Kernelization Upper Bounds for Parameterized Graph Coloring Problems

Pim de Weijer

Master Thesis: ICA-3137910
Supervisor: Hans L. Bodlaender
Computing Science, Utrecht University

Abstract

This thesis studies the kernelization complexity of graph coloring problems with respect to certain structural parameterizations of the input instances, following up on a paper by Jansen and Kratsch (FCT, 2011). In their paper they study how well polynomial-time data reductions can provably shrink instances of coloring problems, in terms of the chosen parameter. In this thesis we do the same except we use different parameters on slightly harder coloring problems. The paper by Jansen and Kratsch shows some interesting results about coloring problems parameterized by the modification distance of the input graph to a graph class on which coloring is polynomial-time solvable, for example parameterizing by the number k of vertex-deletions needed to make the graph chordal. In this thesis we obtain results on parameterizations of Chromatic Number. Every parameterization in this thesis is either by the number k of edge-deletions, edge-additions or edge-modifications (edge-deletions and edge-additions) needed to transform the input graph into a graph which is a member of a well known graph class, e.g. a forest. We obtain various upper and lower bounds for kernels of such parameterizations of Chromatic Number.

Contents

1 Introduction

The kernelization of graph coloring problems is the point of focus in this thesis. Graph coloring problems are interesting to study since there are numerous real world applications that profit from efficient solutions for graph coloring. For instance, oftentimes scheduling problems can be modelled as graph coloring problems. For more information on graph coloring consult [11].

Since graph coloring itself is *NP-complete* the fastest known algorithms for finding the chromatic number of a graph take exponential time. In recent years there have not been huge strides in performance gain for most NP-complete problems. This in contrast to kernelization, a formalization of preprocessing, which has become an increasingly more popular research subject building on a number of interesting results achieved in the last decade or so.

This thesis studies the kernelization complexity of graph coloring problems with respect to certain structural parameterizations of the input instances. In this thesis we follow up on a paper by Jansen and Kratsch [10]. In their paper they study how well polynomial-time data reductions can provably shrink instances of coloring problems, in terms of the chosen parameter. In this thesis we do the same except we use different parameters on slightly harder coloring problems. To study the kernelization complexity of graph coloring problems we will be using the framework of parameterized complexity [6], [9]. We use this framework since it enables us to effectively study how different properties of a graph coloring instance contribute to its difficulty.

A different choice of parameter can lead to significantly different results making the choice of parameter very important. In [10] Jansen and Kratsch describe *structural* parameterizations of graph problems in a notation introduced by Leizhen Cai [4]. Every parameterization in this thesis is either by the number k of edge-deletions, edge-additions or edge-modifications (edge-deletions and edge-additions) needed to transform the graph into a well known graph class $\mathcal{F}$.

- For a graph class $\mathcal{F}$ let $\mathcal{F}{+}ke$ denote the graphs obtained by adding at most k edges to graphs in $\mathcal{F}$; the endpoints of these new edges can be arbitrary. This means that for any graph $G \in \mathcal{F}{+}ke$ there exists a *modulator* $X \subseteq E(G)$ with $|X| \leq k$ such that $G - X \in \mathcal{F}$.

- For a graph class $\mathcal{F}$ let $\mathcal{F}{-}ke$ denote the graphs obtained by deleting at most k edges from graphs in $\mathcal{F}$; the deleted edges can be arbitrary. This means that for any graph $G \in \mathcal{F}{-}ke$ there exists a *modulator* $Y \subseteq \binom{V(G)}{2} \setminus E(G)$ with $|Y| \leq k$ such that $G + Y \in \mathcal{F}$.

- For a graph class $\mathcal{F}$ let $\mathcal{F}{\pm}ke$ denote the graphs obtained by adding and deleting at most k edges in total from graphs in $\mathcal{F}$; the deleted edges as well as the endpoints of the added edges can be arbitrary. This means that for any graph $G \in \mathcal{F}{\pm}ke$ there exist *modulators* $X \subseteq E(G)$ and $Y \subseteq \binom{V(G)}{2} \setminus E(G)$ with $|X| + |Y| \leq k$ such that $G - X + Y \in \mathcal{F}$.

All structural parameters in this thesis are based upon the *edge* modification distances (number of edge deletions/additions) to well-known graph classes. This in contrast to the paper of Jansen and Kratsch [10] where the structural parameters are based upon the *vertex* deletion distances to well-known graph classes, and where the structural parameters can be defined as $\mathcal{F}{+}kv$ with k the number of vertices added to a member of $\mathcal{F}$ to build a graph in $\mathcal{F}{+}kv$. Using this notation we can define a class of parameterized coloring problems with structural parameters.

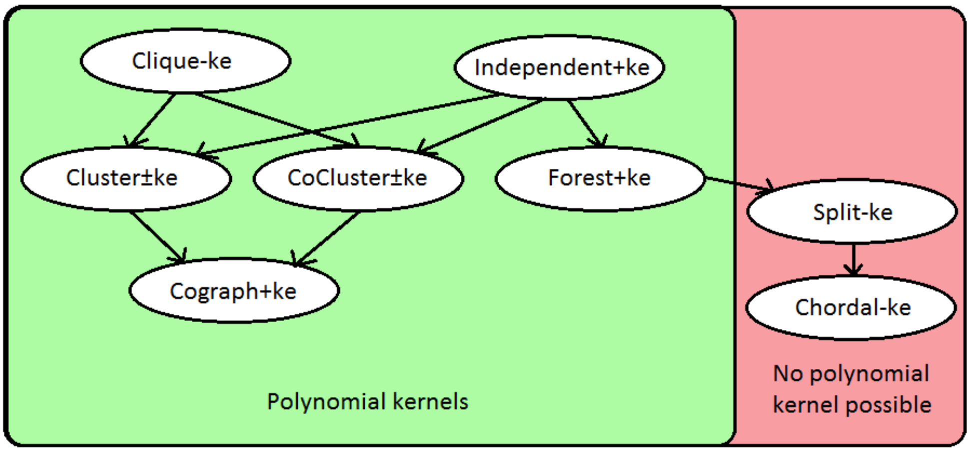

Figure 1: Parameter hierarchy: All parameters depicted in this figure are parameters of Chromatic Number. As such Split$-ke$ should be read as Chromatic Number parameterized by the edge addition distance to a split graph. The fact that there is no polynomial kernel possible for certain parameters is under the assumption that $NP \not\subseteq coNP/poly$.

Chromatic Number on $\mathcal{F}\pm ke$
Input: An undirected graph G, a number q, a modulator $X \subseteq E(G)$ and a modulator $Y \subseteq \binom{V(G)}{2} \setminus E(G)$ such that $G - X + Y \in \mathcal{F}$.
Parameter: The size $k := |X| + |Y|$ of the modulators.
Question: Is $\chi(G) \leq q$?

We assume that modulators are given in the input since we are not interested in finding the modulators itself but rather in finding polynomial kernels.

Hierarchy

We can view the parameter space studied in this thesis as a hierarchy, which is depicted in Figure 1. All parameters in the figure apply to Chromatic Number. The intuition behind the hierarchy is that the origin graph class of an arrow is fully contained in the graph class to which the arrow is pointing. For example Independent$+ke$ has an arrow to Forest$+ke$, which means every independent graph must be a forest, which implies that the parameter of Chromatic Number on Forest$+ke$ can always be smaller or equal to the parameter of Chromatic Number on Independent+ke on a graph G. Now it is not hard to see that if we find a kernel for Chromatic Number on Forest$+ke$, we find a kernel for Chromatic Number on Independent$+ke$. Other way around if we find a lower bound for a kernel of a parameterized problem (e.g. no polynomial kernel possible), we know the lower bound is also a lower bound for all parameters the parameter of the parameterized problem points to. For example Chromatic Number on Split$-ke$ does not admit a polynomial kernel. A split graph is also a chordal graph, so if Chromatic Number on Chordal$-ke$ admits a polynomial kernel then Chromatic Number on Split$-ke$ also admits a polynomial kernel. Since Chromatic Number on Split$-ke$ does not admit a polynomial kernel, we therefore know that Chromatic Number on Chordal$-ke$ does not admit a polynomial kernel.

Results

The results we obtain in this paper are on the parameters displayed in the hierarchy in Figure 1. We find nice upper bounds for Chromatic Number on Forest+ke, Cluster±ke, CoCluster±ke and Cograph+ke. We provide polynomial kernels for those parameterizations of Chromatic Number and prove upper bounds on the amount of vertices left in an instance after using such a kernel. We also find lower bounds for Chromatic Number on Split−ke and Chordal−ke, proving Chromatic Number does not admit a polynomial kernel on both parameters.

Related work

In the first study on structural parameterizations Cai [4] showed that Chromatic Number is in FPT on Split−ke, Split+ke and Chordal−ke. Furthermore, on a related study Marx [12] showed that Chromatic Number on Chordal+ke is also in FPT.

Bodlaender, Jansen and Kratsch [2] introduce the framework of *cross-composition* for obtaining kernelization lower bounds. Fortnow and Santhanem [8] prove the OR-distillation conjecture which is important for cross-composition. Similarly, Drucker [7] proves the AND-distillation conjecture.

Dell and van Melkebeek [5] show some interesting lower bounds for kernels of problems that admit polynomial kernels, such as Vertex Cover.

Organization

We give an introduction to complexity theory and graph theory in Section 2. Some lemmas and propositions we can use to prove upper and lower bounds are given in Section 3.

In Section 4 we prove the upper bounds described in this thesis. Section 4.1 provides a nice upper bound for Chromatic Number parameterized by the edge deletion distance to a forest (Forest+ke). Section 4.2 then provides an upper bound for Chromatic Number parameterized by the edge modification distance to a cluster graph (Cluster±ke). In Section 4.3 Chromatic Number parameterized by the edge modification distance to a cocluster graph (CoCluster±ke) is provided with an upper bound. Lastly, Section 4.4 provides an upper bound for Chromatic Number parameterized by the edge deletion distance to a cograph (Cograph+ke).

In Section 5 we prove the lower bounds described in this thesis. Section 5.1 provides proof that Chromatic Number parameterized by the edge addition distance to a split graph (Split−ke) does not admit a polynomial kernel if $NP \nsubseteq coNP/poly$. In Section 5.2 we prove that Chromatic Number parameterized by the edge addition distance to a chordal graph (Chordal−ke) does not admit a polynomial kernel if $NP \nsubseteq coNP/poly$.

2 Preliminaries

Parameterized Complexity

A parameterized problem Q is a subset of $\Sigma^* \times \mathbb{N}$, with the natural number being the *parameter* providing structural information about the input. A parameterized problem Q is *fixed-parameter tractable* (or: in *FPT*) if $(x, k) \in Q$ can be decided in time $f(k)|x|^{O(1)}$ for some computable function f. A *kernelization algorithm* (Or: *kernel*) for a parameterized problem Q is a polynomial time algorithm which transforms (x, k) to (x', k'), with the properties that $(x, k) \in Q$ if and only if $(x', k') \in Q$ and $|x'|, k' \leq f(k)$ for some computable function f, which is the size of the kernel. If this function f is polynomial in k then this is a polynomial kernel.

Graph Theory

In this thesis we consider only finite, simple and undirected graphs. If G is a graph then $V(G)$ denotes its vertex set, and $E(G)$ denotes the edge set of G containing 2-element subsets of $V(G)$. We consider two vertices u and v connected to each other in G if $\{u, v\} \in E(G)$.

The following displays a number of graph notations used in this thesis.

- For a vertex $v \in V(G)$, $N_G(v)$ is a vertex set containing all vertices connected to v in graph G, also called the (open) neighborhood of v.
- The degree of a vertex x, denoted by $deg(x)$, is the size of the open neighborhood of x and thus $deg(x) = |N_G(x)|$.
- The closed neighborhood of $v \in V(G)$ is defined as $N_G[v] = N_G(v) \cup \{v\}$.
- Any vertex $x \in V(G)$ with $N_G[v] = V(G)$ is a universal vertex, and the vertex is called universal since it is connected to every other vertex in G.
- A subgraph of a graph G is a graph G^* with $V(G^*) \subseteq V(G)$ and $E(G^*) \subseteq E(G)$.
- The induced subgraph of a vertex set $S \subseteq V(G)$ in G is denoted by $G[S]$ and its vertex set is defined as $V(G[S]) = S$ and its edge set is defined as $\{u, v\} \in E(G[S])$ if and only if $u \in S$ and $v \in S$.
- For an edge set $S \subseteq E(G)$, $V(S) \subseteq V(G)$ is a vertex set for which $u \in V(S)$ if and only if there is an edge $\{u, v\} \in S$.
- For a vertex set $S \subseteq V(G)$, the graph $G[V(G) \setminus S]$ is denoted by $G - S$ and the remaining graph contains all vertices in $V(G)$ which are not in S and all edges $\{u, v\} \in E(G)$ with $u \notin S$ and $v \notin S$.
- Similarly, for an edge set $S \subseteq E(G)$, the graph $G[E(G) \setminus S]$ is denoted by $G - S$ and the remaining graph contains all vertices in $V(G)$ and all edges $\{u, v\} \in E(G)$ with $\{u, v\} \notin S$.
- For an edge set S, the graph $G[E(G) \cup S]$ is denoted by $G + S$ and the remaining graph contains all vertices and edges in G as well as all edges in S.

A *clique* in G is a set of vertices $S \subseteq V(G)$ with the property that for any two arbitrary vertices $u \in S$ and $v \in S$ there exists an edge $\{u, v\} \in E(G)$. An *independent* set is a set of vertices $S \subseteq V(G)$ with the property that for any two arbitrary vertices $u \in S$ and $v \in S$, $\{u, v\} \notin E(G)$. A *vertex cover* is a set of vertices $S \subseteq V(G)$ with the property that the set $R = V(G) \setminus S$ is an independent set in G. A *matching* for a graph G is a set of edges $S \subseteq E(G)$ such that each vertex in $V(S)$ occurs in exactly one edge in S and $|V(S)|$ is the size of the matching. A *perfect matching* for a graph G is a matching $S \subseteq E(G)$ with $V(S) = V(G)$

We can merge a number of graphs in different ways to create a new graph.

- We define a *(disjoint) union* of a number of graphs $A_1...A_n$ into a graph B as $V(B) = V(A_1) \cup V(A_2) \cup ... \cup V(A_n)$ with $V(A_1) \cap V(A_2) \cap ... \cap V(A_n) = \emptyset$ and $E(B) = E(A_1) \cup E(A_2) \cup ... \cup E(A_n)$.
- We define a *join* of a number of graphs $A_1...A_n$ into a graph B as $V(B) = V(A_1) \cup V(A_2) \cup ... \cup V(A_n)$ and $E(B) = E(A_1) \cup E(A_2) \cup ... \cup E(A_n) \cup S$, with S an edge set containing edges $\{u, v\}$ for all combinations of $u \in V(A_i)$ and $v \in V(A_j)$ with $i \neq j$.

A *path* in a graph is a series of vertices in the graph such that there exists an edge in the graph between each two consecutive vertices in the series, and the size of the path is the number of edges in the path. A *simple* path in a graph is a path in which each vertex is traversed at most once. A *cycle* is a path in a graph from a vertex to itself using every edge at most once and the size of the cycle is the size of the path.

We consider a number of different graph classes in this thesis.

- A graph is a *bipartite* graph if its vertex set $V(G)$ can be partitioned into two independent sets.
- A graph is a *tree* if there are no cycles in the graph and there exists a path from each vertex to each other vertex in the graph.
- A graph is a *forest* if it consists of a union of trees.
- A graph is a *cluster* graph if it consists of a union of cliques.
- A graph is a *cocluster* graph if it consists of a join of independent sets.
- A graph is a *cograph* if it is either a join or a union of cographs; a graph consisting of a single vertex is also a cograph.
- A graph is a *split* graph if its vertex set $V(G)$ can be partitioned into two sets X and Y such that X is a clique and Y is an independent set.
- A graph is a *chordal* graph if every cycle in the graph with size greater than three has a *chord*, i.e. an edge between two nonconsecutive vertices on the path of the cycle.

For more information on graph classes consult the book by Brandstädt, Le, and Spinrad [3].

There are a number of facts and naming conventions regarding trees that are used in this thesis:

- A tree T has a *root* vertex, which is the only vertex not to have a *parent* vertex.
- A vertex in T has one parent if it is not the root, and can have multiple *children* vertices.
- A vertex with no children in T is called a *leaf* and has a degree of one, the connected vertex being the parent of the leaf (except in the case that the vertex is the only vertex in the graph, in which case it is also the root of T).
- Every vertex on a simple path from a vertex $a \in V(T)$ to the root is called an ancestor of a; a parent vertex is thus an ancestor of all of its children and the root is an ancestor of all vertices in T.

- Every vertex on a simple path from a vertex $a \in V(T)$ to a leaf is called a descendant of a; every vertex in the graph is thus a descendant of the root, except the root itself.
- The height of a vertex $a \in V(T)$ (or: $height(a)$) is the longest simple path from a to a leaf traversing only descendants of a.

A cograph has a corresponding *cotree* T with vertices $V(T)$ and edges $E(T)$ such that every vertex in $V(G)$ is also a leaf in T, and every leaf in T is a vertex in $V(G)$. The induced subgraph of a cograph consisting of the vertices in the leaves of a subtree of T rooted at v, is denoted by its capital letter, in this case V. Each vertex $y \in V(T)$ is either a join vertex, a union vertex or a leaf:

- If y is a leaf or the only vertex in T: $y \in V(G)$.
- If y is a union vertex: for the children $a_1 ... a_n$ of y this means that for arbitrary vertices $u \in A_i$ and $v \in A_j$ with $i \neq j$, $\{u, v\} \notin E(G)$.
- If y is a join vertex: for the children $a_1 ... a_n$ of y this means that for arbitrary vertices $u \in A_i$ and $v \in A_j$ with $i \neq j$, $\{u, v\} \in E(G)$.

Chromatic Number

For natural numbers q we define $[q] := \{1, 2, ..., q\}$. A *proper q-coloring* for a graph G is a function $f : V(G) \rightarrow [q]$ such that vertices connected in G receive different numbers, and each number represents a color. The *chromatic number* of a graph G, denoted by $\chi(G)$, is the smallest number q for which the graph has a proper q-coloring. We can formally define the Chromatic Number problem as:

Chromatic Number
Input: An undirected graph G and a natural number q
Question: Is $\chi(G) \leq q$?

3 Rules

In this section a number of useful general lemmas and propositions are provided. A good example is Proposition 1 since this proposition can be used several times in this thesis. Proposition 1 is so useful because it states in a general way that any subgraph G' of a graph G has $\chi(G') \leq \chi(G)$.

Proposition 1. *For any graph G and subgraph G': $\chi(G') \leq \chi(G)$.*

Proof. Let G' be a subgraph of some graph G, and let $X := V(G) \setminus V(G')$ and $Y := E(G) \setminus E(G')$. Consider a coloring of G using $q = \chi(G)$ colors; removing X and Y from the colored graph gives a proper q-coloring of the graph $G - X - Y = G'$. Hence G' has a proper q-coloring, proving that $\chi(G') \leq \chi(G)$. □

Lemma 1 intuitively states that any vertex with degree less than q (which is the number of colors we are permitted to use) can be deleted without consequence for the chromatic number of the graph. We can use this lemma as a rule in all kernels in Section 4 since it applies to all graphs discussed in this thesis.

Lemma 1. *For any natural number q and vertex $x \in V(G)$ with $deg(x) < q$, $\chi(G) \leq q$ if and only if $\chi(G - x) \leq q$.*

Proof. If $\chi(G) \leq q$, then $\chi(G - x) \leq q$ by Proposition 1.

If $\chi(G - x) \leq q$, then $\chi(G) \leq q$. Proof: If $\chi(G - x) \leq q$, then there is a proper q-coloring of $G - x$. Consider such a coloring, and let S be the set of colors that are used for vertices in $N_G(x)$. As x has degree less than q in G, there are less than q colors in the set S. Hence there is a color c in $[q] \setminus S$. Now create a coloring of G as follows: assign all vertices except x the same color as in the coloring of $G - x$. Assign x the color c. The result is a proper q-coloring. To see this, consider an edge in G. If it has x as an endpoint, then the two endpoints of the edge receive different colors since color c does not occur on the neighbors of x. If the edge does not have x as an endpoint, then its two endpoints belong to graph $G - x$. Since we started from a proper q-coloring of $G - x$, the endpoints also receive different colors. Hence the endpoints of each edge receive different colors, proving that the q-coloring of G is proper. So G has a proper q-coloring, proving that $\chi(G) \leq q$. □

Lemma 2 proves that if two vertices have the exact same neighborhood, one of the vertices can be deleted without consequence for the chromatic number of the graph. This lemma can also be applied to all kernels in Section 4.

Lemma 2. *For any vertices $u \in V(G)$ and $v \in V(G)$ with $N_G(u) = N_G(v)$, $\chi(G) \leq q$ if and only if $\chi(G - v) \leq q$.*

Proof. If $\chi(G) \leq q$, then $\chi(G - v) \leq q$ by Proposition 1.

If $\chi(G - v) \leq q$, then $\chi(G) \leq q$. Proof: If $\chi(G - v) \leq q$, then there is a proper q-coloring of $G - v$. Consider such a coloring, and let c be the color of u in this coloring. Now create a coloring of G as follows: assign all vertices except v the same color as in the coloring of $G - v$. Assign v the color c. The result is a proper q-coloring. To see this, consider an edge in G. If it has v as an endpoint, then its other endpoint k is a neighbor of u since $N(u) = N(v)$. This means there is also an edge with k and u as endpoints, which both belong to graph $G - v$. Since we started with a proper q-coloring of $G - v$, u and k receive different colors, and because v and u receive the same color, v and k receive different colors. If the edge does not have v as an endpoint, then its two endpoints belong to graph $G - v$. Since we started from a proper q-coloring of $G - v$, the endpoints also receive different colors. Hence the endpoints of each edge receive different colors, proving that the q-coloring of G is proper. So G has a proper q-coloring, proving that $\chi(G) \leq q$. □

Lemma 3 proves that deleting a universal vertex from a graph G results in a graph G' with $\chi(G') = \chi(G) - 1$. This Lemma can also be applied to all kernels in Section 4.

Lemma 3. *For any universal vertex $x \in V(G)$, $\chi(G) \leq q$ if and only if $\chi(G - x) \leq q - 1$.*

Proof. Since a universal vertex $x \in V(G)$ is connected to every other vertex in G it must have a unique color compared to the rest of the vertices in any proper coloring of G, and thus removing a universal vertex from a graph will decrease the chromatic number of the graph by 1. □

Proposition 2 can be used to prove an upper bound on the chromatic number of a graph.

Proposition 2. *For any graph G and $X \subseteq E(G)$, $\chi(G) \leq \chi(G - X) + |X|$.*

Proof. Since adding an edge to a graph can increase the chromatic number of the graph by at most 1, adding the $|X|$ edges in X to $G - X$ increases the chromatic number by at most $|X|$. After adding X to $G - X$ we end up with G, and thus $\chi(G) \leq \chi(G - X) + |X|$. □

In Proposition 3 we prove that if there are a great number of edges present in a *bipartite* graph B with a partition of its vertices into sets L and R such that $|L| = |R|$, there exists a perfect matching. We can use this proposition in Section 4.4.

Proposition 3. *For any bipartite graph B with $L \subseteq V(B)$, $R := V(B) \setminus L$, $|L| = |R|$ and $|E(B)| > (0.5|V(B)|)^2 - (0.5|V(B)|)$, there exists a perfect matching.*

Proof. Since $|E(B)| > (0.5|V(B)|)^2 - (0.5|V(B)|)$ there are always less than $0.5|V(B)|$ non-edges between L and R. This means each vertex in L is connected to at least 1 vertex in R. If $|E(B)| = (0.5|V(B)|)^2$ then each vertex in L is connected to each vertex in R and any matching between all vertices in L and all vertices in R is a legal perfect matching. If $|E(B)| < (0.5|V(B)|)^2$ then there is at least 1 vertex $x \in L$ that is not connected to some vertex in R. Since each vertex in L is connected to at least 1 vertex in R, match x to an arbitrary vertex $y \in R$ with $\{x, y\} \in E(B)$. Delete x and y from the graph and find the next matching; repeat this process until all vertices are matched. It is easy to see that this works if we look at the amount of non-edges between L and R. The amount of non-edges in B is smaller than $0.5|V(B)|$ and since we delete a vertex with at least 1 non-edge to get $B' = B - \{x, y\}$ the amount of non-edges in B' is smaller than $0.5|V(B)| - 1 = 0.5|V(B')|$. This means $|E(B')| > (0.5|V(B')|)^2 - (0.5|V(B')|)$ which implies that each vertex in $L' \subseteq V(B')$ is connected to at least 1 vertex in $R' \subseteq V(B')$. Now we can keep finding new matches in resulting graphs while the property $|E(B)| > (0.5|V(B)|)^2 - (0.5|V(B)|)$ holds for any resulting graph B, until $V(B) = \emptyset$ at which time all vertices are matched proving this proposition. □

4 Upper Bounds

In this section the upper bounds described in this thesis are proved. Section 4.1 provides a nice upper bound for Chromatic Number parameterized by the edge deletion distance to a forest (Forest+ke). Section 4.2 then provides an upper bound for Chromatic Number parameterized by the edge modification distance to a cluster graph (Cluster±ke). In Section 4.3 Chromatic Number parameterized by the edge modification distance to a cocluster graph (CoCluster±ke) is provided with an upper bound. Lastly, Section 4.4 provides an upper bound for Chromatic Number parameterized by the edge deletion distance to a cograph (Cograph+ke).

4.1 Forest+ke

In this section a kernel for Chromatic Number on Forest+ke is provided. We define Chromatic Number parameterized by the edge deletion distance to a forest as follows.

CHROMATIC NUMBER on FOREST+ke
Input: An undirected graph G, a natural number q and a modulator $X \subseteq E(G)$ such that $G - X \in$ FOREST.
Parameter: The size $k := |X|$ of the modulator.
Question: Is $\chi(G) \leq q$?

The only tool needed to construct a polynomial kernel for CHROMATIC NUMBER on FOREST+ke is Lemma 1, and Lemma 4 provides such a kernel.

Lemma 4. *CHROMATIC NUMBER on FOREST+ke has a kernel with at most $2k$ vertices and exhaustive application of Lemma 1 gives us such a kernel.*

Proof. If $q < 3$ the problem instance is trivial, since if $q = 1$ only an independent set is colorable and if $q = 2$ a greedy coloring is an optimal coloring and the problem is thus polynomial solvable. So we assume $q \geq 3$. After exhaustive application of Lemma 1 each vertex $x \in V(G)$ has $deg(x) \geq q \geq 3$. Since each edge has 2 endpoints and each vertex is an endpoint of at least 3 edges, $|E(G)| \geq \frac{3|V(G)|}{2}$. A forest has at most $|V(G)| - 1$ edges, and thus $k \geq \frac{3|V(G)|}{2} - (|V(G)| - 1) = \frac{|V(G)|}{2} + 1$. This means $2k > |V(G)|$, which proves this is a kernel with at most $2k$ vertices.

Lemma 1 runs in polynomial time and reduces the graph by one vertex when applied, ensuring the lemma can only be applied $|V(G)|$ times. This means the kernel runs in polynomial time. □

The kernel described in Lemma 4 proves that after application of the kernel on an instance of CHROMATIC NUMBER on FOREST+ke the resulting instance contains at most $2k$ vertices. This kernel will however not see much practical use since the parameter in most instances of CHROMATIC NUMBER on FOREST+ke will be very large because there are only a small amount of edges in a forest (at most the number of vertices minus one).

4.2 Cluster±ke

In this section a kernel for CHROMATIC NUMBER on CLUSTER±ke is provided. We define CHROMATIC NUMBER parameterized by the edge modification distance to a cluster graph as follows.

CHROMATIC NUMBER on CLUSTER±ke
Input: An undirected graph G, a natural number q, a modulator $X \subseteq E(G)$ and a modulator $Y \subseteq \binom{V(G)}{2} \setminus E(G)$ such that $G - X + Y \in$ CLUSTER.
Parameter: The size $k := |X| + |Y|$ of the modulators.
Question: Is $\chi(G) \leq q$?

To obtain a polynomial kernel for CHROMATIC NUMBER parameterized by the edge modification distance to a cluster graph we need some rules the kernel can use to properly reduce the size of G. For ease of use we define G^* to be the induced subgraph of G containing only those vertices not represented in $V(X) \cup V(Y)$, formally defined in Definition 1. Since G^* only contains vertices not in $V(X) \cup V(Y)$, G^* is a cluster graph.

Definition 1. $G^* := G[V(G) \setminus (V(X) \cup V(Y))]$

By applying Rule 1 we know the largest clique in G^* is smaller or equal to q. We need this rule in our kernel to be able to prove the kernel is polynomial in size.

Rule 1. *If the largest clique in G^* is bigger than q then reduce to $(K, \emptyset, 1)$, with K a graph containing two connected vertices.*

Lemma 5. *Rule 1 is sound.*

Proof. In a clique each color can only occur once, since every vertex is connected to every other vertex. This means the chromatic number of a clique is the size of the clique, and if the clique size is larger than q, then $\chi(G^*) > q$. Since $V(G^*) \subseteq V(G)$, and since the chromatic number can never increase when taking an induced subgraph (Proposition 1), $\chi(G) > q$ and (G, q) is a No-instance. Also, $(K, \emptyset, 1)$ is a No-instance since two connected vertices can never get a legal coloring using only one color. □

Rule 2 reduces the size of the kernel by taking one vertex from every clique in G^* and deleting those vertices from the graph. The intuition behind this rule is that if q is big enough, there is a color only used for coloring the vertices in G^*. If there is a color c only used to color vertices in G^*, we know that since G^* is a cluster graph, c can be used to color one vertex in every clique in G^*, which is why the rule deletes an arbitrary vertex from each clique in G^*.

Rule 2. *If $q > |V(X) \cup V(Y)|$ and $V(G^*) \neq \emptyset$ reduce to $(G - S, X \setminus S, q - 1)$, with S a maximum independent set of G^*.*

Lemma 6. *Rule 2 can be applied in polynomial time and is sound when applied to* Chromatic Number *on* Cluster$\pm ke$.

Proof. If (G, X, q) is a Yes-instance, then $(G - S, X \setminus S, q - 1)$ is a Yes-instance. Proof: If (G, X, q) is a Yes-instance, then by the problem definition there is a proper q-coloring of G. Consider such a coloring, and let T be the set of colors that are used for vertices in $V(X) \cup V(Y)$. Since $q > |T|$ there is a color $c \in [q] \setminus T$. Now recolor some vertices of G as follows: for every connected component in G^* that does not contain a vertex that receives the color c, recolor an arbitrary vertex in that connected component with the color c. The result is a proper q-coloring for G. To see this, consider an edge in G.

- If the edge has a vertex x which has been recolored to c as an endpoint, and the other endpoint y is in G^*, they must be in the same connected component in G^* and thus y is not colored c resulting in both endpoints receiving a different color.
- If the edge has a vertex x which has been recolored to c as an endpoint, and the other endpoint y is in $V(X) \cup V(Y)$, y does not receive c as its color by definition so both endpoints receive a different color.
- Since we started with a proper q-coloring before recoloring, any edge with both endpoints not recolored must still have different colors on both endpoints.

Now create a coloring of $G - S'$ as follows: assign all vertices that receive the color c to the set S'. Removing S' from the recolored graph now gives a proper $(q-1)$-coloring of the graph $G - S'$. Since S' contains a vertex from each connected component in G^* and G^* is a cluster graph with every connected component a clique, S' is a maximum independent set of G^*. Both S and S' contain one vertex from every clique in G^*, and if two vertices x and y are in the same clique in G^*, then $N_G[x] = N_G[y]$. This means that $G - x$ and $G - y$ are isomorphic, and thus $G - S$ and $G - S'$ are isomorphic. Since $G - S'$ has a proper $(q-1)$-coloring, $G - S$ has a proper $(q-1)$-coloring using isomorphism, proving that $(G - S, X \setminus S, q - 1)$ is a YES-instance.

If $(G - S, X \setminus S, q - 1)$ is a Yes-instance, then (G, X, q) is a Yes-instance. Proof: If $(G - S, X \setminus S, q - 1)$ is a Yes-instance, then by the problem definition there is a proper $(q-1)$-coloring of $G - S$. Consider such a coloring, and let c be a color not used in this $q - 1$ coloring. Now create a coloring of G as follows: assign all vertices except vertices in the set S the same color as in the coloring of $G - S$. Assign all vertices in the set S the color c. The result is a proper q-coloring. To see this, consider an edge in G. If it has a vertex $x \in S$ as an endpoint, then its other endpoint k is not in set S since S is an independent set. Furthermore since the color c does not occur in $G - S$ and k does occur in $G - S$, k and x receive different colors. If the edge does not have an endpoint which is in set S, then its two endpoints belong to graph $G - S$. Since we started from a proper $(q-1)$-coloring of $G - S$, the endpoints also receive different colors. Hence the endpoints of each edge receive different colors, proving that the q-coloring of G is proper. So G has a proper q-coloring, proving that (G, X, q) is a Yes-instance.

The independent set S is created by taking one arbitrary vertex from every connected component in G^*, and is thus composed in polynomial time ensuring this rule can be applied in polynomial time. □

Lemma 7 provides a polynomial kernel for Chromatic Number parameterized by the edge modification distance to a cluster graph by using the rules provided in this section as well as the lemmas provided in Section 3.

Lemma 7. Chromatic Number *on* Cluster$\pm ke$ *has a kernel with at most* $4k^2 + 2k$ *vertices and exhaustive application of Rules and Lemmas 1, 2, 3, 1 and 2 gives us such a kernel.*

Proof. After exhaustive application of Rule 2 either $q \leq |V(X) \cup V(Y)|$ or $V(G^*) = \emptyset$.

If $V(G^*) = \emptyset$ then $G = V(X) \cup V(Y)$ and thus G contains at most $2k$ vertices.

If $q \leq |V(X) \cup V(Y)|$ then $q \leq 2k$. For any $x \in V(G^*)$ and with $G^c = G - X + Y$, $N_{G^c}[x] \neq N_{G^*}[x]$ since if $N_{G^c}[x] = N_{G^*}[x]$ and $|N_{G^*}[x]| > q$, (G, q) will be reduced to $(K, 1)$ with K a graph containing two connected vertices following Rule 1. If $N_{G^c}[x] = N_{G^*}[x]$ and $|N_{G^*}[x]| \leq q$, then $deg(x) < q$ and Lemma 1 is applicable on graph G until there are no more vertices with these properties. This means that every connected component in G^c has at least one vertex in $V(X) \cup V(Y)$, and thus there are a maximum of $2k$ connected components in $N_{G^c}[x]$. Then there are also a maximum of $2k$ connected components in G^* and following Rule 1 each clique has size at most q. Since $q \leq 2k$ this means there are at most $4k^2$ vertices in G^*, and seen as $G^* \cup V(X) \cup V(Y) = G$ and $V(X) \cup V(Y) \leq 2k$, G contains at most $4k^2 + 2k$ vertices.

All rules and lemmas applied in this kernel run in polynomial time. Also all rules and lemmas reduce the graph by at least one vertex when applied, and thus the rules and lemmas can be applied at most $|V(G)|$ times. This means the kernel runs in polynomial time. □

The kernel described in Lemma 7 proves that after application of the kernel on an instance of Chromatic Number on Cluster$\pm ke$ the resulting instance contains at most $4k^2 + 2k$ vertices. For most graphs the parameter for Chromatic Number on CLUSTER±ke will be very large. It might however be worth researching which kind of graphs have a small parameter for CHROMATIC NUMBER on CLUSTER±ke, for instance whether it are sparse graphs or dense graphs.

4.3 CoCluster±ke

In this section a kernel for CHROMATIC NUMBER on COCLUSTER±ke is provided. We define CHROMATIC NUMBER parameterized by the edge modification distance to a cocluster graph as follows.

> CHROMATIC NUMBER on COCLUSTER±ke
> **Input:** An undirected graph G, a number q, a modulator $X \subseteq E(G)$ and a modulator $Y \subseteq \binom{V(G)}{2} \setminus E(G)$ such that $G - X + Y \in$ COCLUSTER.
> **Parameter:** The size $k := |X| + |Y|$ of the modulators.
> **Question:** Is $\chi(G) \leq q$?

To obtain a polynomial kernel for CHROMATIC NUMBER parameterized by the edge modification distance to a cocluster graph we only need the lemmas in Section 3. Lemma 8 shows that a combination of Lemmas 3 and 2 can result in a large reduction of the number of vertices in G. The intuition behind Lemma 8 is that since a cocluster graph is a join of a number of independent sets, every vertex in an independent set has the same neighborhood and thus all vertices except one can be deleted using Lemma 2. If only a single vertex remains in such an independent set then that vertex is a universal vertex in a cocluster graph and can thus be deleted using Lemma 3.

Lemma 8. *If for some x $(G-N(x)) \cap (V(X) \cup V(Y)) = \emptyset$ where G is a* COCLUSTER±ke *graph, exhaustive application of Lemmas 3 and 2 will reduce (G, X, q) to $(G - (G - N(x)), E(G - (G - N(x))) \cap X, q - 1)$.*

Proof. Since $(G - N(x)) \cap (V(X) \cup V(Y)) = \emptyset$ the subgraph $G - N(x)$ is a cocluster graph, and since x has no neighbors in this subgraph it is an independent set following the definition of a cocluster from which follows that if a number of vertices are not adjacent to some vertex, they form an independent set. Every vertex in $G - N(x)$ is connected to every vertex in $G-(G-N(x))$ again following the definition of a cocluster graph which states that independent sets are joined with all other independent sets. Lemma 2 now works on any two vertices y and z within $G - N(x)$ since $N(y) = N(z)$, and after exhaustive application of Lemma 2 only one vertex will remain in $G - N(x)$. Since the one remaining vertex in $G - N(x)$ is connected to every vertex in $G - (G - N(x))$, the vertex is universal and will thereby be removed from the graph by Lemma 3, which decreases the chromatic number by 1. This yields the reduction of (G, X, q) to $(G - (G - N(x)), E(G - (G - N(x))) \cap X, q - 1)$. □

Lemma 9 provides a polynomial kernel for CHROMATIC NUMBER parameterized by the edge modification distance to a cocluster graph by only applying lemmas provided in Section 3.

Lemma 9. CHROMATIC NUMBER *on* COCLUSTER±ke *has a kernel with at most* $4k$ *vertices and exhaustive application of Lemmas 2 and 3 gives us such a kernel.*

Proof. We define the *coclique* of a vertex x to be all vertices in $G - N_{G^{cc}}(x)$, with $G^{cc} = G - X + Y$. If $(G - N_{G^{cc}}(x)) \cap (V(X) \cup V(Y)) = \emptyset$ then following Lemma 8 G is reduced to $G - (G - N(x))$, since $G - N_{G^{cc}}(x) \cap (V(X) \cup V(Y)) = \emptyset$ implies $(G - N(x)) \cap (V(X) \cup V(Y)) = \emptyset$. This means that after exhaustive application of the lemmas used in Lemma 8 every coclique contains at least one vertex from $(V(X) \cup V(Y))$, which means there are at most $2k$ cocliques. A coclique contains at most one vertex not in $(V(X) \cup V(Y))$ after extensive application of Lemma 2. To see this, consider the case that there are multiple vertices in a coclique which are all not in $(V(X) \cup V(Y))$. These vertices form an independent set with the same neighborhood for all vertices following the definition of a cocluster, which states that independent sets are joined to all other vertices. In this scenario Lemma 2 is applicable until only one vertex remains, proving at most one vertex which is not in $V(X) \cup V(Y)$ remains per coclique. Since there are at most $2k$ cocliques and every coclique contains at most one vertex not in $V(X) \cup V(Y)$, there are at most $2k$ vertices not in $V(X) \cup V(Y)$ left over after kernelization. There are at most $2k$ vertices in $V(X) \cup V(Y)$, so the graph contains at most $4k$ vertices after kernelization.

All lemmas applied in this kernel run in polynomial time. Also all lemmas reduce the graph by at least one vertex when applied, and thus the lemmas be applied at most $|V(G)|$ times. This means the kernel runs in polynomial time. □

The kernel described in Lemma 9 proves that after application of the kernel on an instance of Chromatic Number on CoCluster$\pm ke$ the resulting instance contains at most $4k$ vertices. For most graphs the parameter for Chromatic Number on CoCluster$\pm ke$ will be very large. As in Section 4.2, it might be worth researching which kind of graphs have a small parameter for Chromatic Number on CoCluster$\pm ke$.

4.4 Cograph+ke

In this section a kernel for Chromatic Number on Cograph$+ke$ is provided. We define Chromatic Number parameterized by the edge deletion distance to a cograph as follows.

Chromatic Number on Cograph$+ke$
Input: An undirected graph G, a number q, a modulator $X \subseteq E(G)$ such that $G - X \in$ Cograph and a cotree T obtained from $G - X$.
Parameter: The size $k := |X|$ of the modulator.
Question: Is $\chi(G) \leq q$?

To obtain a polynomial kernel for Chromatic Number parameterized by the edge deletion distance to a cograph we need a great number of rules the kernel can use to properly reduce the size of G, as well as a great number of lemmas to prove a bound on the size of the kernel. This section gives all lemmas and rules needed for the kernel provided at the end of this section in Lemma 24.

In Definition 2 a perfect vertex in the cotree is defined. Intuitively a perfect vertex in the cotree means that in the case of a union vertex $u \in V(T)$, two children $a \in V(T)$ and $b \in V(T)$ of u with their respective vertex sets A and B will not have any edges between A and B in G, and thus not just in the cograph $G - X$. Similarly, in the case of a perfect join vertex $j \in V(T)$, two children $a \in V(T)$ and $b \in V(T)$ of j with their respective vertex sets A and B are in a join with each other in G (meaning all vertices in A are connected to all vertices in B in G). In Proposition 10 we prove that every join vertex in T is perfect.

Definition 2. *A vertex $y \in T$ is either a leaf, a union or a join.*

- *If y is a leaf then y is perfect if $y \notin V(X)$.*
- *If y is a union vertex then y is perfect if for the children $a_1 ... a_n$ of y, for arbitrary vertices $u \in A_i$ and $v \in A_j$ with $i \neq j$, $\{u, v\} \notin E(G)$ and thus $\{u, v\} \notin X$.*
- *If y is a join vertex then y is perfect if for the children $a_1 ... a_n$ of y, for arbitrary vertices $u \in A_i$ and $v \in A_j$ with $i \neq j$, $\{u, v\} \in E(G)$.*

Lemma 10. *Every join vertex $j \in V(T)$ is a perfect join vertex.*

Proof. Suppose j is not a perfect join vertex. Following the definition of perfect join vertices there are two children $a \in V(T)$ and $b \in V(T)$ of j with a $y \in A$ and a $z \in B$ such that $\{y, z\} \notin E(G)$. Since the cograph is based on $G - X$ we know $\{y, z\} \in E(G - X)$. Also, $E(G - X) \subseteq E(G)$ and thus $\{y, z\} \in E(G)$, proving a scenario with $\{y, z\} \notin E(G)$ can not occur and thus proving j is a perfect join vertex. □

To give a nice lower bound on the chromatic number of $G[Y]$, with Y obtained from a vertex $y \in V(T)$ with only perfect ancestors, we compute the chromatic number of $G[Y] - X$ and call it $lb(y)$. We can easily compute the chromatic number of $G[Y] - X$ since it is a cograph, and we know $\chi(G[Y] - X) \leq \chi(G[Y])$ by using Proposition 1. To give an upper bound on $\chi(G[Y])$, denoted by $ub(y)$, we use $ub(y) = lb(y) + |E(G[Y]) \cap X|$ since we know $ub(y) = lb(y) + |E(G[Y]) \cap X| \geq \chi(G[Y])$ by using Proposition 2.

Proposition 4 intuitively states that any two vertices that share a common ancestor $y \in V(T)$ with only perfect ancestors, are only connected to a vertex $z \in V(T)$, which does not have y as an ancestor, if both vertices are connected to z.

Proposition 4. *For any two leaf vertices $u_1 \in V(T)$ and $u_2 \in V(T)$ with join/union vertex $u \in V(T)$ as their common ancestor with lowest height and only perfect ancestors, for any leaf vertex $v \in V(T)$ for which $v \notin U$, $\{u_1, v\} \in E(G)$ if and only if $\{u_2, v\} \in E(G)$.*

Proof. Since v is not in the subtree of u, if $\{u_1, v\} \in E(G)$ then the common ancestor with lowest height of v and u_1 in the cotree is a perfect join vertex since there is an edge between v and u_1 and all ancestors of u are perfect. The common ancestor with lowest height of v and u_2 is the same perfect join vertex, ensuring that $\{u_2, v\} \in E(G)$. Symmetrically, if $\{u_2, v\} \in E(G)$ then the common ancestor with lowest height of v and u_2 in the cotree is a perfect join vertex and the common ancestor with lowest height of v and u_1 is the same perfect join vertex, ensuring that $\{u_1, v\} \in E(G)$ and proving $\{u_1, v\} \in E(G)$ if and only if $\{u_2, v\} \in E(G)$. □

Intuitively Proposition 5 proves that if a perfect leaf vertex x shares a common ancestor with height a with a leaf vertex $y \in V(T)$ in G, and if $\{x, z\} \in E(G)$ for a leaf vertex $z \in V(T)$ in G for which the common ancestor of x and z with lowest height has height $b > a$, then $\{y, z\} \in E(G)$.

Proposition 5. *For a perfect leaf vertex $u_1 \in V(T)$ and leaf vertex $u_2 \in V(T)$ with join/union vertex $u \in V(T)$ as their common ancestor with lowest height and any leaf vertex $v \in V(T)$ for which $v \notin U$, if $\{u_1, v\} \in E(G)$ then also $\{u_2, v\} \in E(G)$.*

Proof. Following the definition of a perfect leaf vertex in T, if $\{u_1, v\} \in E(G)$ then $\{u_1, v\} \notin X$. This means the common ancestor of u_1 and v with lowest height is a join vertex, and all join vertices are perfect following Proposition 10. The common ancestor with lowest height of v and u_2 is the same perfect join vertex, ensuring that $\{u_2, v\} \in E(G)$. □

Rule 3 says that if there are join vertices $j_1 \in V(T)$ and $j_2 \in V(T)$, which are both children of union vertex $u \in V(T)$ and only have a small amount of edges between vertex sets V_1 and V_2 in G, then under certain conditions we can delete the edges between J_1 and J_2 in G. This rule is a tool to try to make an imperfect union vertex perfect when exhaustively applied to all children of an imperfect union vertex. Lemma 12 is used to describe the effect of exhaustive application of Rule 3 to the children of an union vertex. The lemma states that after exhaustive application either the union vertex is perfect or $\chi(G[U])$ is polynomially related to the number of edges which are in both X and $E(G[U])$.

Rule 3. *If a join vertex $j \in V(T)$ has a union vertex $u \in V(T)$ with only perfect ancestors and if $m < lb(u)$, with m the number of edges between j and the other children of u, then reduce to $(G - M, X \setminus M, q)$, with M the set containing all m edges between j and the other children of u.*

Lemma 11. *Rule 3 can be applied in polynomial time and is sound when applied to* Chromatic Number *on* Cograph+ke.

Proof. If (G, X, q) is a Yes-instance, then $(G - M, X \setminus M, q)$ is a Yes-instance by Proposition 1.

If $(G - M, X \setminus M, q)$ is a Yes-instance, then (G, X, q) is a Yes-instance. Proof: If $(G - M, X \setminus M, q)$ is a Yes-instance, then there is a proper q-coloring of $G - M$. Consider such a coloring, and let S be the set of colors that are used for vertices in U. Now consider all vertices with the same color as independent sets, and consider these $|S|$ independent sets for both $G[J]$ and $G[U] - J$ separately. If none of the vertices in an independent set in $G[J]$ are connected to the vertices of an independent set in $G[U] - J$, these two independent sets form an independent set in G and all vertices in the newly formed combined set can thus get the same color. If we can match all independent sets in $G[J]$ to a separate/unique independent set in $G[U] - J$, we know that there is a coloring of $G[U]$ in at most $|S|$ colors because assigning each matched combined independent set in U an arbitrary color from S and assigning each remaining vertex in G the same color as in the coloring of $G - M$ results in a proper q-coloring of G. To see this, consider an edge in G.

- If the edge has both endpoints in $G[U]$ then both endpoints receive different colors since vertices with the same color are in the same independent set.
- If the edge has one endpoint in $G[U]$ and the other endpoint y in $G - U$ then following Proposition 4 all vertices in U are connected to y and since we started from a proper q-coloring of $G - M$ in which all S colors are used for the vertices in U, y receives a color not in S and thus both endpoints receive a different color.

- Any edge with both endpoints in $G - U$ must still have different colors on both endpoints since the endpoints receive the same color as in the proper q-coloring of $G - M$.

This means that we only need a matching between independent sets of $G[J]$ and $G[U]-J$ to get a proper q-coloring for G. Now consider this as a perfect matching problem between the independent sets in $G[J]$ and the independent sets in $G[U]-J$, where the independent sets can only be matched to each other if there is no edge between them. If $G[J]$ has less than $|S|$ independent sets we simply add empty sets (which are also independent) until $G[J]$ has $|S|$ independent sets; we do the same for $G[U] - J$. Now model this as a *bipartite* graph B in which there is a vertex for every independent set in $G[J]$ and in $G[U]-J$, and any vertex j which represents an independent set in $G[J]$ is only connected to a vertex u, which represents an independent set in $G[U]-J$, if the union of the two independent sets is itself an independent set in G. We define vertex set $L \subseteq V(B)$ as the vertex set containing all vertices which represent an independent set in J and similarly we define vertex set $R \subseteq V(B)$ as the vertex set containing all vertices which represent an independent set in $U - J$; both L and R are independent sets. Since $|S| = lb(u)$ and $t < lb(u)$ there are less than $|S| = 0.5|V(B)|$ non-edges between L and R. This means we now have a matching problem with $|V(B)| = 2|S|$ and $|E(B)| > (0.5|V(B)|)^2 - (0.5|V(B)|)$. Following Proposition 3 there exists a perfect matching and it can be found in polynomial time; consider such a matching. Using the perfect matching and the coloring for $G - M$ we can now as previously mentioned obtain a proper q-coloring for G. So G has a proper q-coloring, proving that (G, X, q) is a Yes-instance.

Since the set M can be found in polynomial time using T, the rule can be applied in polynomial time. □

Lemma 12. *If after exhaustive application of Rule 3 on the children join vertices of a union vertex $u \in V(T)$ with only perfect ancestors there are still edges between the join vertices, then $\chi(G[U]) \leq 2|E(G[U]) \cap X|$.*

Proof. Since there are still edges between join vertices, it must be the case that there is a child join vertex of u which has $t \geq lb(u)$ outgoing edges to other children join vertices of u. Also $\chi(G[U]) \leq ub(u) = lb(u) + |E(G[U]) \cap X|$ and since $lb(u) \leq t \leq |E(G[U]) \cap X|$, $\chi(G[U]) \leq 2|E(G[U]) \cap X|$ which proves this lemma. □

If $ub(j)$, with j the child of a perfect union vertex with only perfect ancestors, is low enough, meaning the chromatic number of $\chi(G[J])$ is low enough, Rule 4 states we can delete all vertices J from G with no consequence to the chromatic number of G.

Rule 4. *If a join vertex $j \in V(T)$ with parent union vertex $u \in V(T)$ only has perfect ancestors and if $lb(u) \geq ub(j)$ and $lb(u) > lb(j)$, then reduce to $(G-J, E(G-J) \cap X, q)$.*

Lemma 13. *Rule 4 can be applied in polynomial time and is sound when applied to* Chromatic Number *on* Cograph+*ke*.

Proof. If (G, X, q) is a Yes-instance, then $(G - J, E(G - J) \cap X, q)$ is a Yes-instance by Proposition 1.

If $(G - J, E(G - J) \cap X, q)$ is a Yes-instance, then (G, X, q) is a Yes-instance. Proof: If $(G - J, E(G - J) \cap X, q)$ is a Yes-instance, then there is a proper q-coloring of $G - J$. Consider such a coloring, and let S be the set of colors that are used for vertices in U. Since $lb(u) \leq \chi(G[U])$, $|S| \geq lb(u)$. Also $lb(u) \geq lb(j) + |E(G[J]) \cap X|$ and $lb(j) + |E(G[J]) \cap X| \geq \chi(G[J])$, so $lb(u) \geq \chi(G[J])$. This means $|S| \geq \chi(G[J])$, and thus there are enough colors in S to color $G[J]$. Consider an arbitrary proper coloring of $G[J]$ using only colors in S. Now create a coloring of G as follows: assign all vertices except vertices in J the same color as in the coloring of $G - J$. Assign the vertices in J the same color as in the arbitrary coloring of $G[J]$. The result is a proper q-coloring. To see this, consider an edge in G.

- If it has an endpoint in J its other endpoint is either also in J or in $V(G - U)$ since the perfect union vertex u ensures vertices in J are not connected to any vertex in $U \setminus J$.
  - If the edge has both endpoints in J the endpoints receive different colors, since we started with a proper coloring of $G[J]$.
  - If the edge has an endpoint in J and its other endpoint l is in $G - U$, then l is also connected to all vertices in $U \setminus J$ using Proposition 4 and the fact that all ancestors of u are perfect. Since all colors in S are used in $U \setminus J$, l will receive a color not in S and thus both endpoints receive different colors.
- If the edge has both endpoints in $G - J$ the endpoints receive different colors, since we started with a proper q-coloring of $G - J$.

Hence the endpoints of each edge receive different colors, proving that the q-coloring of G is proper. So G has a proper q-coloring, proving that (G, X, q) is a Yes-instance.

All information needed to apply this rule can be extracted from T in constant time ensuring the rule can be applied in polynomial time. □

Lemma 14 states that if there is a join child $j \in V(T)$ of a perfect union vertex with only perfect ancestors left after exhaustive application of Rule 4, and the join child has a polynomial relationship between $\chi(G[J])$ and the multiplication of $height(j)$ and the number of edges both in X and $E(G[J])$, the polynomial relation exists for the parent union vertex. Similarly, Lemma 15 proves that for a vertex $j \in V(T)$ under some conditions there exists a polynomial relation between $\chi(G[J])$ and the multiplication of $height(j)$ and the number of edges both in X and $E(G[J])$, only this time based on all of the children union vertices of join vertex j.

Lemma 14. *If Rule 4 is applied to all children join vertices of a perfect union vertex $u \in V(T)$ with only perfect ancestors and if one of the remaining children join vertices $j \in V(T)$ has $\chi(G[J]) \leq (height(j)+1) \times |E(G[J]) \cap X|$, then $\chi(G[U]) < (height(u)+1) \times |E(G[U]) \cap X|$.*

Proof. For some child join vertex $l \in V(T)$ of u, $ub(u) = lb(l) + |E(G[L]) \cap X|$ since because u is perfect, $\chi(G[U])$ is the maximum of all $\chi(G[Z])$ of the children $z \in V(T)$ of u. Because $lb(u) < lb(j) + |E(G[J]) \cap X|$ and $lb(u) \geq lb(l)$, $lb(l) < lb(j) + |E(G[J]) \cap X|$. Since $\chi(G[U]) \leq ub(u)$ and $lb(l) < lb(j) + |E(G[J]) \cap X|$, $\chi(G[U]) < lb(j) + |E(G[J]) \cap X| + |E(G[L]) \cap X|$. Also $lb(j) \leq \chi(G[J]) \leq (height(j) + 1) \times |E(G[J]) \cap X|$, so $\chi(G[U]) < (height(j) + 1) \times |E(G[J]) \cap X| + |E(G[J]) \cap X| + |E(G[L]) \cap X|$. Since $|E(G[J]) \cap X| + |E(G[L]) \cap X| \leq |E(G[U]) \cap X|$, we get $\chi(G[U]) < (height(j) + 1) \times |E(G[J]) \cap X| + |E(G[U]) \cap X|$. Also $|E(G[J]) \cap X| \leq |E(G[U]) \cap X|$ so $\chi(G[U]) < (height(j) + 2) \times |E(G[U]) \cap X|$. Also u is the parent vertex of j, so $height(u) \geq height(j) + 1$ and thus $\chi(G[U]) < (height(u) + 1) \times |E(G[U]) \cap X|$ proving this lemma. □

Lemma 15. *If all children union vertices $u \in V(T)$ of a join vertex $j \in V(T)$ with only perfect ancestors adhere to $\chi(G[U]) < (height(u) + 1) \times |E(G[U]) \cap X|$, then $\chi(G[J]) < (height(j) + 1) \times |E(G[J]) \cap X|$.*

Proof. Let S be the vertex set containing all children of j. Now $\sum_{u \in S} |E(G[U]) \cap X| = |E(G[J]) \cap X|$ since $\cup_{u \in S} E(G[U]) \subseteq E(G[J])$, and $(E(G[J]) \setminus \cup_{u \in S} E(G[U])) \cap X = \emptyset$ since j is a perfect join vertex. Also $height(j) = \max_{u \in S} height(u) + 1$ and $\chi(G[J]) = \sum_{u \in S} \chi(G[U])$. Since $\chi(G[U]) < (height(u) + 1) \times |E(G[U]) \cap X|$ for all $u \in S$ this results in $\chi(G[J]) < (height(j) + 1) \times |E(G[J]) \cap X|$ proving this lemma. □

In Definition 3 the color lowering set is defined. This is a tool to describe a vertex set that decreases the chromatic number of the graph by exactly one when deleted from the graph. Lemmas 16, 17 and 18 show how a color lowering set for a vertex in the cotree can be determined.

Definition 3. *A* color lowering set *is a set of vertices S in a subgraph $G[V]$ obtained from a vertex v in T with only perfect ancestors such that $\chi(G[V] - S) = \chi(G[V]) - 1$.*

Lemma 16. *A leaf vertex v in T with only perfect ancestors has a color lowering set containing the only vertex in V, which is v.*

Proof. The subgraph $G[V]$ contains exactly 1 vertex which is v so $\chi(G[V]) = 1$. Since the chromatic number of an empty set is 0 we get $\chi(G[V] - \{v\}) = 0$, and thus $\chi(G[V] - \{v\}) = \chi(G[V]) - 1$ proving this lemma. □

Lemma 17. *If all children join vertices of a union vertex u in T have a color lowering set then u has a color lowering set.*

Proof. Since the children of u have a color lowering set u is a perfect union vertex with only perfect ancestors by Definition 3. We can get $\chi(G[U])$ by taking the maximum chromatic number of all subgraphs obtained from the children of u since u is perfect. This means the union of all color lowering sets of all children of u will result in a color lowering set for u since all children of u will have their chromatic number decreased by 1 when their color lowering sets are removed from the graph and thus the maximum chromatic number over all children will decrease by 1, proving this lemma. □

Lemma 18. *If a join vertex j in T has a child union vertex u with a color lowering set then j has a color lowering set.*

Proof. Since u has a color lowering set j is a perfect join vertex with only perfect ancestors by Definition 3. We can get $\chi(G[J])$ by adding the chromatic number of all subgraphs obtained from the children of j since j is perfect. This means that the color lowering set of u is a color lowering set for j since decreasing the chromatic number of u by 1 decreases the chromatic number of j by 1, proving this lemma. □

Lemma 19 is very important since it states that after exhaustive application of Rules 3 and 4 on T *each* vertex in T is one of three things, one of which is that the vertex has a color lowering set. If it follows from Lemma 19 that the root of T has a color lowering set then we can use Rule 5 to delete the complete color lowering set from G, thereby reducing the chromatic number of G by one.

Lemma 19. *After exhaustive application of Rules 3 and 4 on T a vertex $v \in V(T)$ either has an imperfect ancestor, a color lowering set or $\chi(G[V]) \leq (height(v)+1) \times |E(G[V]) \cap X|$.*

Proof. This proof is by induction on the height of v in T.

- If $height(v) = 0$ then v a leaf in the cotree; if v only has perfect ancestors then by Lemma 16 v has a color lowering set.
- If $height(v) > 0$ then v is either a join or a union vertex.
  - If v is a join vertex with only perfect ancestors then all children of v have only perfect ancestors since all join vertices are perfect and a child u of v must have a smaller height then v, so u must either have a color lowering set or $\chi(G[U]) \leq (height(u)+1) \times |E(G[U]) \cap X|$. In the case that all children u of v have $\chi(G[U]) \leq (height(u)+1) \times |E(G[U]) \cap X|$ then $\chi(G[V]) < (height(v)+1) \times |E(G[V]) \cap X|$ by Lemma 15; if this is not the case then there is a child with a color lowering set and by Lemma 18 v has a color lowering set.
  - If v is a union vertex with only perfect ancestors then v is either perfect of imperfect.
    * If v is imperfect then by Lemma 12 $\chi(G[V]) \leq 2|E(G[V]) \cap X|$ and so $\chi(G[V]) \leq (height(v)+1) \times |E(G[V]) \cap X|$ since $height(v) \geq 1$.
    * If v is perfect all children of v have only perfect ancestors and a child j of v must have a smaller height then v, so j must either have a color lowering set or $\chi(G[J]) \leq (height(j)+1) \times |E(G[J]) \cap X|$. If v has a child j with $\chi(G[J]) \leq (height(j)+1) \times |E(G[J]) \cap X|$ then by Lemma 14 $\chi(G[V]) \leq (height(v)+1) \times |E(G[V]) \cap X|$. In the case that all children of v have a color lowering set, v has a color lowering set by Lemma 17.

All cases have now been covered proving this lemma. □

Rule 5. *If it follows from Lemma 19 after exhaustive application of Rules 3 and 4 that the root v of T has a color lowering set then reduce to $(G-S, E(G-S) \cap X, q-1)$, with S a color lowering set constructible in polynomial time.*

Lemma 20. *Rule 5 can be applied in polynomial time and is sound when applied to* Chromatic Number *on* Cograph+ke.

Proof. We can construct a color lowering set using the height t of the vertices in T to iterate through the vertices.

- For $t = 0$: All vertices v with $height(v) = 0$ are leafs. If v only has perfect ancestors then v has a color lowering set containing only v itself following Lemma 16. If v has an imperfect ancestor then we assume it has no color lowering set, even though it might have one.
- For $t > 0$: A vertex v with $height(v) > 0$ is either a join or a union vertex.
  - If v is a join vertex and its has a child with a color lowering set then set the color lowering set of v to be the color lowering set of that child which works following the proof in Lemma 18. If v has no children with a color lowering set then assume v has no color lowering set.

– If v is a union vertex and all of its children have a color lowering set then the union of the color lowering sets of all children is a color lowering set following the proof in Lemma 17 and thus we define the color lowering set of v to be the union of the color lowering sets of all its children. If v has a child without a color lowering set then assume v has no color lowering set.

The algorithm works since it follows from Lemma 19 that the root has a color lowering set and that every vertex of which Lemma 19 states it has a color lowering set either is a leaf with only perfect ancestors, a join vertex with at least one child with a color lowering set or a union vertex which has only children with a color lowering set. Also, all vertices in T are considered at most twice in the algorithm, once as itself and once as the child of its parent vertex, ensuring the algorithm runs in polynomial time. A color lowering set S obtained from the algorithm can now be used to reduce the problem to $(G-S, E(G-S) \cap X, q-1)$. The soundness of the reduction itself follows from Lemmas 16, 17 and 18 and Definition 3.

Following the proof, the color lowering set can be constructed in polynomial time ensuring the rule can be applied in polynomial time. □

For some $v \in V(T)$ with only perfect vertices in V, Rule 6 reduces $G[V]$ to a clique the size of $\chi(G[V])$. This works since we know the induced subgraph $G[V]$ is a cograph and thus has an easily computable chromatic number, and since all vertices in V are perfect we know they all have the same neighborhood outside of the vertices in V. The vertices in the clique get assigned that same neighborhood outside of the vertices in the clique. In any proper coloring for G we can assign all colors assigned to vertices in V to the vertices in the clique, so if we delete the vertices in V from the graph and add the clique to the graph we end up with a proper coloring for the resulting graph.

Rule 6. *If all vertices in V obtained from $v \in V(T)$ are perfect then reduce to $(((G-V) \cup K) + S, X, q)$, with K a clique with $|V(K)| = \chi(G[V])$ and S a set of edges which contains edges $\{z, y\} \in S$ for every $z \in V(K)$ and a $y \in G - V$ if and only if there is an edge $\{x, y\} \in E(G)$ with $x \in V$.*

Lemma 21. *Rule 6 can be applied in polynomial time and is sound when applied to* Chromatic Number *on* Cograph+*ke*.

Proof. If (G, X, q) is a Yes-instance, then $(((G-V) \cup K) + S, X, q)$ is a Yes-instance. Proof: If (G, X, q) is a Yes-instance, then by the problem definition there is a proper q-coloring of G. Consider such a coloring, and let M be the set of colors that are used for vertices in V. We know $|M| \geq V(K)$ since $|V(K)| = \chi(G[V])$. Now color the vertices in $V(K)$ as follows: assign every vertex in $V(K)$ an arbitrary unique color from M. Now add the colored graph K to the colored graph G, add all edges in S and remove all vertices in V. The result is a proper q-coloring for $((G-V) \cup K) + S$. To see this, consider an edge in $((G-V) \cup K) + S$.

- If the edge has both endpoints in K then both endpoints receive a different color since all vertices in $V(K)$ are assigned their own unique color from M.
- If the edge has both endpoints in $G-V$ then both endpoints receive a different color since we started with a proper q-coloring and nothing was recolored in $G-V$.

- If the edge has an endpoint $u \in K$ and an endpoint $v \in G - V$, then $\{u, v\} \in S$. Following the definition of the set S there must be a $y \in V$ for which $\{y, v\} \in E(G)$ and following Proposition 5 there is thus an edge $\{y, v\} \in E(G)$ for all $y \in V$ since all vertices in V are perfect. This means that in the proper q-coloring of G v is assigned a color not in M and thus v has a different color then u in the coloring for $((G - V) \cup K) + S$.

The endpoints of each edge receive different colors, proving that the q-coloring of $((G - V) \cup K) + S$ is proper. So $((G - V) \cup K) + S$ has a proper q-coloring, proving that $(((G - V) \cup K) + S, X, q)$ is a Yes-instance.

If $(((G - V) \cup K) + S, X, q)$ is a Yes-instance, then (G, X, q) is a Yes-instance. Proof: If $(((G - V) \cup K) + S, X, q)$ is a Yes-instance, then by the problem definition there is a proper q-coloring of $((G - V) \cup K) + S$. Consider such a coloring and let M be the set of colors used to color the vertices in K. We know $\chi(G[V]) = |V(K)| = |M|$ since $|V(K)| = \chi(G[V])$ and K is a clique. Now color the vertices in V as follows: $G[V]$ is a cograph since all vertices in V are perfect, and since a cograph is colorable in polynomial time consider such a coloring for $G[V]$ using only colors from M (which we can do since $\chi(G[V]) = |M|$). Now add the colored graph $G[V]$ to the colored graph $((G - V) \cup K) + S$, remove S and K from the graph and add all edges $\{u, v\} \in E(G)$ with $u \in V$ and $v \in V(G - V)$ to the graph. The result is a proper q-coloring for G. To see this, consider an edge in G.

- If the edge has both endpoints in V then both endpoints receive a different color since we used a proper coloring for $G[V]$.

- If the edge has both endpoints in $G - V$ then both endpoints receive a different color since we started with a proper q-coloring and nothing was recolored in $G - V$.

- If the edge has an endpoint $u \in V$ and an endpoint $v \in G - V$, then there is an edge $\{y, v\} \in S$ for each $y \in V(K)$. Since all colors in M are used to color the vertices in $V(K)$ vertex v must have a color which is not in M since we started with a proper q-coloring for $((G - V) \cup K) + S$, and thus both endpoints receive a different color.

The endpoints of each edge receive different colors, proving that the q-coloring of G is proper. So G has a proper q-coloring, proving that (G, X, q) is a Yes-instance.

The effect of this rule in the cotree is that if v is a join vertex, the parent of v is a union vertex u and the vertices in S will be leafs and all children of a new join vertex which is a child of u. If v is a union vertex with join parent vertex j then all vertices in S will be leafs and all children of j.

The information to apply this rule can be extracted from T in polynomial time ensuring the rule can be applied in polynomial time. □

Rule 7 cuts certain subgraphs with only perfect vertices in them from the graph if the subgraphs do not contribute to the chromatic number of the graph, i.e. deleting the subgraphs will not alter $\chi(G)$. The structure of the cotree makes it easy to find the subgraphs that can be cut from the graph. Similarly, Rule 8 checks in the cotree for a certain structure to cut. Rule 8 checks if there is a union vertex with a join child, which itself has a union child, and all three mentioned vertices have exactly one child which has an imperfect leaf in its subtree. The rule when applied on such a structure in T results in the removal of a union vertex from T and the merging of the child and parent join vertices of the removed union vertex in T.

Rule 7. *If a join vertex $j \in V(T)$ with only perfect vertices in J with a parent union vertex $u \in V(T)$ has $lb(l) \geq lb(j)$ for some other child join vertex $l \in V(T)$ of u, then reduce to $(G - J, E(G - J) \cap X, q)$.*

Lemma 22. *Rule 7 can be applied in polynomial time and is sound when applied to* Chromatic Number *on* Cograph+ke.

Proof. If (G, X, q) is a Yes-instance, then $(G - J, E(G - J) \cap X, q)$ is a Yes-instance by Proposition 1.

If $(G - J, E(G - J) \cap X, q)$ is a Yes-instance, then (G, X, q) is a Yes-instance. Proof: If $(G - J, E(G - J) \cap X, q)$ is a Yes-instance, then there is a proper q-coloring of $G - J$. Consider such a coloring, and let S be the set of colors that are used for vertices in L. Since $lb(l) \geq lb(j)$, $|S| \geq lb(j)$. Consider an arbitrary proper coloring of $G[J]$ using only colors in S, which we can do because $ub(j) = lb(j)$ since there are only perfect vertices in J and thus $|E(G[J]) \cap X| = 0$. Now create a coloring of G as follows: assign all vertices except vertices in J the same color as in the coloring of $G - J$. Assign the vertices in J the same color as in the arbitrary coloring of $G[J]$. The result is a proper q-coloring. To see this, consider an edge in G.

- If it has an endpoint in J its other endpoint is either also in J or in $V(G - U)$ since all vertices in J are perfect and are thus not connected to vertices in $U \setminus J$ since the common ancestor of lowest height is a union in this case.
  - If the edge has both endpoints in J the endpoints receive different colors, since we started with a proper coloring of $G[J]$.
  - If the edge has an endpoint in J and its other endpoint k is in $G - U$, then k is also connected to all vertices in L using Proposition 5 and the fact that all vertices in J are perfect. Since all colors in S are used in L, k will receive a color not in S and thus both endpoints receive different colors.
- If the edge has both endpoints in $G - J$ the endpoints receive different colors, since we started with a proper q-coloring of $G - J$.

Hence the endpoints of each edge receive different colors, proving that the q-coloring of G is proper. So G has a proper q-coloring, proving that (G, X, q) is a Yes-instance.

Getting the set J can be done in polynomial time using the cotree, ensuring the rule can be applied in polynomial time. □

Rule 8. *If after exhaustive application of Rules 6 and 7 on T there is a union vertex $a \in V(T)$ with a child $b \in V(T)$ with only perfect vertices in B and a join child $c \in V(T)$ with at least one imperfect vertex in C, and c has all of its children containing only perfect vertices except for a union child $d \in V(T)$, and d has a perfect child $e \in V(T)$ and a join child $f \in V(T)$ with at least one imperfect vertex in F, reduce to $(G - E, X, q)$.*

Lemma 23. *Rule 8 can be applied in polynomial time and is sound when applied to* Chromatic Number *on* Cograph+ke.

Proof. If (G, X, q) is a Yes-instance, then $(G - E, X, q)$ is a Yes-instance by Proposition 1.

If $(G - E, X, q)$ is a Yes-instance, then (G, X, q) is a Yes-instance. Proof: If $(G - E, X, q)$ is a Yes-instance, then there is a proper q-coloring of $G - E$. Consider such a coloring, and let M be the set of colors that are used for vertices in B. It must be the case that $lb(a) = lb(b) > lb(c)$, otherwise B would have been deleted by Rule 7. All children of c except for d only contain perfect vertices, which means they have all been turned into cliques following Rule 6 which means all these perfect vertices are leafs and children of c (see the last part of Lemma 21). Consider all perfect children of c as one big clique S and let N be the set of colors that are used for vertices in $V(S)$. Now we get $lb(c) = |V(S)| + lb(d)$ and since $lb(d) = lb(e)$ (otherwise E would have been deleted by Rule 7) also $lb(c) = |V(S)| + lb(e)$. So we get $lb(b) > |V(S)| + lb(e)$ and since all vertices in B, S and E are perfect, $\chi(G[B]) > \chi(S) + \chi(G[E])$ and thus $M > N + \chi(G[E])$. Now consider an arbitrary proper coloring of $G[E]$ using only colors in $M \setminus N$. Now create a coloring of G as follows: assign all vertices except vertices in E the same color as in the coloring of $G - E$. Assign the vertices in E the same color as in the arbitrary proper coloring of $G[E]$. The result is a proper q-coloring. To see this, consider an edge in G.

- If it has an endpoint in E its other endpoint is either also in E or in $V(G - D)$ since all vertices in E are perfect and because D is a union they are thus not connected to vertices to F.
  - If the edge has both endpoints in E the endpoints receive different colors, since we started with a proper coloring of $G[E]$.
  - If the edge has an endpoint in E and its other endpoint k is in $G[A] - D$, then $k \in V(S)$ since all vertices in E are perfect and vertex a is a union which means if a vertex in E would be connected to a vertex in B, the vertex in E is imperfect which is not the case. Since the endpoint in E receives a color from $M \setminus N$ and the endpoint in $V(S)$ receives a color from N, both endpoints receive a different color.
  - If the edge has an endpoint in E and its other endpoint k is in $G - A$, then k is also connected to all vertices in B using Proposition 5 and the fact that all vertices in E are perfect. Since all colors in M are used in B, k will receive a color not in M and thus both endpoints receive different colors.
- If the edge has both endpoints in $G - E$ the endpoints receive different colors, since we started with a proper q-coloring of $G - E$.

Hence the endpoints of each edge receive different colors, proving that the q-coloring of G is proper. So G has a proper q-coloring, proving that (G, X, q) is a Yes-instance.

The effect of this rule in the cotree is that union vertex d becomes obsolete since it has only one child left and d will thus be thrown away, merging the join vertices c and f into one vertex, and thus possibly reducing the height of T by two.

We can check for the structures we want to cut from the graph in polynomial time using T, ensuring the rule can be applied in polynomial time. □

Finally, Lemma 24 provides a kernel for Chromatic Number on Cograph+ke using all rules introduced in this section.

Lemma 24. Chromatic Number *on* Cograph+ke *has a kernel with at most* $112k^3 - 18k^2 - 2k$ *vertices and exhaustive application of Rules 3, 4, 5, 6, 7 and 8 gives us such a kernel.*

Proof. By Lemma 19 we know the root v of T either has an imperfect ancestor, a color lowering set or $\chi(G[V]) \leq (height(v) + 1) \times |E(G[V]) \cap X|$. Since v is the root it has no ancestors and thus no imperfect ancestor. If it follows from Lemma 19 that v has a color lowering set then Rule 5 is applicable to v implying that the rules have not been exhaustively applied, which is not the case. This means $\chi(G[V]) \leq (height(v)+1) \times |E(G[V]) \cap X|$ and thus $\chi(G) \leq (height(T)+1) \times |X|$. After exhaustive application of Rule 8 we know that there can not be a union vertex with a join vertex child which in turn has a union vertex child with all vertices only having one child containing imperfect vertices. This means at most a join vertex with a union child with a join child all with only one child containing imperfect vertices can exist and thus either the last join vertex has a leaf child which is imperfect or the join vertex has a union child which has multiple imperfect children. This means the total height of T can be no more than $8|X|$ because there can only be $2|X|-1$ join/union vertices with multiple children containing imperfect vertices since there are $|X|$ edges all with 2 endpoints and each endpoint marks an imperfect vertex. Since $\chi(G) \leq (height(T)+1) \times |X|$ this means $\chi(G) \leq (8|X|+1) \times |X| = 8|X|^2 + |X|$. There are at most $2|X|-1$ vertices with multiple children containing imperfect vertices and $2|X|$ imperfect leaf vertices and all of the $4|X|-1$ vertices have at most 3 ancestors direct in line until there is another ancestor with multiple children containing imperfect vertices because of Rule 8. This means there are at most $4 \times (2|X|-1) + 3 \times 2|X| = 14|X| - 4$ join/union vertices in T. After exhaustive application of Rule 7 there is at most one child join vertex with only perfect leafs under each union vertex, and following Rule 6 this is a clique K (a join vertex with only perfect leaf vertices), and since $\chi(G) \leq 8|X|^2 + |X|$ we know that $|V(K)| \leq 8|X|^2 + |X|$. Also the total amount of perfect children vertices under a join vertex, which are all leafs following Rule 6, must be at most $8|X|^2 + |X|$ since the perfect leaf vertices form a clique. This means the total amount of perfect leafs is at most $(8|X|^2 + |X|) \times (14|X| - 4) = 112|X|^3 - 18|X|^2 - 4|X|$. There are at most $2|X|$ imperfect leafs, so the total amount of leafs in T and thus the total amount of vertices in $V(G)$ is at most $112|X|^3 - 18|X|^2 - 2|X|$. Since $k = |X|$ this means $|V(G)| \leq 112k^3 - 18k^2 - 2k$, proving a bound on the kernel polynomial in k.

All rules applied in this kernel run in polynomial time. Also except for Rule 6 all rules reduce the graph by at least one vertex or edge when applied, and thus the rules can be applied at most $|V(G)| + |E(G)|$ times. Rule 6 should only be applied to subgraphs which are not cliques already, and in this case this rule can only be applied a polynomial number of times in $|V(G)|$. This means the kernel runs in polynomial time in the size of the graph. □

The kernel described in Lemma 24 proves that after application of the kernel on an instance of Chromatic Number on Cograph+ke the resulting instance contains at most $112k^3 - 18k^2 - 2k$ vertices, which might still be a lot depending on the size of the parameter. This kernel therefore serves solely as an upper bound, a proof that a polynomial kernel is possible indeed, and as such this is a good result.

5 Lower Bounds

In this section the lower bounds described in this thesis are proved. Section 5.1 provides proof that Chromatic Number parameterized by the edge addition distance to a split graph (Split$-ke$) does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$. In Section 5.2 we prove that Chromatic Number parameterized by the edge addition distance to a chordal graph (Chordal$-ke$) does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$.

5.1 Split-ke

In this section we prove a lower bound for kernels of Chromatic Number on Split$-ke$, namely that there is no polynomial kernel possible if $NP \not\subseteq coNP/poly$. We define Chromatic Number parameterized by the edge addition distance to a split graph as follows.

> Chromatic Number on Split$-ke$
> **Input:** An undirected graph G, a natural number q and a modulator $X \subseteq \binom{V(G)}{2} \setminus E(G)$ such that $G + X \in$ Split.
> **Parameter:** The size $k := |X|$ of the modulator.
> **Question:** Is $\chi(G) \leq q$?

We need Definition 4, which is a definition from Bodlaender [1, Definition 2], to help us prove there is no kernel for Chromatic Number parameterized by the edge addition distance to a split graph.

Definition 4. *(cf. [1, Definition 2]) Let P and Q be parameterized problems. We say that P is polynomial time and parameter reducible to Q, written $P \leq_{ptp} Q$, if there exists a polynomial time computable function $f : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ and a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that for all $(x, k) \in (\Sigma^*, \mathbb{N})$ the following hold:*

- *$(x, k) \in P$ if and only if $(x', k') = f(x, k) \in Q$.*
- *$k' \leq p(k)$.*

We call f a polynomial time and parameter transformation from P to Q.

Lemma 25 proves why Chromatic Number parameterized by the edge addition distance to a split graph does not admit a polynomial kernel using the fact that if $NP \not\subseteq coNP/poly$ then Chromatic Number parameterized by the vertex cover does not admit a polynomial kernel (Cai [4]) and that Chromatic Number parameterized by the vertex cover is polynomial time and parameter reducible to Chromatic Number parameterized by the edge addition distance to a split graph.

Lemma 25. Chromatic Number *parameterized by the edge addition distance to a split graph does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$.*

Proof. As proven by Leizhen Cai [4], Chromatic Number parameterized by the vertex cover does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$. By the definition of a vertex cover we know that for any vertex cover $S \subseteq V(G)$, $R = V(G) \setminus S$ is an independent set. If we would add a set of edges K to G such that $G[S]+K$ is a clique, then $G+K$ is a split graph in which $G[S]+K$ is a clique and R is an independent set. Also, $|K| \leq 0.5|S|^2$ since the number of edges in a clique is at most half of the square of the number of vertices in the clique. This means Chromatic Number parameterized by the vertex cover is polynomial time and parameter reducible to Chromatic Number parameterized by the edge addition distance to a split graph since the operation of getting the split graph from a vertex cover just described is the function f we need to prove polynomial time and parameter reducibility as described in Definition 4. As a function p (also from Definition 4) we can use the square function since $|K| < |S|^2$. Boldlaender states in [1, Theorem 8] that for parameterized problems P and Q, if $P \leq_{ptp} Q$ then, under a mild technical assumption which is satisfied in this case, a polynomial kernel for Q yields a polynomial kernel for P. This means that if Chromatic Number parameterized by the edge addition distance to a split graph would admit a polynomial kernel then Chromatic Number parameterized by the vertex cover would also admit a polynomial kernel, which is not the case and thus Chromatic Number parameterized by the edge addition distance to a split graph does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$. □

5.2 Chordal-ke

In this section we prove a lower bound for kernels of Chromatic Number on Chordal$-ke$, namely that there is no polynomial kernel possible if $NP \not\subseteq coNP/poly$. We define Chromatic Number parameterized by the edge addition distance to a chordal graph as follows.

> Chromatic Number on Chordal$-ke$
> **Input:** An undirected graph G, a natural number q and a modulator $X \subseteq \binom{V(G)}{2} \setminus E(G)$ such that $G + X \in$ Chordal.
> **Parameter:** The size $k := |X|$ of the modulator.
> **Question:** Is $\chi(G) \leq q$?

Lemma 26 gives a simple proof why Chromatic Number parameterized by the edge addition distance to a chordal graph does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$ using Lemma 25 which states that Chromatic Number parameterized by the edge addition distance to a *split* graph does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$.

Lemma 26. Chromatic Number *parameterized by the edge addition distance to a chordal graph does not admit a polynomial kernel if* $NP \not\subseteq coNP/poly$.

Proof. Since a *split* graph is also a chordal graph, if Chordal$-ke$ would admit a polynomial kernel we could use it to make a polynomial kernel for *split*$-ke$, which can not exist following Lemma 25 and thus Chordal$-ke$ does not admit a polynomial kernel if $NP \not\subseteq coNP/poly$. □

6 Conclusions

This thesis only considers parameterizations of Chromatic Number and achieves a number of nice upper and lower bounds for kernels of these parameterized problems. Note however that every upper bound found for a parameterization of Chromatic Number is also an upper bound for a parameterization of q-Coloring, which adds an extra dimension to the scope of results achieved in this thesis.

For further work it is worth studying if Chromatic Number on Cograph$-ke$ and Chromatic Number on Cograph$\pm ke$ admit a polynomial kernel, since this thesis only finds a kernel for Chromatic Number on Cograph$+ke$. It is also interesting to investigate lower bounds and better upper bounds for polynomial kernels of parameterized problems, such as the polynomial kernels found in this thesis. The goal then is to bring the lower and upper bounds together for a kernel of a parameterized problem to know exactly how efficient kernelization is for that parameterized problem. For instance Dell and van Melkebeek [5] prove that kernels for Feedback Vertex Set can never reduce instances to a size smaller than $O(k^{2-\epsilon})$, with k the parameter, while Thomassé [13] provides a kernel for Feedback Vertex Set of size $O(k^2)$.

Acknowledgments

I am grateful to Bart M. P. Jansen for constantly steering me in the right direction and Hans L. Bodlaender for numerous valuable tips.

References

[1] Hans L. Bodlaender. Kernelization: New upper and lower bound techniques. In Jianer Chen and Fedor V. Fomin, editors, *IWPEC*, volume 5917 of *Lecture Notes in Computer Science*, pages 17–37. Springer, 2009.

[2] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *CoRR*, abs/1206.5941, 2012.

[3] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes: a Survey.* Society for Industrial and Applied Mathematics, 1999.

[4] Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

[5] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In Leonard J. Schulman, editor, *STOC*, pages 251–260. ACM, 2010.

[6] R. Downey and Michael R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer, 1999.

[7] Andrew Drucker. New limits to classical and quantum instance compression. In *FOCS*, pages 609–618. IEEE Computer Society, 2012.

[8] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct pcps for np. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[9] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[10] Bart M. P. Jansen and Stefan Kratsch. Data reduction for graph coloring problems. In Olaf Owe, Martin Steffen, and Jan Arne Telle, editors, *FCT*, volume 6914 of *Lecture Notes in Computer Science*, pages 90–101. Springer, 2011.

[11] Tommy R. Jensen and B. Toft. *Introduction to Graph Coloring.* John Wiley & Sons, Inc., 1994.

[12] Dániel Marx. Parameterized coloring problems on chordal graphs. *Theor. Comput. Sci.*, 351(3):407–424, 2006.

[13] Stéphan Thomassé. A quadratic kernel for feedback vertex set. In Claire Mathieu, editor, *SODA*, pages 115–119. SIAM, 2009.