduce INDUCED H -MATCHING to H -MATCHING. Complete details of this are given in Section 4.

For obtaining our polynomial kernel when H is a fixed clique of order h , we reduce the size of the decomposition-structure to $O(h^4 k^2)$. We then construct a kernel that mimics an easy algorithm for INDUCED H -MATCHING on such reduced decomposition-structures. This kernel actually reduces the problem to an equivalent instance of WEIGHTED INDEPENDENT SET of size polynomial in k , which in turn can be reduced to an equivalent instance of INDUCED H -MATCHING on claw-free graphs. The technique that we use here substantially simplifies the technique that was used before to obtain a polynomial kernel for DOMINATING SET on claw-free graphs [16].

2 Preliminaries

Let G be a graph. We call $I \subseteq V(G)$ an *independent set* if no two vertices of I are adjacent, and use $\alpha(G)$ to denote the size of a maximum independent set of G . The neighborhood of a vertex $v \in V(G)$ is denoted by $N(v) = \{u \in V(G) \mid (u, v) \in E(G)\}$, and the closed neighborhood of v is $N[v] = N(v) \cup \{v\}$. This notation extends to subsets of vertices. Given $X \subseteq V(G)$, the subgraph induced by X is $G[X] = (X, E(G) \cap (X \times X))$, and the graph $G - X$ is the subgraph induced by $V(G) \setminus X$. Then G is *claw-free* if $\alpha(G[N(v)]) \leq 2$ for any $v \in V(G)$.

We next define a variant of circular-arc graphs that will play an important role in the paper. Given a set \mathcal{A} of objects, a graph G is the *intersection graph* of \mathcal{A} if each vertex corresponds to an object of \mathcal{A} and there is an edge between two vertices if and only if the corresponding objects intersect. In

a *fuzzy intersection graph* G of \mathcal{A} , there is an edge between two vertices if the corresponding objects intersect in more than one point, and there is no edge if the corresponding objects do not intersect. The fuzziness stems from what happens if the two objects intersect in precisely one point: the graph may have an edge or not. We call \mathcal{A} a *representation* of G . Now consider a set of arcs on a circle (throughout, we assume that arcs are not a single point). Then the set is called *proper* if no arc is a subset of another, *almost proper* if any arc that is a subset of another is in fact equal to the other, and *almost strict* if, for any maximal set of arcs with the same endpoints, at most one of these endpoints is also an endpoint of an arc outside the set. A graph is a *proper circular-arc graph* if it is the intersection graph of a set of proper arcs on a circle. A graph is a *fuzzy circular-arc graph* if it is the fuzzy intersection graph of an almost-proper, almost-strict set of arcs on a circle. Note that any intersection graph of an almost-proper set of arcs on a circle is a proper circular-arc graph; hence a fuzzy circular-arc graph without any fuzziness is just a proper circular-arc graph. It is known that any fuzzy circular-arc graph is claw free.

A *hypergraph* \mathcal{R} consists of a set of vertices $V(\mathcal{R})$ and a multi-set of edges $E(\mathcal{R})$ such that each $e \in E(\mathcal{R})$ is a subset of $V(\mathcal{R})$. We explicitly allow empty edges. Note that if $|e| \leq 2$ for every $e \in E(\mathcal{R})$, then \mathcal{R} is a multi-graph, possibly with self-loops. A *strip-structure* of a graph G consists of a hypergraph \mathcal{R} and for each edge $e \in E(\mathcal{R})$ a *strip* (J_e, Z_e) such that

- for each $e \in E(\mathcal{R})$, J_e is a claw-free graph and $Z_e \subseteq V(J_e)$ is an independent set;
- $V(J_e) \setminus Z_e \neq \emptyset$ for any $e \in E(\mathcal{R})$, and the sets $V(J_e) \setminus Z_e$ over all $e \in E(\mathcal{R})$ partition $V(G)$;
- for each $e \in E(\mathcal{R})$ and for each $r \in e$, there is a unique $z_e^r \in Z_e$ and $Z = \bigcup_{r \in e} z_e^r$;
- for each $r \in V(\mathcal{R})$, the union of $N(z_e^r)$ over all $e \in E(\mathcal{R})$ for which $r \in e$ induces a clique $C(r)$ in G ;
- if $u, v \in V(G)$ are adjacent, then $u, v \in V(J_e)$ for some $e \in E(\mathcal{R})$ or $u, v \in C(r)$ for some $r \in V(\mathcal{R})$;
- $J_e[V(J_e) \setminus Z_e]$ equals $G[V(J_e) \setminus Z_e]$.

Given a strip (J, Z) , the set $N(z)$ for $z \in Z$ is referred to as a *boundary* of (J, Z) , while the (possibly empty) graph $J - N[Z]$ is called the *interior* of (J, Z) . We call (J, Z) a *spot* if J is a three-vertex path and Z consists of its ends, and a *stripe* if no vertex of J is adjacent to more than one $z \in Z$.

We prove the following theorem in the appendix.

Theorem 2.1 *Let G be a connected claw-free graph that is not a fuzzy circular-arc graph and has $\alpha(G) > 4$. Then G admits a strip-structure such that each strip is either a spot, or a stripe (J, Z) with $1 \leq |Z| \leq 2$ that is a fuzzy circular-arc graph or satisfies $\alpha(J) \leq 4$. Moreover, we can find such a strip-structure in polynomial time.*

3 (Fuzzy) Circular-Arc Graphs

In this section, we consider the complexity of INDUCED H -MATCHING on (fuzzy) circular-arc graphs.

3.1 Circular-Arc Graphs

In this subsection, we consider the complexity of INDUCED H -MATCHING on circular-arc graphs. These results generalize results by Heggernes et al. [15] on INDUCED SUBGRAPH ISOMORPHISM on interval graphs. Recall that INDUCED SUBGRAPH ISOMORPHISM is the problem that asks, given two graph G and H , whether G contains an induced subgraph which is isomorphic to H .

Recall that a (*proper*) *circular-arc graph* is the intersection graph of a set of (proper) arcs on a circle. If the union of the arcs does not cover the circle, then the graph is a (*proper*) *interval graph*.

Theorem 3.1 *INDUCED H -MATCHING on a circular-arc graph G and a connected proper interval graph H can be solved in polynomial time, even if G , H , and k are part of the input.*

Proof: We first show that we can enumerate all occurrences of H in G . To this end, we fix a representation of G . Now consider all possibilities for the leftmost and the rightmost vertex of H . That is, we guess which vertex l_H of H will have the leftmost left endpoint in an isomorphism of H into an induced subgraph of G , and we guess the vertex r_H having the rightmost right endpoint as well. Similarly, we consider all possibilities for the vertex l_G of G to which l_H can be mapped, and all possibilities for the vertex r_G of G to which r_H can be mapped. This can be achieved by simply trying all possible pairs of vertices of G and H . Now remove all vertices that have their left endpoint to the left of l_G and all vertices that have their right endpoint to the right of r_G . Finally, create four new cliques of size $1 + \max\{|V(G)|, |V(H)|\}$ and add them as neighbors of l_H , r_H , l_G , and r_G . Call the resulting graphs G' and H' . Note that G' has to be an interval graph and that H' has to be a proper interval graph. It remains to verify that G' contains an induced subgraph isomorphic to H' . This can be done in polynomial time using a result of Heggernes et al. [15]. In this way, we find all possible occurrences of H as an induced subgraph of G .

Given all occurrences of H in G (note that there are polynomially many), we can use a result of Cameron and Hell [2] to reduce the problem to an INDEPENDENT SET problem on a circular-arc graph. Since INDEPENDENT SET can be solved in polynomial time on circular-arc graphs, we obtain the polynomial-time algorithm claimed in the theorem statement. \square

The following result follows from a simple extension of a similar result by Heggernes et al. [15] on interval graphs and connected proper interval graphs. Essentially, we need to 'cut open' G and H at a maximal clique to reduce the theorem statement to their result.

Theorem 3.2 INDUCED SUBGRAPH ISOMORPHISM *on a circular-arc graph G and a connected proper circular-arc graph H can be solved in polynomial time, even if both G and H are part of the input.*

Proof: We can determine in linear time whether G is an interval graph. If so, then H has to be a proper interval graph.

If H is a proper interval graph, then we can just use the method described in the proof of Theorem 3.1 above to find an induced subgraph isomorphic to H .

If H is not a proper interval graph, then G must be a circular-arc graph. Suppose that G indeed contains an induced subgraph isomorphic to H (by abusing notation, we call this subgraph H as well). Then, similar as in Theorem 3.1 of [15], there exists a maximal clique K_G in G and a maximal clique K_H in H such that K_H is mapped to a subset of K_G and no other vertices of H are mapped to K_G . We guess these maximal cliques K_G and K_H . Find a point x_G such that the intervals of G containing x_G correspond exactly to K_G , and find a point x_H such that the intervals of H containing x_H correspond exactly to K_H . Now remove an infinitesimally small part of the arcs around x_G and x_H . Then, as in the proof of Theorem 3.1 above, add cliques of size $1 + \max\{|V(G)|, |V(H)|\}$ on both sides of these two gaps as an anchor. Call the resulting graphs G' and H' . Note that G' has to be an interval graph and that H' has to be a proper interval graph. It remains to verify that G' contains an induced subgraph isomorphic to H' . This can be done in polynomial time using a result of Heggernes et al. [15]. \square

Theorem 3.3 INDUCED H -MATCHING *on circular-arc graphs G and connected proper circular-arc graphs H can be solved in polynomial time, even if both G and H are part of the input.*

Proof: We can check in linear time whether H is a proper interval graph. If so, then we can solve the problem in polynomial time by using Theorem 3.1. Hence we may assume that H is not a proper interval graph. Then, clearly, G can have an induced H -matching of size at most one. This can be checked by verifying whether G has an induced subgraph isomorphic to H by using Theorem 3.2. \square

Theorem 3.4 INDUCED H -MATCHING on circular-arc graphs G and proper circular-arc graphs H is FPT when parameterized by k and the number of connected components of H .

Proof: If H is connected, then the result follows immediately from Theorem 3.3. If H is not connected, then H has to be a proper interval graph. But then we can cut open G . Guess where and reduce G to an interval graph. Then create H' as k copies of H , and find H' as an induced subgraph in G' using the algorithm of Heggenes et al. [15]. \square

Note that INDUCED H -MATCHING is NP-hard if both G and H are a proper interval graph, even if $k = 1$ [15]. Hence parameterization by some property of H is really necessary in Theorem 3.4.

It is open whether parameterization by k is necessary, i.e. whether INDUCED H -MATCHING is FPT when parameterized by $|V(H)|$ and k is part of the input. The approaches presented above seem to fail. For example, even if all occurrences of H in G can be found in FPT time, we would then be required to solve an instance of INDEPENDENT SET on a multi-interval graph, which is known to be W[1]-hard [10]. Note that if H is assumed to be fixed, then problem is polynomial-time solvable due to a result by Cameron and Hell [2].

3.2 Induced H -Matching on Fuzzy Circular-Arc Graphs

A graph G is *co-bipartite* if it is the complement of a bipartite graph.

Theorem 3.5 ([13]) INDUCED H -MATCHING is NP-hard and W[1]-hard when parameterized by $|V(H)|$, even if $k = 1$ and both G and H are connected co-bipartite graphs.

Observe that any co-bipartite graph can be obtained as a thickening of a single semi-edge. Moreover, the graph consisting of a single semi-edge is a circular interval trigraph. Hence any co-bipartite graph is a fuzzy circular-arc graph and Theorem 3.5 immediately gives the following corollary.

Corollary 3.6 INDUCED H -MATCHING is NP-hard, and W[1]-hard when parameterized by $|V(H)|$, even if $k = 1$ and both G and H are connected fuzzy circular-arc graphs or connected claw-free graphs.

Hence, there is little hope of generalizing Theorem 3.3 or Theorem 3.4 to fuzzy circular-arc graphs. Instead, we consider the case when H is fixed.

Theorem 3.7 INDUCED H -MATCHING on a fuzzy circular-arc graph G can be solved in polynomial time when H is a fixed, connected graph.

Proof: We generalize the approach of Theorem 3.1. First, observe that since H is fixed, we can find the set \mathcal{H} of all occurrences of H in G in polynomial time. We call two occurrences of H *compatible* if no vertex of one occurrence is a neighbor of a vertex of the other.

We may assume that H occurs at least once in G . Pick an arbitrary $H^* \in \mathcal{H}$, and remove $N[H^*]$ from G . Since fuzzy circular-arc graphs are closed under deletion, the resulting graph is also a fuzzy circular-arc graph, which we call G as well by abusing notation.

Now find a representation of G , using the algorithm of Oriolo, Pietropaolo, and Stauffer [25]. This gives a representation \mathcal{A} of G as described in Lemma A.2. For each $H' \in \mathcal{H}$, let $l(H')$ (resp. $r(H')$) be the leftmost (resp. rightmost) endpoint of any arc of H' . Let \mathcal{P} be the set of endpoints of the arcs of \mathcal{A} . For each point $p \in \mathcal{P}$, let \mathcal{H}_p be the set of all $H' \in \mathcal{H}$ for which $r(H') = p$. Finally, since we removed one occurrence of H from G , we can uniquely order \mathcal{P} from left to right as $p_1, \dots, p_{|\mathcal{P}|}$. For each $p \in \mathcal{P}$, we order the elements of \mathcal{H}_p arbitrarily.

We describe a dynamic programming approach that solves INDUCED H -MATCHING. For each $i = 1, \dots, |\mathcal{P}|$ and for each $j = 1, \dots, |\mathcal{H}_{p_i}|$, let

$$M[i, j] = \max \left\{ \begin{array}{l} \max_{j^*=1, \dots, |\mathcal{H}_{p_{i-1}}|} M[i-1, j^*] \\ 1 + \max_{1 \leq i^* < i} \max_{\substack{j^*=1, \dots, |\mathcal{H}_{p_{i^*}}| \\ \mathcal{H}_{p_i}^j, \mathcal{H}_{p_{i^*}}^{j^*} \text{ compatible}}} M[i^*, j^*] \end{array} \right\}$$

For simplicity, we let $M[0, 1] = 0$. Then the result follows by computing

$$\max_{j=1, \dots, |\mathcal{H}_{p_{|\mathcal{P}|}}|} \{M[|\mathcal{P}|, j]\}.$$

By running this dynamic programming algorithm for all possible choices of H^* , we obtain a polynomial-time algorithm (when H is fixed) to solve INDUCED H -MATCHING. \square

4 Fixed-Parameter Algorithm

In this section, we prove the following theorem.

Theorem 4.1 *Let H be a fixed connected graph. Then INDUCED H -MATCHING is fixed-parameter tractable on claw-free graphs when parameterized by the size of the matching k .*

Throughout the section, we use H to denote a fixed connected graph of order h , and G to denote a claw-free graph on n vertices given as input to the INDUCED H -MATCHING problem. Observe that H is part of the problem description rather than part of the problem instance, and so its

size is constant. We may assume that G is connected as the extension to the disconnected case is immediate.

The algorithm we use to prove Theorem 4.1 deploys the claw-free decomposition theorem stated in Theorem 2.1. Note that in order to apply this decomposition theorem, we need to first handle two cases: the case where $\alpha(G)$ is small, and the case where G is a fuzzy circular-arc graph. The latter is handled by Theorem 3.7, and the former is easily handled as follows.

Lemma 4.2 INDUCED H -MATCHING *on graphs G with $\alpha(G) \leq 4$ can be solved in polynomial time.*

This lemma is immediate from the observation that if $\alpha(G) \leq 4$, then any induced H -matching of G will have size at most 4.

4.1 Guessing the Induced Strip-Graph

The first step of our algorithm in case G is not a fuzzy circular-arc graph and has $\alpha(G) > 4$ is to apply Theorem 2.1 to obtain a strip-structure of G .

Assume that G has an induced H -matching M of size k . Our goal from now on will be to discover M . We will call an edge of \mathcal{R} *covered* if its corresponding strip contains a vertex of M . Similarly, we call a vertex r of \mathcal{R} *covered* if the clique $C(r)$ corresponding to r contains a vertex of M . With these definitions, we can talk about the subgraph of \mathcal{R} induced by covered edges and vertices, which we call the *covered subgraph* of \mathcal{R} .

As a first step, we guess what the covered subgraph of \mathcal{R} looks like. Obviously, we cannot guess which vertices of \mathcal{R} will be in the covered subgraph, as this would yield an $n^{O(hk)}$ -time algorithm. However, we can guess the graph that the covered subgraph will be isomorphic to, which we will call the *base graph*. To bound the size of the base graph, note that each edge of \mathcal{R} corresponds to at least one vertex of an occurrence of H . Hence the base graph has at most hk edges and at most $2hk$ vertices. As the base graph can be seen as a multigraph, there are at most $2^{O((hk)^3)}$ possibilities for this graph. (A tighter bound might be obtainable here, but the above bound is sufficient for our purposes.) Let B denote the base graph induced by M . Since we go through all possible choices of the base graph, we can assume that our guess is correct and isomorphic to B .

As a second step, we guess how the k occurrences of H are ‘spread’ over B . That is, for each edge e of B , we guess exactly what the intersection of M with the hypothetical strip corresponding to e would look like. In particular, we need to guess the intersection of M with the boundaries of the strip. We call this the *B -assignment*. To this end, we simply try all possible assignments of the hk vertices of M to the edges of B . For each edge of B , we have one bucket to indicate that the vertex should be in the interior of the strip and a bucket for each boundary of the strip. If we guess

that the strip is a spot, then there is only a single bucket. Since there are at most three buckets per strip, this takes $O((3hk)^{hk})$ time. (Again, a better bound may be possible here, but this bound is sufficient for our purposes.)

For each assignment, we verify whether it is proper. That is, we need to verify whether the assignment could possibly correspond to an induced H -matching of size k . First, if two vertices of H are adjacent, then in each occurrence of H they must be assigned to the same edge of B or to two different edges of B that share an endpoint. In the latter case, they must also be assigned to the boundary buckets corresponding to that endpoint. Secondly, for each vertex b of B , the boundary buckets corresponding to b of the edges incident to b can contain vertices of at most one occurrence of H . Moreover, those vertices must induce a clique in H .

As we go through all possible B -assignments, we may assume that our guess corresponds to the actual way M is spread out over B . Thus, our goal now is to ensure the existence of a subgraph \mathcal{R}_B of \mathcal{R} which is isomorphic to B such that the k occurrences of H are spread in \mathcal{R}_B according to the correct B -assignment. Obviously, we cannot actually find \mathcal{R}_B in FPT-time, since SUBGRAPH ISOMORPHISM is known to be W[1]-hard, but we show that this is not necessary to reach the above goal.

4.2 Color-Coding

We apply the technique of color-coding [1] in a non-trivial way. We use a set of colors C_B containing colors for each vertex of B and for each bucket of each edge of B . Before we apply color-coding, we need to describe the elements that we wish to color. We will have an element for each vertex of \mathcal{R} (a *vertex-element*), and for each edge of \mathcal{R} which corresponds to a strip, we create an element for the interior of the strip and for each boundary of the strip (called *edge-elements*). If however the strip is a spot, then we create only a single element. Note that this corresponds directly to the creation of buckets as we did for the B -assignment. We denote the set of elements by \mathcal{E} .

We now color the elements of \mathcal{E} with the colors of C_B . For this, we use a $|C_B|$ -perfect hash family of colorings \mathcal{F} where we are guaranteed that for any subset of C_B vertices and edges in \mathcal{R} , there is a subset of colorings in \mathcal{F} that will assign distinct colors to this subset in all possible ways. According to [1] there exists such a family \mathcal{F} of size at most $f(|C_B|) \cdot n^{O(1)} = f'(k) \cdot n^{O(1)}$ for computable functions f and f' , and this family can be constructed in FPT-time with respect to $|C_B|$ (and hence with respect to k). We call a coloring *nice* if \mathcal{R}_B is colored in a correct way, i.e. if the colors of C_B indicate the isomorphism between B and \mathcal{R}_B . Observe that the set of colorings \mathcal{F} above assures us that if M is indeed an induced subgraph of G , one of the colorings in \mathcal{F} will be nice. Thus, to complete the description of our algorithm, all that remains is to show the lemma below.

Lemma 4.3 *Given a nice coloring σ of \mathcal{E} , the induced H -matching M can be found in G in FPT-time with respect to k .*

Proof: First, we check whether the coloring of the vertices and edges is consistent. That is, each vertex-element should have received a vertex-color, each edge-element should have received an edge-color of the right type, each of the at most three edge-elements of the same edge should have received edge-colors that correspond to the coloring of an edge of B , and for each edge e its incident vertices have received colors that correspond to the coloring of the vertices of B incident to the edge colored in the way that e is colored. Any vertex or edge that is inconsistently colored is blanked, i.e. we remove its color. Note that if the fourth check above fails, we blank the elements of the edge and not of its incident vertices.

Second, we consider all strips (J, Z) in turn for which the corresponding edge was not blanked. Through the current B -assignment and using the coloring σ , we know what we need to find in this strip. In particular, consider the occurrences of H assigned to the boundaries of the strip. Recall that for each boundary, vertices of at most one occurrence can be assigned. Denote the vertices of the at most two occurrences assigned to the boundaries of the strip by H^{b_i} . We then use exhaustive enumeration to try all possible realizations X of the vertices of each H^{b_i} in J that are consistent with the current B -assignment. Then we remove $N[X]$ from $J \setminus Z$ and, if no vertices are assigned by the B -assignment to a boundary $N(z)$, we remove $N[z]$ from J as well. Call the resulting graph J' . From Theorem 2.1, we know that $\alpha(J') \leq 4$ or J' is a fuzzy circular-arc graph. Hence, it follows from Lemma 4.2 and Theorem 3.7 that we can solve INDUCED H -MATCHING on J' in polynomial time. Denote the solution for X by $M^e(X)$. Let X_{\max}^e denote a set X such that $|M^e(X)|$ is maximal. We then verify that $|M^e(X)|$ is at least the number of occurrences of H prescribed by the B -assignment; otherwise, we blank all elements of the edge.

After this step, we have actually found all k' occurrences of H that are in the interior of a strip according to M and the B -assignment. It remains to find occurrences of H that span multiple strips or affect a strip boundary. For this, we color the vertices of G with a set of colors $C_{k''} = \{0, \dots, k''\}$ where $k'' = k - k'$. For each edge e that is not blanked, consider X_{\max}^e . Recall that X_{\max}^e consists of vertices of at most two occurrences of H , say occurrences $i, j \in \{1, \dots, k''\}$. We first color the vertices of X_{\max}^e (in G) by colors i and j from $C_{k''}$ in the obvious way (i.e. vertices from occurrence i receive color i), and color all remaining vertices of $V(J) \setminus Z$ with color 0. The crucial observation is that, by construction, any pair of vertices from different color classes in $C_{k''}$ will be independent. It is thus sufficient to find a copy of H in each color class of $C_{k''}$. This takes $n^{O(h)}$ time in total, where $h = |V(H)|$. By combining these copies of H with the copies in $M^e(X_{\max}^e)$ for each $e \in E(\mathcal{R})$, we find an induced H -matching of size at least k . \square

5 Polynomial Kernel

The main result of this section is the following theorem.

Theorem 5.1 *Let H be a fixed clique. Then INDUCED H -MATCHING has a polynomial kernel with respect to the parameter k .*

We now prove Theorem 5.1. Throughout, we assume that $\alpha(G) > 4$ and that G is not a fuzzy-circular arc graph. Otherwise, we can solve INDUCED H -MATCHING in polynomial time by Lemma 4.2 and Theorem 3.7, and reduce to a trivial “yes”- or “no”-instance. We can then also assume that we have used Theorem 2.1 and computed a strip-structure with a corresponding strip-graph \mathcal{R} .

5.1 Bounding the Size of the Strip-Graph

First, we show that we can assume that \mathcal{R} is a hypergraph with a bounded number of edges. The approach is based on one in [26] and [6] for H -MATCHING. We need the following auxiliary lemmas.

We define the *dis-degree* of a vertex $r \in \mathcal{R}$ as the cardinality of the set $\bigcup_{e \in \mathcal{R}: r \in e} e \setminus \{r\}$. In other words, this is the number of distinct neighbors of r in \mathcal{R} .

Proposition 5.2 *If a vertex $v \in V(G)$ is not contained in any induced subgraph isomorphic to H , then we can safely remove v from G .*

Lemma 5.3 *Let $r \in \mathcal{R}$ be a vertex in the strip-graph of dis-degree $d \geq hk$. Then G has an induced H -matching of size k if and only if $G' := G - C(r)$ has an induced H -matching of size $k - 1$.*

Proof: If G has an induced H -matching M of size k , then G' has an induced H -matching of size $k - 1$, since at most one occurrence of H in M can include vertices of the clique $C(r)$. For the converse direction, let M be an induced H -matching of G' of size $k - 1$. Consider the set D of distinct vertices contained in an edge together with r . We call an element x of D *marked* if a vertex of M is contained in a strip corresponding to an edge between x and r or contained in $C(x)$. Clearly, any occurrence of H in M can be responsible for marking at most h vertices of D . It follows that at least h vertices of D are unmarked. For each unmarked vertex, pick a vertex of $C(r)$ that is contained in a strip corresponding to an edge between x and r . This yields a new occurrence of H . Since neither the chosen strip nor $C(x)$ for unmarked vertices x contains any vertices of M , this occurrence is independent of the occurrences of H in M . Hence we have obtained an induced H -matching of size k in G . \square

We call an edge $e \in E(\mathcal{R})$ *promising* if it corresponds to a strip (J_e, Z_e) such that $J \setminus N[Z_e]$ contains a subgraph isomorphic to H . The proof of the following proposition is trivial.

Proposition 5.4 *The strip-graph has at most $k - 1$ promising edges, or G has an induced H -matching of size k .*

We can also bound the number of nonpromising edges on each two vertices. We call an edge of \mathcal{R} *used* by an occurrence of H if the strip corresponding to this edge contains a vertex contained in this occurrence of H . Similarly, given an induced H -matching M , we call an edge of \mathcal{R} *used* by M if the edge is used by an occurrence of H in M .

Lemma 5.5 *If $E(\mathcal{R})$ contains $d \geq 2h$ nonpromising edges on the vertices $x, y \in V(\mathcal{R})$ (where possibly $x = y$), then we can remove at least $d - 2h$ such edges and their corresponding strips.*

Proof: Let M be any induced H -matching of G . Any occurrence of H in M which contains a vertex of a strip corresponding to a nonpromising edge on x, y must have a vertex in $C(x)$ or $C(y)$. Moreover, since $C(x)$ and $C(y)$ are cliques, neither can contain vertices of more than one occurrence of H in M . If $x = y$, then the occurrence of H in M containing a vertex of $C(x)$ (if it exists) can be replaced by an occurrence obtained by just taking one vertex from the boundary of the strips corresponding to the first h edges $\{x\}$. Hence we can remove $d - h$ arbitrary nonpromising edges $\{x\}$ and their corresponding strips.

So suppose that $x \neq y$. Let E' denote the set of nonpromising edges in $E(\mathcal{R})$ on x, y . Call an edge of E' that corresponds to a stripe *helpful*. Suppose that at least $2h$ edges of E' are helpful. Then the occurrence of H in M containing a vertex of $C(x)$ (if it exists) can be replaced by an occurrence obtained by just taking one vertex from the boundary of the stripes corresponding to the first h helpful edges. Simultaneously, the occurrence of H in M containing a vertex of $C(y)$ (if it exists) can be replaced by an occurrence obtained by just taking one vertex from the boundary of the stripes corresponding to the second h helpful edges. Hence we can just keep $2h$ arbitrary helpful edges and remove all other edges of E' and their corresponding strips.

So suppose that less than $2h$ edges of E' are helpful. Observe that if two edges of E' are used by two distinct occurrences of H in M , then all edges used by these two occurrences are helpful. Hence we need to keep all helpful edges. We also note that if M contains a vertex of a strip that corresponds to an edge of E' that is not helpful, i.e. that corresponds to a spot, then the union of all vertices in the strips corresponding to the edges of E' contains only one occurrence of H in M . This occurrence of H may be

easily replaced by an occurrence obtained by taking one vertex of each strip corresponding to the first h edges of E' . It follows that we need to keep all helpful edges and a minimum of h edges in total, and can remove all other edges of E' and their corresponding strips. \square

Finally, we prove that the number of strips can indeed be bounded.

Lemma 5.6 *Either the strip-graph has $O(h^4k^2)$ edges, or we can reduce (G, k) , or G has an induced H -matching of size k .*

Proof of Lemma 5.6: By Lemma 5.3, we can assume that any vertex of the strip-graph has dis-degree at most hk . By Lemma 5.5, it follows that we can assume that any vertex is incident on $2h^2k$ nonpromising edges. Now apply a greedy algorithm to find a maximal induced H -matching M . Call a vertex used if it is incident to an edge used by M .

We can assume that M has size at most $k - 1$. Note that any occurrence of H in G uses at most h edges of \mathcal{R} . Hence M uses at most $h(k - 1)$ edges of \mathcal{R} . Moreover, since M is maximal, any occurrence of H in G must use an edge of \mathcal{R} incident to a used vertex. There are at most $4h^3k(k - 1) \leq 4h^3k^2$ such edges that are nonpromising. Hence it follows from Proposition 5.4 that, in total, there are at most $2h^3k^2 + k$ edges that are not used but have an endpoint that is used.

Observe that if $r \in V(\mathcal{R})$ is a vertex that is not used by M , then r can be contained in at most $h - 1$ edges, or it would be possible to extend M . For if r is contained in at least h edges, then none of these edges are used in M . Hence we could add h vertices of $C(r)$ to M , contradicting the maximality of M . The lemma follows. \square

5.2 Using the Reduced Strip-Graph

To obtain an intuition of the kernel, we give the following description of a simple algorithm for INDUCED H -MATCHING if H is a fixed clique. We first reduce the strip-graph \mathcal{R} using Lemma 5.6. Then we observe that if we know the behavior of the induced H -matching on $C(r)$ for each $r \in \mathcal{R}$, then we can reduce to polynomial-time solvable instances of induced H -matching on individual strips. Since the number of vertices of \mathcal{R} is bounded, this gives a fixed-parameter algorithm. The kernel will mimic this algorithm by reducing to an instance of WEIGHTED INDEPENDENT SET on general graphs of size polynomial in k . By using a Karp-reduction (as shown in the appendix), we can obtain an instance of INDUCED H -MATCHING on claw-free graphs again.

For our purposes, we define WEIGHTED INDEPENDENT SET as the problem of given a graph G' , a weight function w' , and integers k', K' , to decide whether G' has an independent set of size at least k' and weight at least K' .

Theorem 5.7 *Let H be a fixed clique. In polynomial time, we can reduce an instance of INDUCED H -MATCHING on claw-free graphs with parameter k to an equivalent instance of WEIGHTED INDEPENDENT SET with $O(2^h h^{4h+4} k^{2h+2})$ vertices, maximum vertex-weight at most k , k' bounded by $O(h^4 k^2)$, and K' bounded by k .*

Proof: First, we reduce the strip-graph using Lemma 5.6. Then, for each edge $e \in E(\mathcal{R})$, and each pair of integers $i, j \in \{-1, 0, \dots, h\}$, we define the following two functions. Suppose that $|e| = 2$, and $e = \{x, y\}$. Then the function $\omega(e, i, j)$ is the size of a maximum induced H -matching on $J_e \setminus Z_e$ that,

- if $i = -1$, contains at least one vertex of $N(z_e^x)$;
- if $0 \leq i \leq h$, contains no vertices of $N[X \cup \{z_e^x\}]$ for at least one $X \subseteq N(z_e^x)$ of size i ,
- if $j = -1$, contains at least one vertex of $N(z_e^y)$;
- if $j \geq 0$, contains no vertices of $N[Y \cup \{z_e^y\}]$ for at least one $Y \subseteq N(z_e^y)$ of size j .

The function ω evaluates to $-\infty$ if an induced H -matching with the given constraints does not exist. It follows immediately from Lemma 4.2 and Theorem 3.7 that the function ω can be computed in polynomial time. We define a similar function $\omega(e, i)$ if $|e| = 1$, and compute its values in polynomial time. Below, we will largely ignore edges e with $|e| = 1$, as they are dealt with in essentially the same manner as edges with $|e| = 2$.

We now describe the instance of WEIGHTED INDEPENDENT SET that we will construct. For each edge $e \in E(\mathcal{R})$ and each $i, j \in \{-1, 0, \dots, h\}$, we create a vertex $v_{i,j}^e$ of weight $\omega(e, i, j)$. For any $e \in E(\mathcal{R})$, we construct a clique on the vertices $v_{i,j}^e$. This expresses that for each strip, we can choose at most one solution.

It remains to coordinate the solutions of the strips. Let \mathcal{U} denote the set of all sets U of integers such that $\sum_{i \in U} i = h$ and $\min_{i \in U} i \geq 1$ for each $U \in \mathcal{U}$. We also add the set $\{0\}$ to \mathcal{U} . Let E_r denote the set of edges in \mathcal{R} containing a vertex $r \in V(\mathcal{R})$ and let $d_r = |E_r|$. Finally, we construct the set \mathcal{P}_r . Each $P \in \mathcal{P}_r$ will be an assignment of the numbers of some $U \in \mathcal{U}$ to the edges of E_r . If $|U| < |E_r|$, then all remaining edges will be assigned the number zero. The set of all such assignments over all $U \in \mathcal{U}$ is \mathcal{P}_r .

We create two types of vertices for each $r \in \mathcal{R}$. Whenever we refer to an edge e containing r , we assume that the second parameter of the function ω corresponds to the vertex z_r^e . Now, first, we create a vertex v_P^r for each $P \in \mathcal{P}_r$. The weight of v_P^r is set to 1, unless P assigns zero to every edge e , in which case the weight is set to zero. If P assigns i' to edge e , then we make v_P^r adjacent to $v_{i,j}^e$ for all $i \neq i'$ and for all j . This expresses that if an

occurrence of H is a subset of $C(r)$, how the vertices of this occurrences are distributed over $C(r)$. Second, we create a vertex v_e^r for each e containing r and give it weight zero. We make this adjacent to $v_{i,j}^e$ for each $i \geq 0$ and for each j , and to $v_{i,j}^{e'}$ for each $e' \neq e$ containing r , each $i \neq 0$, and each j . This expresses that if an occurrence of H contains at least one vertex of an occurrence of H (but less than h), in which strip incident to r these vertices must be found. Finally, we construct a clique on the vertices created for r .

This completes the construction of the instance of WEIGHTED INDEPENDENT SET. It follows from Lemma 4.2 and Theorem 3.7 that this instance can be constructed in polynomial time. Denote the graph by G' and the weight function by w' . We claim that G' has $O(2^h h^{4h+2} k^{2h+2})$ vertices. To see this, we note that for each edge of \mathcal{R} , we create $(h+2)^2$ vertices. By Lemma 5.6, this gives a total of $O(h^6 k^2)$ vertices. We can count the number of vertices created for each vertex r of \mathcal{R} as follows. Observe that $|\mathcal{U}| \leq 2^{h+1}$ (more precise bounds are known, but this will suffice). We can assume that the numbers of each $U \in \mathcal{U}$ are ordered from small to large. To bound $|\mathcal{P}_r|$, we note that the number of ordered subsets of $|E_r|$ of size at most h is at most $|E_r|^h$. For each ordered subset X , we create a bijection to each $U \in \mathcal{U}$ with $|U| = |X|$ by assigning the i -th element of U to the i -th element of X . We then extend this assignment to E_r by assigning zero to any elements of $E_r \setminus X$. This shows that $|\mathcal{P}_r|$ is $O(2^h h^{4h} k^{2h})$. The number of vertices created for each r thus is $O(2^h h^{4h} k^{2h})$. The claim follows.

To bound the weights, observe that if $\omega(e, i, j) > k$ for some e, i, j , then the instance of INDUCED H -MATCHING is a “yes”-instance and we can easily reduce to a trivial “yes”-instance of WEIGHTED INDEPENDENT SET. Hence all the weights are at most k . Finally, let k' be equal to the number of vertices and edges in \mathcal{R} , and let $K' = k$. Note that k' is bounded by $O(h^4 k^2)$ by Lemma 5.6.

We claim that INDUCED H -MATCHING on G, k is a “yes”-instance if and only if WEIGHTED INDEPENDENT SET on G', w', k', K' is a “yes”-instance. For the one direction, consider an induced H -matching M of size k . Now construct an independent set I of G' as follows. Initially, mark all vertices $v_{i,j}^e$ over all i, j as *potentials*. For each $r \in V(\mathcal{R})$, check if an occurrence of H in M is fully contained in $C(r)$. If so, find the assignment P of \mathcal{P}_r induced by this occurrence. Then we add v_P^r to I . Moreover, we remove the potential mark from all neighbors of v_P^r . We proceed similarly if no vertex of M is contained in $C(r)$. Otherwise, a vertex of an occurrence of H in M is contained in $C(r)$, but there is also a vertex of this occurrence in $J_e \setminus N[z_r^e]$ for some e containing r . Then we add v_e^r to I and remove the potential mark from all its neighbors. All remaining potential vertices form an independent set by construction of G' . Moreover, for each edge e exactly one vertex $v_{i,j}^e$ is a potential vertex. This gives the required independent set. It has weight at least K' as M has size k .

For the converse, let I be an independent set of G' of size at least k' and weight at least K' . Observe that by construction of G' and k' , $|I| = k'$. We now use the inverse of the above process to construct an induced H -matching M . Since $K' = k$, M will have size k . The theorem follows. \square

It is now straightforward to prove Theorem 5.1.

Proof of Theorem 5.1: If $|V(H)| = 1$, then the problem is actually INDEPENDENT SET. Since this can be solved in $O(n^3)$ time on claw-free graphs [8], we can easily reduce to a trivial “yes”- or “no”-instance. So assume that $|V(H)| > 1$. Since WEIGHTED INDEPENDENT SET is in NP and INDUCED H -MATCHING on claw-free graphs is NP-hard when H is a fixed clique of size at least two [12, 19, 21], there exists a Karp-reduction from WEIGHTED INDEPENDENT SET to INDUCED H -MATCHING on claw-free graphs when H is a clique. Note that when the input graph G to WEIGHTED INDEPENDENT SET is a graph with size polynomial in k and maximum vertex-weight at most k , the INDUCED H -MATCHING instance produced by this reduction is has size polynomial in k . The proof now follows from Theorem 5.7. \square

6 Hardness

In this section we prove that INDUCED H -MATCHING is W[1]-hard with respect to k on $K_{1,4}$ -free graphs when H is any fixed clique. For this, we first show that INDEPENDENT SET is W[1]-hard on $K_{1,4}$ -free graphs. We will present a reduction from the MULTICOLORED CLIQUE problem. In this problem, we are given a graph G and an assignment of colors $\{1, \dots, k\}$ to the vertices of the graph, and the question is whether G admits a k -clique using exactly one vertex of each color. This problem is well known to be W[1]-hard (see e.g. [10]).

Theorem 6.1 INDEPENDENT SET on $K_{1,4}$ -free graphs is W[1]-hard.

Proof: Let (G, k) be an instance of MULTICOLORED CLIQUE where each vertex has been assigned a color in $\{1, \dots, k\}$. Let V_i denote the set of vertices in $V(G)$ having color i . For each color i , construct a gadget on $|V_i| \cdot (k - 1)$ vertices. We may imagine them to be organized in a matrix-like shape with $|V_i|$ rows and $k - 1$ columns, such that each row corresponds to a vertex of V_i and each column to a color other than i . We then add edges from $(v, j) \in V_i \times \{1, \dots, k\} \setminus \{i\}$ to (u, j) and to $(u, j + 1)$ for all $u \in V_i \setminus \{v\}$, where we take $j + 1$ to be 1 if $j = k$. Observe that in this way each row of the gadget is an independent set of size $k - 1$. Moreover, the edges to (u, j) and $(u, j + 1)$ ensure that any independent set of size $k - 1$ of the gadget consists of vertices of exactly one row.

For each pair of colors i, j ($i \neq j$), we also construct a matrix-like gadget. It has $|E_{ij}|$ rows, where $E_{ij} = (V_i \times V_j) \cap E$, and two columns, one for color i and one for color j . Each element of the matrix-like gadget again consists of one vertex. We add all edges between the vertices of the gadget, except that the edge between (e, i) and (e, j) is not present for any $e \in E_{ij}$.

We then connect the vertex and edge gadgets as follows. Consider a pair of colors i, j and some $u \in V_i$. We then connect (u, j) in the gadget for V_i to all (e, i) in the gadget for E_{ij} for which $u \notin e$.

We call the resulting graph G' . Note that G' is indeed $K_{1,4}$ -free, as the neighborhood of each vertex can be partitioned into three cliques. Finally, we set $k' = k(k-1) + 2\binom{k}{2} = 2k(k-1)$. The resulting instance is (G', k') .

Suppose that G has a multicolored clique K of size k . We construct the following independent set I . For the unique vertex u in $K \cap V_i$ for each color i , we add the $k-1$ vertices on the row of the gadget of V_i that corresponds to u to I . For the unique edge e in $K \cap E_{ij}$ for each pair of colors i, j , we add the two vertices on the row of the gadget of V_i that corresponds to e to I . It is immediate from the description of the construction that this is an independent set and that it has size k' .

Suppose that G' has an independent set I of size k' . By construction, it follows that each edge gadget can have at most 2 vertices in I , as the columns of the gadget are cliques. Moreover, they must be in the same row. Similarly, each vertex gadget can have at most $k-1$ vertices in I . Moreover, they must be in the same row, because otherwise there would be $(u, j) \in I$ and $(u', j+1) \in I$ with $u, u' \in V_i$. This is explicitly disallowed by the construction. Hence each vertex gadget and each edge gadget has exactly one selected row. Now observe that the selected vertex rows and edge rows together induce a multicolored clique. This clique has size k by our choice of k' . \square

We now show how this theorem implies the hardness of INDUCED H -MATCHING in $K_{1,4}$ -free graphs for any fixed clique H .

Theorem 6.2 INDUCED H -MATCHING on $K_{1,4}$ -free graphs is $W[1]$ -hard for any fixed clique H .

Proof: Let (G, k) be an instance of INDEPENDENT SET on $K_{1,4}$ -free graphs. Construct a graph G' by, for each $v \in V(G)$, adding $|V(H)|-1$ new vertices and making them adjacent to each other and to all vertices of $N[v]$. That is, we replace v by a twin set of size $|V(H)|$. Call this twin set H_v . Clearly, G' has an induced $K_{1,4}$ if and only if G has an induced $K_{1,4}$. The resulting instance of INDUCED H -MATCHING is (G', k) .

Suppose that G has an independent set I of size k . Let M be the set of H_v for all $v \in I$. Since I is an independent set, M must be an induced H -matching. It clearly has size k .

Suppose that (G', k) has an induced H -matching M of size k . Consider a copy H' of H in M which uses vertices from H_v for any v in some set $V_{H'} \subseteq V(G)$ with $|V_{H'}| \geq 2$. Then, by construction of G' , no copy of H in M can use a vertex of H_u for any $u \in V(G) \setminus V_{H'}$. Moreover, no copy of H in M other than H' can use a vertex of H_u for any $u \in V(H')$. But then we can just replace H' by H_v for some $v \in V_{H'}$. It follows that $\{v \mid H_v \in M\}$ is an independent set of G of size k .

The theorem is now immediate from Theorem 6.1. \square

7 Discussion

In this section, we discuss the possibility of extending the work of this paper. In particular, we give some open problems.

Even though we showed that INDUCED H -MATCHING on $K_{1,3}$ -free graphs is fixed-parameter tractable for fixed connected graphs H , and even admits a polynomial kernel if H is a clique, these results are unlikely to extend to $K_{1,4}$ -free graphs, as shown in the previous section.

We also note that the running times of our results contain a term $n^{O(h)}$, where $h = |V(H)|$. Unfortunately, extending to the case where h is also a parameter is unlikely to succeed, as INDUCED H -MATCHING becomes W[1]-hard even on line graphs and co-bipartite graphs [13, 14].

It is unclear however whether the results of this paper extend to graphs H with multiple components. This seems difficult even for fuzzy circular-arc graphs, as the natural reduction to an INDEPENDENT SET problem fails [10].

Another interesting open question is whether the polynomial kernel for INDUCED H -MATCHING on claw-free graphs extends beyond the case when H is a fixed clique. The proof of Theorem 5.1 actually extends to this more general case. However, the obstacle to extend our current techniques is that we are unable to find a lemma similar to Lemma 5.6 if H is not a clique.

References

- [1] Alon, N., Yuster, R., Zwick, U.: Color-coding. J. ACM 42(4), 844–856 (1995)
- [2] Cameron, K., Hell, P.: Independent packings in structured graphs. Math. Programming 105(2-3), 201–213 (2006)
- [3] Chudnovsky, M., Ovetsky, A.: Coloring quasi-line graphs. J. Graph Theory 54(1), 41–50 (2007)
- [4] Chudnovsky, M., Seymour, P.D.: The structure of claw-free graphs. In: Surveys in Combinatorics. vol. 327, pp. 153–171. Cambridge University Press (2005)

- [5] Cygan, M., Philip, G., Pilipczuk, M., Pilipczuk, M., Woitaszczyk, J.O.: Dominating set is fixed parameter tractable in claw-free graphs. *Theoretical Computer Science* 412(50), 6982–7000 (2011)
- [6] Dell, H., Marx, D.: Kernelization of packing problems. In: *Proc. SODA 2012*. pp. 68–81 (2012)
- [7] Downey, R.G., Fellows, M.R.: *Parameterized complexity*. Springer (1999)
- [8] Faenza, Y., Oriolo, G., Stauffer, G.: An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$ -algorithm for the weighted stable set problem. In: *Proc. SODA 2011*. pp. 630–646 (2011)
- [9] Faudree, R.J., Flandrin, E., Ryjáček, Z.: Claw-free graphs - A survey. *Discrete Math.* 164(1-3), 87–147 (1997)
- [10] Fellows, M.R., Hermelin, D., Rosamond, F.A., Vialette, S.: On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science* 410(1), 53–61 (2009)
- [11] Fellows, M.R., Knauer, C., Nishimura, N., Ragde, P., Rosamond, F.A., Stege, U., Thilikos, D.M., Whitesides, S.: Faster fixed-parameter tractable algorithms for matching and packing problems. *Algorithmica* 52(2), 167–176 (2008)
- [12] Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman (1979)
- [13] Golovach, P., Paulusma, D., van Leeuwen, E.J.: Induced disjoint paths in AT-free graphs. In: *Proc. SWAT 2012* (2012), to appear
- [14] Golovach, P.A., Paulusma, D., van Leeuwen, E.J.: Induced disjoint paths in claw-free graphs. *CoRR* abs/1202.4419 (2012)
- [15] Heggernes, P., Meister, D., Villanger, Y.: Induced subgraph isomorphism on interval and proper interval graphs. In: Cheong, O., Chwa, K.Y., Park, K. (eds.) *Proc. ISAAC 2010*. LNCS, vol. 6507, pp. 399–409. Springer-Verlag, Berlin (2010)
- [16] Hermelin, D., Mních, M., van Leeuwen, E.J., Woeginger, G.J.: Domination when the stars are out. In: L. Aceto, M. Henzinger, J.S. (ed.) *Proc. ICALP 2011*. LNCS, vol. 6755, pp. 462–473. Springer-Verlag, Berlin (2011)
- [17] Hermelin, D., Wu, X.: Weak compositions and their applications to polynomial lower bounds for kernelization. In: *Proc. SODA 2012*. pp. 104–113 (2012)

- [18] King, A.D., Reed, B.A.: Bounding χ in terms of ω and Δ for quasi-line graphs. *J. Graph Theory* 59(3), 215–228 (2008)
- [19] Kirkpatrick, D., Hell, P.: On the complexity of general graph factor problems. *SIAM J. Computing* 12(3), 601–609 (1983)
- [20] Kneis, J., Mölle, D., Richter, S., Rossmanith, P.: Divide-and-color. In: Fomin, F. (ed.) *Proc. WG 2006*. LNCS, vol. 4271, pp. 58–67. Springer-Verlag, Berlin (2006)
- [21] Kobler, D., Rotics, U.: Finding maximum induced matchings in subclasses of claw-free and p_5 -free graphs, and in graphs with matching and induced matching of equal maximum size. *Algorithmica* 37(4), 327–346 (2003)
- [22] Marx, D.: Efficient approximation schemes for geometric problems? In: Brodal, G.S., Leonardi, S. (eds.) *Proc. ESA 2005*. LNCS, vol. 3669, pp. 448–459. Springer-Verlag, Berlin (2005)
- [23] Moser, H.: A problem kernelization for graph packing. In: Nielsen, M., Kucera, A., Miltersen, P.B., Palamidessi, C., Tuma, P., Valencia, F.D. (eds.) *Proc. SOFSEM 2009*. LNCS, vol. 5404, pp. 401–412. Springer-Verlag, Berlin (2009)
- [24] Moser, H., Thilikos, D.M.: Parameterized complexity of finding regular induced subgraphs. In: Broersma, H., Dantchev, S.S., Johnson, M., Szeider, S. (eds.) *Proc. ACiD 2006*. Texts in Algorithmics, vol. 7, pp. 107–118. King’s College, London (2006)
- [25] Oriolo, G., Pietropaoli, U., Stauffer, G.: On the recognition of fuzzy circular interval graphs. *Discrete Math.* 312(8), 1426–1435 (2012)
- [26] Prieto, E., Sloper, C.: Looking at the stars. *Theoretical Computer Science* 351(3), 437–445 (2006)
- [27] Roussopoulos, N.: A $\max\{m, n\}$ algorithm for determining the graph H from its line graph G . *Inf. Process. Lett.* 2(4), 108–112 (1973)

A Structure of Claw-Free Graphs

In this section, we prove the structure theorem for claw-free graphs given in Theorem 2.1.

A.1 Trigraphs

To describe the algorithmic structure theorem for claw-free graphs, we require an extended notion of what a graph is. A *trigraph* consists of a set of vertices and a set of two types of edges, normal edges and *semi-edges*, and any vertex can be incident to at most one semi-edge. Observe that any graph is a trigraph, and that any trigraph without semi-edges is a graph.

We call two vertices of a trigraph *strongly adjacent* if there is a normal edge between them, *semi-adjacent* if there is a semi-edge between them, and *strongly anti-adjacent* if there is no edge between them. Two vertices are called *adjacent* if they are strongly adjacent or semi-adjacent, and called *anti-adjacent* if they are semi-adjacent or strongly anti-adjacent.

A trigraph G is a *thickening* of a trigraph G' if there is a set $\mathcal{X} = \{X_{v'} \mid v' \in V(G')\}$ such that each $X_{v'}$ is nonempty, the $X_{v'}$ partition $V(G)$, and

- if u' is strongly adjacent to v' , then each $u \in X_{u'}$ is strongly adjacent to each $v \in X_{v'}$;
- if u' is strongly anti-adjacent to v' , then each $u \in X_{u'}$ is strongly anti-adjacent to each $v \in X_{v'}$;
- if u' is semi-adjacent to v' , then there exist $u_1, u_2 \in X_{u'}$, $v_1, v_2 \in X_{v'}$ such that u_1 is adjacent to v_1 and u_2 is anti-adjacent to v_2 .

We sometimes talk about the thickening \mathcal{X} of G' to G .

Let G be a trigraph. Given disjoint sets $A, B \subseteq V(G)$, we say that A is (*strongly*) *complete* to B if each vertex of A is (strongly) adjacent to each vertex of B . Similarly, we define the notions of *anti-complete* and *strongly anti-complete*. A set $C \subseteq V(G)$ is a (*strong*) *clique* if every pair of vertices of C is (strongly) adjacent. A set $I \subseteq V(G)$ is a (*strong*) *independent set* if every pair of vertices of C is (strongly) anti-adjacent. We use $\alpha(G)$ to denote the size of a largest independent set of G . For any $X \subseteq V(G)$, $G[X]$ is the subgraph of G induced by X . If $G[X]$ is isomorphic to some trigraph G' , then G' is said to be an *induced subtrigraph* of G . A *claw* is the trigraph with four vertices c, c_1, c_2, c_3 , three edges from c (the *center* of the claw) to c_1, c_2, c_3 respectively, and $\{c_1, c_2, c_3\}$ is an independent set. Then G is *claw-free* if no induced subtrigraph of G is isomorphic to a claw.

A.2 Joins and Other Structures

Let G be a trigraph throughout. A *0-join* is a partition of $V(G)$ into V_1, V_2 such that V_1 is strongly anti-complete to V_2 .

A *1-join* is a partition of $V(G)$ into V_1, V_2 for which there exist $A_1 \subseteq V_1$ and $A_2 \subseteq V_2$ such that

- $A_1 \cup A_2$ is a strong clique;
- $V_1 \setminus A_1$ is strongly anti-complete to V_2 and $V_2 \setminus A_2$ is strongly anti-complete to V_1 ;
- $A_1, A_2, V_1 \setminus A_1,$ and $V_2 \setminus A_2$ are nonempty.

The definition of a *pseudo-1-join* is the same, except that the last condition is replaced by the condition that V_1, V_2 are not allowed to be strong independent sets. Observe that a trigraph admitting a 1-join but not a 0-join also admits a pseudo-1-join.

A *generalized 2-join* is a partition of $V(G)$ into V_0, V_1, V_2 for which there exist disjoint sets $A_1, B_1 \subseteq V_1$ and $A_2, B_2 \subseteq V_2$ such that

- $A_1 \cup A_2 \cup V_0$ and $B_1 \cup B_2 \cup V_0$ form a strong clique;
- $V_1 \setminus (A_1 \cup B_1)$ is strongly anti-complete to $V_0 \cup V_2$ and $V_2 \setminus (A_2 \cup B_2)$ is strongly anti-complete to $V_0 \cup V_1$;
- $A_1, A_2, B_1, B_2, V_1 \setminus (A_1 \cup B_1),$ and $V_2 \setminus (A_2 \cup B_2)$ are nonempty.

The definition of a *pseudo-2-join* is the same, except that the last condition is replaced by the condition that V_1, V_2 are not allowed to be strong independent sets. Observe that a trigraph admitting a generalized 2-join but not a 1-join also admits a pseudo-1-join.

Two disjoint sets $A, B \subseteq V(G)$ form a *W-join* if A and B are strong cliques, every vertex of $V(G) \setminus (A \cup B)$ is either strongly complete or strongly anti-complete to A and either strongly complete or strongly anti-complete to B , A is neither strongly complete nor strongly anti-complete to B , and $\max\{|A|, |B|\} \geq 2$.

The trigraph G admits *twins* if it has two strongly adjacent vertices for which any third vertex is either strongly adjacent or strongly anti-adjacent to both vertices. A set of vertices that are pairwise twins is called a *twin set*.

The definition of a strip-structure extends naturally to trigraphs, where we demand that $C(h)$ is a strong clique, and that for any strip (J, Z) , Z is a strong independent set and any vertex of $V(J)$ that is adjacent to some vertex of Z is in fact strongly adjacent to it. Similarly, we call a strip (J, Z) a spot if J consists of three vertices z_1, s, z_2 , such that z_1, z_2 are strongly anti-adjacent and both z_1 and z_2 are strongly adjacent to s . The definition of a stripe again extends directly. Then a strip-structure of a trigraph is called *purified* if each strip is either a spot or a stripe.

Finally, a stripe (J, Z) is a *thickening of a stripe* (J', Z') if J is a thickening \mathcal{X} of J' and there is a bijection σ between Z and Z' such that $X_{z'} = \{\sigma(z')\}$.

A.3 Special Trigraphs

We identify eight special classes of trigraphs, $\mathcal{S}_0, \dots, \mathcal{S}_7$. For the precise definitions of classes $\mathcal{S}_1, \mathcal{S}_4, \mathcal{S}_5, \mathcal{S}_6$, and \mathcal{S}_7 , we refer to Hermelin et al. [16]; here we will only use the property that all trigraphs G in these classes satisfy $\alpha(G) \leq 3$ [16].

The class \mathcal{S}_0 contains all line trigraphs. We say that G is a *line trigraph* of a graph G' if each vertex of G corresponds to an edge of G' and two vertices e, f are adjacent if and only if the corresponding edges are incident to the same vertex $h \in V(G')$. Moreover, if h has degree at least three, then e, f are strongly adjacent.

The class \mathcal{S}_2 contains all XX-trigraphs. We say that G is an *XX-trigraph* if it can be obtained by removing any subset X of $\{v_7, v_{11}, v_{12}, v_{13}\}$ from the following trigraph on vertex set v_1, \dots, v_{13} :

- v_i is adjacent to v_{i+1} for $i = 1, \dots, 5$ and v_6 is adjacent to v_1 ; also v_i is anti-adjacent to v_j for each $i = 1, \dots, 4$ and each $i + 2 \leq j \leq 6$,
- v_7 is strongly adjacent to v_1 and v_2 ,
- v_8 is strongly adjacent to v_4, v_5 , and possibly adjacent to v_7 ,
- v_9 is strongly adjacent to v_1, v_2, v_3 , and v_6 ,
- v_{10} is strongly adjacent to v_3, v_4, v_5 , and v_6 , and adjacent to v_9 ,
- v_{11} is strongly adjacent to v_1, v_3, v_4, v_6, v_9 , and v_{10} ,
- v_{12} is strongly adjacent to v_2, v_3, v_5, v_6, v_9 , and v_{10} ,
- v_{13} is strongly adjacent to v_1, v_2, v_4, v_5, v_7 , and v_8 .

All other pairs of vertices are strongly anti-adjacent.

We need the following observation about XX-trigraphs.

Proposition A.1 *Let G be a claw-free XX-trigraph obtained from the above graph on $\{v_1, \dots, v_{13}\}$ by removing $X \subseteq \{v_7, v_{11}, v_{12}, v_{13}\}$. Then $\alpha(G) \leq 4$.*

Proof: We first claim that v_i is strongly adjacent to v_{i+1} for $i = 1, \dots, 5$ and v_6 is strongly adjacent to v_1 . To see this, observe that

- if v_1, v_2 are semi-adjacent, then v_9, v_1, v_2, v_{10} is a claw;
- if v_2, v_3 are semi-adjacent, then v_9, v_2, v_3, v_6 is a claw;
- if v_3, v_4 are semi-adjacent, then v_{10}, v_3, v_4, v_6 is a claw;
- if v_4, v_5 are semi-adjacent, then v_{10}, v_4, v_5, v_9 is a claw;
- if v_5, v_6 are semi-adjacent, then v_{10}, v_3, v_5, v_6 is a claw;

- if v_6, v_1 are semi-adjacent, then v_9, v_1, v_3, v_6 is a claw.

The claim follows.

Let I be a maximum independent set of G . Since adjacency between v_7, v_8 can only decrease the size of a maximum independent set, we may assume that v_7, v_8 are strongly anti-adjacent. Similarly, v_i can be assumed to be strongly anti-adjacent to v_j for each $i = 1, \dots, 4$ and each $i + 2 \leq j \leq 6$. Moreover, we can assume that $X = \emptyset$. Then, since $N[v_7] \subseteq N[v_1], N[v_2], N[v_{13}]$ and v_1, v_2, v_{13} are pairwise strongly adjacent, we can assume that $I \cap \{v_1, v_2, v_{13}\} = \emptyset$. Since $N(v_7) = \{v_1, v_2, v_{13}\}$, $v_7 \in I$. Similarly, arguing about v_8 in relation to v_4, v_5, v_{13} , we can deduce that $v_8 \in I$. Note that then $v_1, v_2, v_4, v_5, v_{13} \notin I$. If $v_3 \in I$ or $v_6 \in I$, then $v_9, v_{10}, v_{11}, v_{12} \notin I$, and $\alpha(G) \leq 4$. So suppose that $v_3, v_6 \notin I$. If $v_9 \in I$ or $v_{10} \in I$, then $v_{11}, v_{12} \notin I$, and $\alpha(G) \leq 4$. So suppose that $v_9, v_{10} \notin I$. Then $\alpha(G) \leq 4$. \square

The class \mathcal{S}_3 contains all circular interval trigraphs. A trigraph G' is a (*long*) *circular interval trigraph* if there exists a set of arcs $\mathcal{F} = \{F_1, \dots, F_\ell\}$ on a circle Σ , such that no two arcs share an endpoint and the union of any three arcs does not contain the entire circle, and there is a mapping $p : V(G') \rightarrow \Sigma$ such that all points $p_{v'}$ are distinct and

- u', v' are adjacent if $p_{u'}, p_{v'}$ are contained in a common arc F_i ; moreover, if one of $p_{u'}, p_{v'}$ is not an endpoint of F_i , then u', v' are strongly adjacent;
- u', v' are strongly anti-adjacent if there is no arc F_i containing both $p_{u'}, p_{v'}$.

We prove some properties of circular interval trigraphs below.

A.4 Fuzzy Circular-Arc Graphs vs Circular Interval Trigraphs

We show that any graph that is a thickening of a circular interval trigraph is a fuzzy circular-arc graph.

We need the following definitions. Consider a set of arcs on a circle (throughout, we assume that arcs are not a single point). Then the set is called *proper* if no arc is a subset of another, *long* if the union of any three arcs does not contain the entire circle, *almost proper* if any arc that is a subset of another is in fact equal to the other, *strict* if no two arcs share an endpoint, and *almost strict* if any arc can have at most one endpoint that is also an endpoint of another arc that is not equal to it.

Note that in the definition of a circular interval trigraph one can assume that the set \mathcal{F} of arcs is proper. Moreover, the set of arcs is long and strict.

Lemma A.2 *Any graph G that is a thickening of a circular interval trigraph has a representation such that each vertex v of G corresponds to an arc A_v , the set of arcs is long, almost proper, and almost strict, and*

- *if $(u, v) \in E(G)$, then A_u and A_v intersect;*
- *if $(u, v) \notin E(G)$, then either A_u and A_v do not intersect or $A_u \cap A_v$ consists of precisely one point.*

Proof: Suppose that G is a thickening $\mathcal{X} = \{X_{v'} \mid v' \in V(G')\}$ of a circular interval trigraph G' . We first show that G' has a representation such that each vertex v' of G corresponds to an arc $A_{v'}$, the set of arcs is proper, and

- u', v' are strongly adjacent if and only if $A_{u'}$ and $A_{v'}$ intersect in more than one point;
- u', v' are semi-adjacent if and only if $A_{u'} \cap A_{v'}$ contains exactly one point, which is an endpoint of $A_{u'}$ and an endpoint of $A_{v'}$;
- u', v' are strongly anti-adjacent if and only if A_u and A_v do not intersect.

To this end, for each vertex $v' \in V(G')$, let $A_{v'}$ be the arc from $p_{v'}$ to the clockwise furthest endpoint $r_{v'}$ of any arc F_i containing $p_{v'}$. If $r_u = r_{v'}$ for distinct $u, v \in V(G)$, then extend $A_{u'}$ or $A_{v'}$ by an infinitesimal amount clockwise if $p_{u'}$ appears after $p_{v'}$ respectively if $p_{v'}$ appears after $p_{u'}$. This gives the requested representation.

Now the representation as described in the lemma statement can be found by copying $A_{v'}$ for each $v \in X_{v'}$. \square

We observe the following corollary.

Corollary A.3 *Any graph that is a thickening of a circular interval trigraph G' which contains no semi-edge is a proper circular-arc graph.*

We show that the representation of Lemma A.2 indeed provides an equivalent definition of thickenings of circular interval trigraphs.

Lemma A.4 *Any graph G that has a representation as in Lemma A.2 is a thickening of a circular interval trigraph.*

Proof: Let $\mathcal{A} = \{A_v \mid v \in V(G)\}$ be a representation for G as in Lemma A.2. We construct a circular interval trigraph G' . The set of vertices of G' is constructed by taking the left endpoints of all arcs in \mathcal{A} . The set \mathcal{F} of arcs is obtained by first removing all duplicates from \mathcal{A} , and secondly, if two arcs share an endpoint, then the rightmost one can be extended by an infinitesimal amount. It is easy to see that G' is a circular interval trigraph, and that G is a thickening of G' . \square

From Lemma A.2, any graph that is a thickening of a circular interval tri-graph is a fuzzy circular-arc graph. Note that any intersection graph of an almost-proper set of arcs on a circle is a proper circular-arc graph; hence a fuzzy circular-arc graph without any fuzziness is just a proper circular-arc graph. Moreover, any fuzzy circular-arc graph is claw free, and fuzzy circular-arc graphs can be recognized in polynomial time [25].

A.5 Supporting Results

We need the following results from Hermelin et al. [16].

Call a stripe (J, Z) *almost-unbreakable* if J does not admit a 0-join, a pseudo-1-join, or a pseudo-2-join; $J[V(J) \setminus Z]$ does not admit twins; J does not have a W-join (A, B) such that $Z \cap (A \cup B) = \emptyset$.

Lemma A.5 ([16]) *Every claw-free trigraph admits a purified strip-structure for which its strips are either spots or thickenings of almost-unbreakable stripes. Moreover, if G is a graph, such a strip-structure can be found in polynomial time.*

Lemma A.6 ([16]) *Let (J, Z) be (a thickening of) an almost-unbreakable stripe. Then either J is a thickening of a member of $\mathcal{S}_0, \dots, \mathcal{S}_7$ or J is a union of at most three strong cliques.*

Lemma A.7 ([16]) *Let a graph G be a thickening of a line trigraph such that G admits no 0-join, pseudo-1-join, or pseudo-2-join. Then G is the line graph of a multigraph, or G is a union of two strong cliques.*

Lemma A.8 ([16]) *Let G be a trigraph such that $G \in \mathcal{S}_1 \cup \mathcal{S}_4 \cup \mathcal{S}_5 \cup \mathcal{S}_6 \cup \mathcal{S}_7$. Then $\alpha(G) \leq 3$.*

We also require the following observation.

Proposition A.9 *Let G be a trigraph that is a thickening of a trigraph G' . Then $\alpha(G) = \alpha(G')$.*

A.6 Proof of the Structure Theorem

Theorem A.10 (Theorem 2.1 restated) *Let G be a connected claw-free graph that is not a fuzzy circular-arc graph and has $\alpha(G) > 4$. Then G admits a strip-structure such that each strip is either a spot, or a stripe (J, Z) with $1 \leq |Z| \leq 2$ that is a fuzzy circular-arc graph or satisfies $\alpha(J) \leq 4$. Moreover, we can find such a strip-structure in polynomial time.*

Proof: By Lemma A.5, G has a purified strip-structure $(\mathcal{R}, \{(J_e, Z_e) \mid e \in E(\mathcal{R})\})$ such that its strips are spots or thickenings of almost-unbreakable stripes. Since G is connected, \mathcal{R} is connected as well.

Suppose that $|E(\mathcal{R})| = 1$. Then the strip-structure consists of a single strip (G, Z) with $|Z| = 0$. Since spots have $|Z| = 2$, (G, Z) is a thickening of an almost-unbreakable stripe. Hence G does not admit a 0-join, a 1-join, or a generalized 2-join. Since $\alpha(G) > 4$, G is not a union of at most three strong cliques. Now choose G' for which G is a thickening \mathcal{X} of G' such that $|V(G')|$ is minimum. Then $G' \in \mathcal{S}_0 \cup \dots \cup \mathcal{S}_7$ by Lemma A.6. By Lemma A.8, Proposition A.1, and Proposition A.9, we may assume that $G' \in \mathcal{S}_0 \cup \mathcal{S}_3$. If $G' \in \mathcal{S}_3$, then G is a fuzzy circular-arc graph, which we can verify in polynomial time [25]. Hence, by Lemma A.7, G is the line graph of a multigraph M . We can find M in linear time [27] (in case [27] only works for line graphs of simple graphs, reduce all twin sets to a single vertex, mark remaining vertices with the size of their original twin set, run the algorithm of [27], and duplicate all edges according to the markings). From M , we can easily compute the required strip-structure by choosing $\mathcal{R} = M$.

Suppose that $|E(\mathcal{R})| > 1$. Consider each strip (J, Z) in turn. We can check in constant time whether G is a spot, so assume otherwise. Then (J, Z) is a thickening of an almost-unbreakable stripe. Choose (J', Z') for which (J, Z) is a thickening of (J', Z') such that $|V(J')|$ is minimum. We can check in polynomial time whether $\alpha(J') = \alpha(J) \leq 4$, so assume otherwise. Hence we can deduce that $J' \in \mathcal{S}_0 \cup \mathcal{S}_3$ as before. If $J' \in \mathcal{S}_3$, then J' is a fuzzy circular-arc graph, which can be verified in polynomial time. Hence $J' \in \mathcal{S}_0$, by Lemma A.7, J is the line graph of a multigraph M . We can find M in linear time as before. Using M , we can easily refine the strip-structure such that it satisfies the constraints of the theorem statement. \square