

# Parameterized coloring problems on chordal graphs

Dániel Marx\*

26th March 2005

## Abstract

In the precoloring extension problem (PREXT) a graph is given with some of the vertices having preassigned colors and it has to be decided whether this coloring can be extended to a proper coloring of the graph with the given number of colors. Two parameterized versions of the problem are studied in the paper: either the number of precolored vertices or the number of colors used in the precoloring is restricted to be at most  $k$ . We show that for chordal graphs these problems are polynomial-time solvable for every fixed  $k$ , but W[1]-hard if  $k$  is the parameter. For a graph class  $\mathcal{F}$ , let  $\mathcal{F} + ke$  (resp.,  $\mathcal{F} + kv$ ) denote those graphs that can be made to be a member of  $\mathcal{F}$  by deleting at most  $k$  edges (resp., vertices). We investigate the connection between PREXT in  $\mathcal{F}$  (with the two parameters defined above) and the coloring of  $\mathcal{F} + ke$ ,  $\mathcal{F} + kv$  graphs (with  $k$  being the parameter). Answering an open question of Leizhen Cai [5], we show that coloring chordal+ $ke$  graphs is fixed-parameter tractable.

## 1 Introduction

In graph vertex coloring we have to assign colors to the vertices such that neighboring vertices receive different colors. In the *precoloring extension* (PREXT) problem a subset  $W$  of the vertices have preassigned colors and we have to extend this precoloring to a proper  $k$ -coloring of the whole graph. Since vertex coloring is the special case when  $W = \emptyset$ , the precoloring extension problem is NP-complete in every class of graphs where ordinary vertex coloring is NP-complete. However, there are classes of graphs where coloring is polynomial-time solvable, but the more general precoloring extension problem is NP-complete, see [2, 11, 12] for more background and results on PREXT.

In this paper we study the precoloring extension problem restricted to interval and chordal graphs. PREXT is NP-complete for interval graphs [2] (even for unit interval graphs [17]), hence it is NP-complete for chordal graphs as well. On the other hand, if every color is used only once in the precoloring (this special case is called 1-PREXT), then the problem becomes polynomial-time solvable for interval graphs [2], and more generally, for chordal graphs [16]. Here we introduce two new restricted versions of PREXT: we investigate the complexity of the problem when either there are only  $k$  precolored vertices, or there are only  $k$  colors used in the precoloring. Clearly, the former is a special case of the latter. By giving an algorithm that runs in  $O(kn^{k+2})$  time on an  $n$  vertex graph, we show that for fixed  $k$  both problems are polynomial-time solvable on chordal graphs.

---

\*Dept. of Computer Science and Information Theory, Budapest University of Technology and Economics (dmarx@cs.bme.hu). Research is supported in part by grants OTKA 44733, 42559, and 42706 of the Hungarian National Science Fund.

An algorithm with running time  $O(kn^{k+2})$  is not practical even for, say,  $k = 10$ . Therefore, we study the precoloring extension problem also in the framework of parameterized complexity. Our aim is to investigate whether there is an algorithm where  $k$  does not appear in the exponent of  $n$ . The central notion of parameterized complexity is uniformly polynomial time: we say that an algorithm solves a problem in uniformly polynomial time if the running time is  $f(k)p(n)$  for some arbitrary function  $f$  and polynomial  $p$ . If a problem has such an algorithm, then the problem is said to be fixed-parameter tractable (FPT). Parameterized complexity gives us a wide range of tools to design uniformly polynomial algorithms. On the negative side, the theory of W[1]-hardness gives us a method to show that a problem is not fixed-parameter tractable (under some plausible complexity-theoretic assumptions).

The parameterized complexity analysis shows that we cannot expect to improve the  $O(kn^{k+2})$  time algorithm for PREXT to a uniformly polynomial algorithm, since the problem is W[1]-hard even for interval graphs. To establish W[1]-hardness, we use the recent result of Slivkins [23] that the edge-disjoint paths problem is W[1]-hard.

Leizhen Cai [5] introduced a whole new family of parameterized problems. If  $\mathcal{F}$  is an arbitrary class of graphs, then denote by  $\mathcal{F} - kv$  (resp.,  $\mathcal{F} - ke$ ) the class of those graphs that can be obtained from a member of  $\mathcal{F}$  by deleting at most  $k$  vertices (resp.,  $k$  edges). Similarly, let  $\mathcal{F} + kv$  (resp.,  $\mathcal{F} + ke$ ) be the class of those graphs that can be made to be a member of  $\mathcal{F}$  by deleting at most  $k$  vertices (resp.,  $k$  edges). For any class of graphs  $\mathcal{F}$  and for any graph problem, we can ask what is the complexity of the problem restricted to these “almost  $\mathcal{F}$ ” graphs. This question is investigated in [5] for the vertex coloring problem.

Although there is a large amount of work in the literature on the complexity of coloring for various classes of graphs, there are relatively few results concerning these modified classes. It seems that graph coloring is a particularly interesting problem that is worth studying on these classes. In the case of problems such as MAXIMUM CLIQUE, MAXIMUM INDEPENDENT SET, and MINIMUM VERTEX COVER, a polynomial-time algorithm for  $\mathcal{F}$  graphs immediately gives a uniformly polynomial time algorithm for  $\mathcal{F} + kv$  graphs. There are  $2^k$  possibilities for including the  $k$  extra vertices in the solution, and if we fix one possibility, then we have to solve the problem for an  $\mathcal{F}$  graph. On the other hand, coloring  $\mathcal{F} + kv$  or  $\mathcal{F} + ke$  graphs can be very different from coloring graphs in  $\mathcal{F}$ , and might involve significantly new approaches. For example, bipartite graphs are easy to color, but coloring bipartite+ $2v$  graphs is NP-complete [5]. Edge coloring bipartite graph is also easy: a classical result of König [15] states that number of colors required to edge color a bipartite graph equals the maximum degree. However, edge coloring bipartite+ $1v$  graphs [20] requires new techniques and insight into the problem.

We investigate the relations between PREXT and the coloring of the modified graph classes. We show that for several reasonable graph classes, reductions are possible between PREXT for graphs in  $\mathcal{F}$  and the coloring of  $\mathcal{F} + kv$  or  $\mathcal{F} + ke$  graphs. Based on this correspondence between the problems, we show that both chordal+ $ke$  and chordal+ $kv$  graphs can be colored in polynomial time for fixed  $k$ , but chordal+ $kv$  graph coloring is W[1]-hard. Moreover, answering an open question of Cai [5], we develop a uniformly polynomial algorithm for coloring chordal+ $ke$  graphs. A key idea in the analysis of the algorithm is to bound the running time using the celebrated inequality of Bollobás. Table ?? summarizes the results of the paper.

The paper is organized as follows. Section 2 briefly reviews the most important notions of parameterized complexity. Section 3 contains preliminary notions. Section 4 reviews tree decomposition, which will be our main tool when dealing with chordal graphs. In Section 5, we investigate the parameterized PREXT problems for chordal graphs. The connections between PREXT and coloring  $\mathcal{F} + ke$ ,  $\mathcal{F} + kv$  graphs are investigated in Section 6. Finally, in Section 7, we show that coloring chordal+ $ke$  graphs is fixed-parameter tractable.

Table 1: Results of the paper on interval and chordal graphs

Problem	Interval graphs	Chordal graphs
PREXT with $k$ precolored vertices	W[1]-hard, in P for fixed $k$	W[1]-hard, in P for fixed $k$
PREXT with $k$ colors in the precoloring	W[1]-hard, in P for fixed $k$	W[1]-hard in P for fixed $k$
Coloring $\mathcal{F} + ke$ graphs	FPT	FPT
Coloring $\mathcal{F} + kv$ graphs	W[1]-hard, in P for fixed $k$	W[1]-hard, in P for fixed $k$

## 2 Parameterized Complexity

In parameterized complexity we are dealing with *parameterized problems*, where every input instance  $(x, k)$  has a distinguished part  $k$  called the *parameter*. For example, in the MAXIMUM CLIQUE problem the parameter  $k$  is the size of the clique to be found. In problems such as MAXIMUM CLIQUE, MINIMUM VERTEX COVER, and LONGEST PATH the problem can be solved trivially by trying all the  $O(n^k)$  possibilities for the solution. However, such an algorithm is not really practical:  $n^k$  can be huge even for moderate values of  $n$  and small values of  $k$ . Therefore, we are interested in the question whether there is an algorithm where  $k$  does not appear in the exponent of  $n$ . We say that a parameterized problem is *fixed-parameter tractable (FPT)* if it has an algorithm with running time  $f(k)n^c$ , where  $c$  is a constant independent of  $k$  and  $n$ , and  $f$  depends only on  $k$ . Such an algorithm can be useful even for large values of  $n$ , provided that  $f(k)$  is relatively small and  $c$  is a small constant. It turns out that several NP-hard problems, e.g., MINIMUM VERTEX COVER, LONGEST PATH,  $k$ -DISJOINT TRIANGLES, etc. are fixed-parameter tractable. There is a standard toolbox of techniques for designing FPT algorithms: kernelization, bounded search trees, color coding, well-quasi ordering, just to name some of the more important ones (see [7] and [19]).

The theory of NP-completeness can be used to show that certain problems are unlikely to be polynomial-time solvable. In parameterized complexity, W[1]-hardness plays an analogous role: by showing that a problem is W[1]-hard, we can give strong evidence that the problem is not fixed-parameter tractable. We omit the somewhat technical definition of the complexity class W[1], see [7] for details. Here it will be sufficient to know that there are several problems, including MAXIMUM CLIQUE, that were proved to be W[1]-hard. Furthermore, we also expect that there is no  $O(n^{o(k)})$  algorithm for MAXIMUM CLIQUE: recently it was shown that there exists an  $O(n^{o(k)})$  algorithm for MAXIMUM CLIQUE if and only if there are subexponential-time algorithms for 3-SAT (see [6] and [9]).

To prove that a parameterized problem  $Q'$  is W[1]-hard, we have to present a parameterized reduction from a known W[1]-hard problem  $Q$  to  $Q'$ . A *parameterized reduction* from problem  $Q$  to problem  $Q'$  is a function that transforms a problem instance  $(x, k)$  of  $Q$  into a problem instance  $(x', k')$  of  $Q'$  such that

- $(x', k') \in Q'$  if and only if  $(x, k) \in Q$ ,
- $k'$  is a function of  $k$  independent of  $x$ , and

- the transformation can be computed in time  $f(k) \cdot |x|^c$  for some constant  $c$  and function  $f(k)$ .

It is easy to see that if there is a parameterized reduction from  $Q$  to  $Q'$ , and  $Q'$  is fixed-parameter tractable, then it follows that  $Q$  is fixed-parameter tractable as well.

We remark that there can be many different parameterizations of the same problem. For example, in the MAXIMUM CLIQUE problem, the required solution size seems to be the most natural choice for the parameter. However, there are several other possibilities: the parameter can be the maximum degree of the graph, the treewidth of the graph, or some other graph parameter. In [5] and [10], some new types of parameters are investigated, for example the parameter can be the distance of the input instance from some (defined) easy class.

### 3 Preliminaries

Given a color set  $C$ , a  $C$ -coloring of graph  $G(V, E)$  is an assignment  $\psi: V \rightarrow C$  such that  $\psi(u) \neq \psi(v)$  whenever  $u \in V$  and  $v \in V$  are connected by an edge. We introduce two different parameterizations of the precoloring extension problem. Formally, the problem is defined as follows:

PRECOLORING EXTENSION (PREXT)	
<i>Input:</i>	A graph $G(V, E)$ , a set of colors $C$ , and a precoloring $\psi: W \rightarrow C$ for a set of vertices $W \subseteq V$ .
<i>Parameter 1:</i>	$ W $ , the number of precolored vertices.
<i>Parameter 2:</i>	$ \{\psi(w) : w \in W\}  =  C_W $ , the number of colors used in the precoloring.
<i>Question:</i>	Is there a proper $C$ -coloring $\psi'$ of $G$ that extends $\psi$ (i.e., $\psi'(w) = \psi(w)$ for every $w \in W$ )?

Note that  $C_W \subseteq C$  is the set of colors appearing on the precolored vertices, and can be much smaller than the set of available colors  $C$ . When we consider parameter 1, then the problem will be called PREXT with fixed number of precolored vertices, while considering parameter 2 corresponds to PREXT with fixed number of colors in the precoloring. The first problem is obviously easier than the latter.

For every class  $\mathcal{F}$  and every fixed  $k$ , one can ask what is the complexity of vertex coloring on the four classes  $\mathcal{F} + ke$ ,  $\mathcal{F} + kv$ ,  $\mathcal{F} - ke$ ,  $\mathcal{F} - kv$ . The first question is whether the problem is NP-complete for some fixed  $k$  (for example, coloring bipartite+ $2v$  graphs is NP-complete [5]). If the problem is solvable in polynomial time for every fixed  $k$ , then the next question is whether the problem is fixed-parameter tractable, that is, whether there is a uniformly polynomial-time algorithm for the given classes.

If  $\mathcal{F}$  is hereditary with respect to taking induced subgraphs, then  $\mathcal{F} - kv$  is the same as  $\mathcal{F}$ , hence coloring  $\mathcal{F} - kv$  graphs is the same as coloring in  $\mathcal{F}$ . Moreover, it is shown in [5] that if  $\mathcal{F}$  is closed under edge contraction and has a polynomial time algorithm for coloring, then coloring  $\mathcal{F} - ke$  graphs is fixed parameter tractable. Therefore, we can conclude that coloring chordal- $kv$  and coloring chordal- $ke$  graphs are in FPT. In this paper we show that coloring chordal+ $ke$  graphs is in FPT, but coloring chordal+ $kv$  graphs is W[1]-hard.

The *modulator* of an  $\mathcal{F} + ke$  graph  $G$  is a set of at most  $k$  edges whose removal makes  $G$  a member of  $\mathcal{F}$ . Similar definitions apply for the other classes. We will call the edges

and vertices of the modulator *special edges and vertices*. In the case of  $\mathcal{F} + ke$  and  $\mathcal{F} - ke$  graphs, the vertices incident to the special edges will be called the special vertices.

When considering the complexity of coloring in a given parameterized class, then we can assume either that only the graph is given in the input, or that a modulator is also given. In the case of coloring chordal $-ke$  graphs, this makes no difference as finding the modulator of such a graph (i.e., the at most  $k$  edges that can make the graph chordal) is in FPT [4, 13]. On the other hand, the parameterized complexity of finding the modulator of a chordal $+ke$  graph is open. Thus in our algorithm for coloring chordal $+ke$  graphs, we assume that the modulator is given in the input.

## 4 Tree decomposition

A graph is *chordal* if it does not contain a cycle of length greater than 3 as an induced subgraph. Equivalently, a graph is chordal if and only if every cycle of size greater than 3 contains a chord, that is, an edge between two vertices not neighbors in the cycle. This section summarizes some well-known properties of chordal graphs. First, chordal graphs can be also characterized as the intersection graphs of subtrees in a tree (see e.g., [8]):

**Theorem 4.1.** *The following two statements are equivalent:*

1.  $G(V, E)$  is chordal.
2. There exists a tree  $T(U, F)$  and a subtree  $T_v \subseteq T$  for each  $v \in V$  such that  $u, v \in V$  are neighbors in  $G(V, E)$  if and only if  $T_u \cap T_v \neq \emptyset$ .

The tree  $T$  together with the subtrees  $T_v$  is called the *tree decomposition* of  $G$ .<sup>1</sup> A tree decomposition of a chordal graph  $G$  can be found in polynomial time (see [8, 22]).

We assume that  $T$  is a rooted tree with some root  $r \in U$ . For clarity, we will use the word “vertex” when we refer to the chordal graph  $G(V, E)$ , and “node” when referring to the tree  $T(U, F)$ . For a node  $x \in U$ , denote by  $V_x$  those vertices whose subtree contains  $x$  or a descendant of  $x$ . The subgraph of  $G$  induced by  $V_x$  will be denoted by  $G_x = G[V_x]$ . For a node  $x \in U$  of  $T$ , denote by  $K_x$  the union of  $v$ ’s where  $x \in V(T_v)$ . Clearly, the vertices of  $K_x$  are in  $V_x$ , and they form a clique in  $G_x$ , since the corresponding trees intersect in  $T$  at node  $x$ . An important property of the tree decomposition is the following: for every node  $x \in U$ , the clique  $K_x$  separates  $V_x \setminus K_x$  and  $V \setminus V_x$ . That is, among the vertices of  $V_x$ , only the vertices in  $K_x$  can be adjacent to  $V \setminus V_x$ .

A tree decomposition will be called *nice* [14], if it satisfies the following additional requirements (see Figure 1):

- Every node  $x \in U$  has at most two children.
- If  $x \in U$  has two children  $y, z \in U$ , then  $K_x = K_y = K_z$  ( $x$  is a *join node*).
- If  $x \in U$  has only one child  $y \in U$ , then either  $K_x = K_y \cup \{v\}$  ( $x$  is an *introduce node*) or  $K_x = K_y \setminus \{v\}$  ( $x$  is a *forget node*) for some  $v \in V$ .
- If  $x \in U$  has no children, then  $K_x$  contains exactly one vertex ( $x$  is a *leaf node*).

---

<sup>1</sup>A note on terminology: what we call here “tree decomposition” is sometimes called “clique tree.” Moreover, here we are defining a special type of tree decomposition: usually, when dealing with non-chordal graphs, it is not required that  $u$  and  $v$  are neighbors whenever  $T_u \cap T_v \neq \emptyset$ .

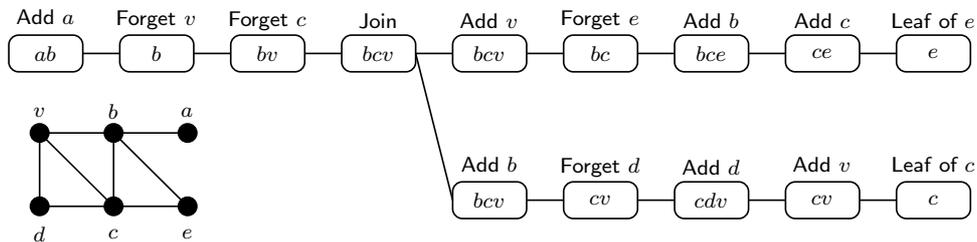


Figure 1: Nice tree decomposition of a chordal graph.

By splitting the nodes of the tree in an appropriate way, a tree decomposition of  $G$  can be transformed into a nice tree decomposition in polynomial time.

A vertex  $v$  can have multiple introduce nodes, but at most one forget node (the vertices in clique  $K_r$  of the root  $r$  have no forget nodes, but every other vertex has exactly one). For a vertex  $v$ , its subtree  $T_v$  is the subtree rooted at the forget node of  $v$  (if it exists, otherwise at the root) and whose leaves are exactly the introduce nodes and leaf nodes of  $v$ .

## 5 PREXT on chordal graphs

In this section we show that PREXT can be solved in polynomial time for chordal graphs if the number of colors used in the precoloring is bounded by a constant  $k$ . In general, PREXT is NP-hard for interval graphs, even if every color is used at most twice in the precoloring [2].

The algorithm presented below is a straightforward application of the tree decomposition described in Section 4. The running time of the algorithm is  $O(kn^{k+2})$ , hence it is not uniformly polynomial. In Theorem 5.2 we show that the problem is W[1]-hard, hence we cannot hope to find a uniformly polynomial algorithm.

**Theorem 5.1.** *The PREXT problem can be solved in  $O(kn^{k+2})$  time for chordal graphs if the number of colors in the precoloring is at most  $k$ .*

*Proof.* It can be assumed that the colors used in the precoloring are the colors  $1, 2, \dots, k$ . For each node  $x$  of the nice tree decomposition of the graph, we solve  $O(n^k)$  subproblems using dynamic programming. A subproblem is described by a vector  $[\alpha_1, \dots, \alpha_k]$ , where each  $\alpha_i$  is either a vertex of  $K_x$ , or the symbol  $\star$ . We say that such a vector is *feasible* for node  $x$ , if there is a precoloring extension for  $G_x$  with the following properties:

- If  $\alpha_i$  ( $1 \leq i \leq k$ ) is  $\star$ , then color  $i$  does not appear on the clique  $K_x$ .
- If  $\alpha_i$  is some vertex in  $K_x$ , then color  $i$  appears on this vertex.

Notice that in a feasible vector a vertex can appear at most once (but the star can appear several times), thus in the following we consider only such vectors.

Clearly, the precoloring can be extended to the whole graph if and only if the root node  $r$  has at least one feasible vector. The algorithm finds the feasible vectors for each node of  $T$ . We construct the feasible vectors for the nodes in a bottom-up fashion. First, they are easy to determine for the leaves. Moreover, we show that they can be constructed for an arbitrary node if the feasible vectors for its children are already available. For each such node  $x$ , we create a table that contains one bit for every possible vector saying whether this vector is

feasible for  $x$ . The table is organized in such a way that the bit corresponding to a given vector can be found in  $O(k)$  time.

In the rest of the proof, we consider the different type of nodes separately. At each node, we spend  $O(kn^k)$  time to fill the tables. Since the nice tree decomposition has  $O(n^2)$  nodes, it follows that the total running time is  $O(kn^{k+2})$ .

*Leaf nodes.* If leaf node  $x$  contains a vertex  $v$  precolored to color  $i$ , then  $x$  has only one feasible vector:  $\alpha_i = v$ , and  $\alpha_j = \star$  for  $i \neq j$ . If  $v$  is not precolored, then  $x$  has  $k$  or  $k + 1$  feasible vectors. For every  $1 \leq i \leq k$ , the vector with  $\alpha_i = v$  and  $\alpha_j = \star$  for  $j \neq i$  is feasible. Moreover, if  $|C| > k$ , then the vector containing only stars is also feasible (as  $v$  can receive a color not in  $C_W$ ).

*Introduce node of  $v$ .* Let  $y$  be the child of  $x$ . The feasible vectors for  $x$  can be determined as follows. Assume first that  $v$  is not a precolored vertex. We consider two cases. A vector containing  $v$  is feasible for  $x$  if and only if it becomes feasible for  $y$  after replacing  $v$  with  $\star$ . On the other hand, if the vector does not contain  $v$ , then it is feasible for  $x$  if and only if it is feasible for  $y$  and the following additional constraint holds: the number of non-star components contained in the vector has to be at least  $|K_x| - (|C| - |C_W|)$ . The reason why this has to hold is that we have to extend a coloring of  $G_y$  to  $v$  using a color not in  $C_W$ , and this is only possible if not all such colors are used on  $K_y$ . Therefore, there has to be at least  $|K_y| - (|C| - |C_W| - 1) = |K_x| - (|C| - |C_W|)$  vertices in  $K_y$  that receive colors from  $C_W$ , and each such vertex corresponds to a non-star component of the vector.

Considering every possible vector, we can create a table that determines for each vector whether it is feasible for  $y$ . Assuming that we have a look up table that allows us to check in  $O(k)$  time whether a vector is feasible for  $y$ , the table for  $x$  can be calculated in  $O(k)$  time per entry, which is  $O(kn^k)$  time in total.

If  $v$  is precolored to color  $i$ , then  $\alpha_i = v$  in every feasible vector for  $x$ . Therefore, the feasible vectors for  $x$  can be determined as above, but we have to throw away those vectors where the  $i$ -th component is not  $v$ .

*Forget node of  $v$ .* Let  $y$  be the child of  $x$ . A vector is feasible for  $x$  if and only if it is feasible for  $y$ , or it can be made feasible for  $y$  by replacing some star with  $v$ . The table can be constructed in  $O(kn^k)$  time, as in the previous case.

*Join node.* Let  $y$  and  $z$  be the children of  $x$ . We claim that a vector is feasible for  $x$  if and only if it is feasible for both  $y$  and  $z$ . The only if part is obvious. Now assume that a vector is feasible for  $y$  and  $z$ , let  $\psi_y$  and  $\psi_z$  be the corresponding precoloring extensions of  $G_y$  and  $G_z$ , respectively. Colorings  $\psi_y$  and  $\psi_z$  might be different on  $K_x$ , but they use the colors of  $C_W$  the same way: if a vertex of  $K_x$  receives a color from  $C_W$  in  $\psi_y$ , then it receives the same color in  $\psi_z$ , and vice versa. Therefore, by permuting in  $\psi_y$  the colors of  $C \setminus C_W$ , we can make  $\psi_y$  agree with  $\psi_z$  on  $K_x$  (notice that  $K_x$  is a clique, thus every color is used at most once on  $K_x$ ). There is no precolored vertex with color from  $C \setminus C_W$ , hence  $\psi_y$  remains a valid precoloring extension of  $G_y$  after the permutation. Now  $\psi_y$  and  $\psi_z$  can be merged to obtain a precoloring extension of  $G_x$ , and it shows that the vector is indeed feasible for  $x$ .  $\blacksquare$

To prove that PREXT with fixed number of precolored vertices is W[1]-hard for interval graphs, we present a parameterized reduction from the EDGE DISJOINT PATHS problem, which is the following:

EDGE DISJOINT PATHS

*Input:* A directed graph  $G(V, E)$ , with  $k$  pairs of vertices  $(s_i, t_i)$ .

*Parameter:* The number of pairs  $k$ .

*Question:* Is there a set of  $k$  pairwise edge disjoint directed paths  $P_1, \dots, P_k$  such that path  $P_i$  goes from  $s_i$  to  $t_i$ ?

Recently, Slivkins [23] proved that the EDGE DISJOINT PATHS problem is W[1]-hard for directed acyclic graphs.

**Theorem 5.2.** *PREXT with fixed number of precolored vertices is W[1]-hard for interval graphs.*

*Proof.* The proof is by a parameterized reduction from EDGE DISJOINT PATHS restricted to directed acyclic graphs. Given a directed acyclic graph  $G(V, E)$  and terminal pairs  $s_i, t_i$  ( $1 \leq i \leq k$ ), we construct an interval graph with  $k' = 2k$  precolored vertices in such a way that the interval graph has a precoloring extension if and only if the disjoint paths problem can be solved. Let  $1, 2, \dots, n$  be the vertices of  $G$  in a topological ordering. For each edge  $\overrightarrow{xy}$  of  $G$  we add an interval  $[x, y)$ . For each terminal pair  $s_i, t_i$  we add two intervals  $[0, s_i)$  and  $[t_i, n + 1)$ , and precolor these intervals with color  $i$ .

Denote by  $\ell(x)$  the number of intervals whose *right* end point is  $x$  (i.e., the intervals that arrive to  $x$  from the left), and by  $r(x)$  the number of intervals whose left end point is  $x$ . By construction,  $\ell(x)$  is the number of edges entering  $x$  plus the number of demands starting in  $x$ . If  $\ell(x) < r(x)$ , then add  $r(x) - \ell(x)$  new intervals  $[0, x)$  to the graph; if  $\ell(x) > r(x)$ , then add  $\ell(x) - r(x)$  new intervals  $[x, n + 1)$ . These new intervals ensure that each point of  $[0, n + 1)$  is contained in the same number (denote it by  $c$ ) of intervals: for each point the number of intervals ending there equals the number of intervals starting there. We claim that the constructed interval graph has a precoloring extension with  $c$  colors if and only if the disjoint paths problem has a solution.

Assume first that there are  $k$  disjoint paths joining the terminal pairs. For each edge  $\overrightarrow{xy}$ , if it is used by the  $i$ -th terminal pair, then color the interval  $[x, y)$  with color  $i$ . Notice that the intervals we colored with color  $i$  do not intersect each other, and their union is exactly  $[s_i, t_i)$ . Therefore, considering also the two intervals  $[0, s_i)$  and  $[s_i, n + 1)$  precolored with color  $i$ , each point of  $[0, n + 1)$  is covered by exactly one interval with color  $i$ . This means that each point is contained in exactly  $c - k$  intervals that do not have a color yet. Hence the uncolored intervals induce an interval graph where every point is in exactly  $c - k$  intervals, and it is well-known that such an interval graph has clique number  $c - k$  and can be colored with  $c - k$  colors. Therefore, the precoloring can be extended using  $c - k$  colors in addition to the  $k$  colors used in the precoloring.

Now assume that the precoloring can be extended using  $c$  colors. Each point in the interval  $[0, n + 1)$  is covered by exactly  $c$  intervals, thus each point is covered by an interval of color  $i$ . Hence if an interval with color  $i$  ends at point  $x$ , then an interval with color  $i$  has to start at  $x$ . The interval  $[0, s_i)$  has color  $i$ , thus there has to be an interval  $[s_i, s_{i,1})$  with color  $i$ . Similarly, there has to be an interval  $[s_{i,1}, s_{i,2})$  with color  $i$ , etc. Continuing this way, we will eventually arrive to an interval  $[s_{i,p}, t_i)$ . By the way the intervals were constructed, the edges  $\overrightarrow{s_i s_{i,1}}, \overrightarrow{s_{i,1} s_{i,2}}, \dots, \overrightarrow{s_{i,p} t_i}$  form a directed path  $P_i$  from  $s_i$  to  $t_i$ . It is clear that the paths for different values of  $i$  are disjoint, since each interval has only one color. Thus we constructed a solution to the disjoint paths problem, as required. ■

## 6 Reductions

In this section we give reductions between PREXT on  $\mathcal{F}$  and ordinary vertex coloring of  $\mathcal{F} + kv$ ,  $\mathcal{F} + ke$  graphs. It turns out that if  $\mathcal{F}$  is closed under disjoint union and attaching pendant vertices, then

$$\begin{aligned} \text{coloring } \mathcal{F} + ke \text{ graphs} &\preceq \text{PREXT on } \mathcal{F} \text{ with fixed } |W| \preceq \\ \text{coloring } \mathcal{F} + kv \text{ graphs} &\preceq \text{PREXT on } \mathcal{F} \text{ with fixed } |C_W| \end{aligned}$$

When coloring  $\mathcal{F} + ke$  or  $\mathcal{F} + kv$  graphs, we assume that the modulator of the graph is given in the input.

The reductions presented in this section do not follow exactly the definition of Section 2, as they are not many-to-one reductions. This means that here the reduction from  $Q$  to  $Q'$  is not a function that maps each instance of  $Q$  to an instance of  $Q'$ , but an algorithm that solves an instance of  $Q$  by making repeated calls to an oracle for  $Q'$ . Using this more general notion of reduction (“Turing-reduction”) does not affect the consequence that if  $Q$  is reducible to  $Q'$  then  $Q$  is easier than  $Q'$  in the sense that  $Q' \in \text{FPT}$  implies  $Q \in \text{FPT}$ .

When reducing the coloring of  $\mathcal{F} + ke$  or  $\mathcal{F} + kv$  graphs to PREXT, the idea is to consider each possible coloring of the special vertices and solve each possibility as a PREXT problem. In the other direction, we use the  $k$  additional edges or vertices to build gadgets that force the precolored vertices to the required colors.

First we show that  $\mathcal{F} + ke$  and  $\mathcal{F} + kv$  coloring can be reduced to PREXT:

**Theorem 6.1.** *For every class  $\mathcal{F}$  of graphs, coloring  $\mathcal{F} + ke$  graphs can be reduced to PREXT with fixed number of precolored vertices, if the modulator of the graph is given in the input.*

*Proof.* For a graph  $G \in \mathcal{F} + ke$ , the  $k$  special edges span at most  $k' := 2k$  special vertices, denote this set by  $W$ . We have to determine whether  $G$  has a  $C$ -coloring. It can be assumed that in the  $C$ -coloring the vertices in  $W$  receive colors only from  $1, 2, \dots, 2k$ . Therefore, the set  $W$  has at most  $(2k)^{2k}$  different colorings, for each such coloring we check whether it can be extended to the whole graph  $G$ . Clearly,  $G$  is  $C$ -colorable if and only if at least one such coloring can be extended. If the colors of the vertices in  $W$  are set, then the special edges can be removed, since for each such edge the end vertices already have a color. Deleting the special edges of  $G$  results in a graph in  $\mathcal{F}$ , hence we can use the assumed algorithm for PREXT: we have to check whether the precoloring on the at most  $k'$  vertices of  $W$  can be extended to the whole graph. ■

To reduce the coloring of  $\mathcal{F} + kv$  graphs to precoloring extension, we need that the class  $\mathcal{F}$  is closed under attaching pendant vertices. That is, if  $G \in \mathcal{F}$ , and  $v$  is an arbitrary vertex of  $G$ , then the graph  $G'$  obtained by adding a new vertex  $v'$  and a new edge  $vv'$  is also in  $\mathcal{F}$ . Chordal graphs are closed for this operation, but interval graphs are not.

**Theorem 6.2.** *Let  $\mathcal{F}$  be a class of graphs closed under attaching pendant vertices. Coloring  $\mathcal{F} + kv$  graphs can be reduced to PREXT with fixed number of colors in the precoloring, if the modulator of the graph is given in the input.*

*Proof.* Given a graph  $G \in \mathcal{F} + kv$  and a set  $C$  of colors, we have to decide whether  $G$  is  $C$ -colorable. It can be assumed that the at most  $k$  special vertices of  $G$  receive colors from  $1, 2, \dots, k$  in the coloring. This means that there are at most  $k^k$  different possibilities for coloring the special vertices. For each such possibility, we check whether the coloring can be extended to the rest of the graph. Clearly,  $G$  is  $C$ -colorable if and only if at least one such coloring is extendible.

We want to use the assumed PREXT algorithm to check whether the coloring of the special vertices can be extended to  $G$ , but  $G$  is not in  $\mathcal{F}$ . Therefore, we modify the graph as follows. Let  $w$  be a special vertex. We attach a new pendant vertex to each non-special neighbor of  $w$ , and assign the color of  $w$  to these new vertices. Now vertex  $w$  can be safely removed, since the new degree 1 vertices ensure that the neighbors of  $w$  do not use the color of  $w$ . Repeating this for every special vertex results in a graph in  $\mathcal{F}$  where at most  $k$  colors are used in the precoloring. Now the assumed algorithm can be used to test whether the precoloring can be extended, completing the reduction.  $\blacksquare$

Next we show that if  $\mathcal{F}$  has some additional properties, then parameterized PREXT on  $\mathcal{F}$  can be reduced to coloring  $\mathcal{F} + kv$  graphs. In the reductions we need to find a graph in  $\mathcal{F}$  with a given chromatic number, this graph will be used as a gadget. The proof could be made simpler if we assumed that  $\mathcal{F}$  contains every clique or that it is easy to find a graph in  $\mathcal{F}$  with a given chromatic number. However, we do not want to restrict the generality of the proof with these assumptions. Therefore, we use the trick that the input graph itself is used to construct the gadget we need.

**Theorem 6.3.** *If  $\mathcal{F}$  is a hereditary graph class closed under disjoint union, then PREXT in  $\mathcal{F}$  with fixed number of precolored vertices can be reduced to the coloring of  $\mathcal{F} + kv$  graphs.*

*Proof.* We are given a graph  $G \in \mathcal{F}$  with a set  $W$  of at most  $k$  precolored vertices and a set  $C$  of colors. The idea is that we consider  $G$  as an  $\mathcal{F} + kv$  graph, where  $G \setminus W \in \mathcal{F}$ , and  $W$  is the set of special vertices. We add additional edges to ensure that the set  $W$  is colored as prescribed by the precoloring. Since these new edges are attached to the special vertices, the new graph will remain a  $\mathcal{F} + kv$  graph.

Let  $\ell = |C_W|$  be the number of distinct colors appearing on the  $k$  precolored vertices. Set  $k' := k$ . First we construct a graph  $H \in \mathcal{F}$  that has chromatic number  $\chi(H) = |C| - \ell$ . The chromatic number of  $G \setminus W$  can be determined by calling an appropriate number of times the assumed algorithm for coloring  $\mathcal{F} + k'v$  graphs (in fact, since  $G \setminus W \in \mathcal{F}$ , an algorithm for coloring graphs in  $\mathcal{F}$  would be enough). Clearly, if  $\chi(G \setminus W) > |C|$ , then there is no solution. On the other hand, if  $\chi(G \setminus W) \leq |C| - \ell$ , then the precoloring extension trivially exists: the precolored vertices use  $\ell$  colors, hence the remaining  $|C| - \ell \geq \chi(G \setminus W)$  colors are sufficient to color  $G \setminus W$ . Therefore, we can assume that  $|C| - \ell < \chi(G \setminus W) \leq |C|$ . To decrease the chromatic number, we start to delete the vertices of  $G \setminus W$  one by one. Since  $\mathcal{F}$  is hereditary, the graph remains in  $\mathcal{F}$ , and its chromatic number can be determined with the assumed algorithm. Deleting a vertex can decrease the chromatic number by at most one. When the chromatic number eventually drops to  $|C| - \ell$ , we get the required graph  $H$ .

The reduction will be done as follows. Add a copy of  $H$  to the graph, and connect each vertex of  $H$  with each vertex of  $W$ . Connect  $w_i \in W$  and  $w_j \in W$  if they are two precolored vertices having different colors. It is clear that the resulting graph  $G'$  is in  $\mathcal{F} + k'v$ : deleting the  $k'$  vertices of  $W$  leaves a graph that is the disjoint union of  $G \setminus W \in \mathcal{F}$  and  $H \in \mathcal{F}$ .

We claim that  $G'$  is  $C$ -colorable if and only if there is a precoloring extension in  $G$ . First, any  $C$ -coloring  $\psi$  of  $G'$  can be turned into a precoloring extension of  $G$  with a permutation of colors. If  $w_i, w_j \in W$  have different colors in the precoloring, then  $\psi(w_i) \neq \psi(w_j)$ , since they are connected in  $G'$ . Since  $W$  induces a complete  $\ell$ -partite graph in  $G'$ , coloring  $\psi'$  assigns at least  $\ell$  colors to the vertices in  $W$ . Moreover,  $\chi(H) = |C| - \ell$  implies that there are at least  $|C| - \ell$  colors on  $H$ . A color cannot appear on both of  $H$  and  $W$ , hence we can conclude that exactly  $\ell$  color appears on  $W$ . This means that those vertices of  $W$  that belong to the same class receive the same color. However, in  $W$  exactly those vertices belong to the same class that have the same color in the precoloring; therefore, the colors in  $\psi$  can be renamed to match the precoloring.

The other direction is also easy to see. To extend a precoloring extension of  $G$  to a  $C$ -coloring of  $G'$ , one has to assign colors to  $H$ . There are exactly  $\ell$  colors appearing on the neighbors of  $H$  (i.e., on  $W$ ), hence the remaining  $|C| - \ell$  colors are sufficient to color  $H$ . ■

Let  $G_1(V_1, E_1)$  and  $G_2(V_2, E_2)$  be two graphs with  $v_1 \in V_1$  and  $v_2 \in V_2$ . The operation of *joining*  $G_1$  and  $G_2$  at  $v_1$  and  $v_2$  means that we construct a new  $|V_1| + |V_2| - 1$  vertex graph by identifying  $v_1$  and  $v_2$ . We say that  $\mathcal{F}$  is closed under joining graphs at a vertex if for every  $G_1, G_2 \in \mathcal{F}$ , the graph formed by joining  $G_1$  and  $G_2$  is in  $\mathcal{F}$ . The class of chordal graphs is closed under joining graphs at a vertex, but interval graphs are not.

**Theorem 6.4.** *If  $\mathcal{F}$  is a hereditary graph class closed under joining graphs at a vertex, then PREXT on  $\mathcal{F}$  with fixed number of colors in the precoloring can be reduced to the coloring of  $\mathcal{F} + kv$  graphs.*

*Proof.* We are given a graph  $G \in \mathcal{F}$  where only  $k$  colors are used on the precolored vertices. The main idea of the reduction is the following. We add a clique of  $k$  special vertices to the graph. Without loss of generality, it can be assumed that the  $i$ -th special vertex receives color  $i$  in every coloring. If there is a vertex  $v$  in  $G$  that is precolored with color  $i$ , then  $v$  is connected to the  $i$ -th special vertex via a gadget that ensures that the two vertices receive the same color.

Let  $C$  be the set of all colors. First we construct a graph  $F$  that satisfies the following properties:

- $\chi(F) = c$  for some  $|C| - k < c \leq |C|$ , and in every  $c$ -coloring of  $F$  the two distinguished vertices  $x$  and  $y$  receive the same color,
- $F \setminus x \in \mathcal{F}$ .

We start by determining the chromatic number of  $G$ , let  $c = \chi(G)$ . As in the proof of Theorem 6.3, we have that  $|C| - k < c \leq |C|$ . Add a new vertex  $x$  to the graph, and connect it with every vertex of  $G$ , the resulting graph has chromatic number  $c + 1$ . Now we start deleting the edges incident to  $x$ , and stop when the chromatic number drops to  $c$  (this will eventually happen, the chromatic number is  $c$  when all the edges incident to  $x$  are deleted). Let  $F$  be the resulting graph, and edge  $xy$  be the last edge deleted. In every  $c$ -coloring of  $F$  the vertices  $x$  and  $y$  have to receive the same color, otherwise it would be a proper  $c$ -coloring of  $F + xy$ , but  $F$  was not  $c$ -colorable before deleting  $xy$ . Moreover,  $F \setminus x = G \in \mathcal{F}$ , hence  $F$  satisfies the required properties.

The reduction is done as follows. We add a clique of size  $k$  to the graph containing the vertices  $v_1, \dots, v_k$  (these vertices will be the special vertices of the constructed  $\mathcal{F} + kv$  graph). If  $v$  is a precolored vertex with color  $i$ , then we join a new copy of  $F$  to the graph by identifying vertex  $x$  of the copy with  $v_i$ , and vertex  $y$  with  $v$ . Moreover, we connect vertices  $v_{i+1}, \dots, v_{i+|C|-c}$  (we use the convention that  $v_{i+k} = v_i$ ) to each vertex of this copy of  $F$  (including vertex  $v$ ). Notice that the resulting graph  $G'$  is in  $\mathcal{F} + kv$ : after deleting the vertices  $v_1, \dots, v_k$ , what remains is the graph  $G \in \mathcal{F}$  with copies of  $F \setminus x \in \mathcal{F}$  joined to some vertices.

We claim that the precoloring can be extended in  $G$  if and only if  $G'$  is  $C$ -colorable. Let  $\psi$  be a  $C$ -coloring of  $G'$ . The vertices  $v_1, \dots, v_k$  form a clique in  $G'$ , they have different colors in  $\psi$ , hence without loss of generality it can be assumed that  $\psi(v_i) = i$  for  $1 \leq i \leq k$ . We show that if  $v$  is a precolored vertex with color  $i$ , then  $\psi(v) = i$ . Consider the copy of  $F$  that connects  $v$  and  $v_i$ . Each vertex of this copy is connected to the vertices  $v_{i+1}, \dots, v_{i+|C|-c}$ . Since exactly  $|C| - c$  colors appear on the vertices  $v_{i+1}, \dots, v_{i+|C|-c}$ , this copy of  $F$  is colored with  $c$  colors. We know that in every  $c$ -coloring of  $F$  the colors of vertices  $x = v_i$

and  $y = v$  are the same, hence  $\psi(v) = i$ , as required. Therefore,  $\psi$  induces a precoloring extension of  $G$ , which completes this direction of the reduction.

The other direction is easy to see: given a precoloring extension on  $G$ , it can be extended to  $G'$  as follows. Set vertex  $v_i$  to color  $i$ . Now the coloring can be extended to each copy of  $F$ : there are  $|C| - c$  colors used on the neighbors, hence  $c$  colors are still available for  $F$ . Furthermore, it is also true that the distinguished vertices  $x$  and  $y$  are assigned the same color. ■

Concerning chordal graphs, putting together Theorem 5.1 and Theorems 6.1–6.3 gives

**Corollary 6.5.** *Coloring chordal+ $ke$  and chordal+ $kv$  graphs can be done in polynomial time for fixed  $k$ , if the modulator is given in the input. However, coloring interval+ $kv$  (hence chordal+ $kv$ ) graphs is W[1]-hard.* ■

In Section 7, we improve on this result by showing that coloring chordal+ $ke$  graphs is fixed-parameter tractable.

## 7 Coloring chordal+ $ke$ graphs

In Theorem 6.1 we have seen that coloring a chordal+ $ke$  graph can be reduced to the solution of at most  $(2k)^{2k}$  PREXT problems on a chordal graph, and by Theorem 5.1, each such problem can be solved in polynomial time. Therefore, chordal+ $ke$  graphs can be colored in polynomial time for every fixed  $k$ . However, this algorithm is not uniformly polynomial: in the running time the exponent of  $n$  depends on  $k$ . In this section, we prove that coloring chordal+ $ke$  graphs is fixed-parameter tractable by presenting a uniformly polynomial time algorithm for the problem.

Let  $H$  be a chordal+ $ke$  graph, and denote by  $G$  the chordal graph obtained by deleting the special edges of  $G$  (it is assumed that the special edges are given in the input). We proceed similarly as in Theorem 5.1. First we construct a nice tree decomposition of  $G$ . A subgraph  $G_x$  of  $G$  corresponds to each node  $x$  of the nice tree decomposition (as defined in Section 4). Let  $H_x$  be the graph  $G_x$  plus the special edges induced by the vertex set of  $G_x$ . For each subgraph  $H_x$ , we try to find a proper coloring. In fact, for every node  $x$  we solve several subproblems: each subproblem corresponds to finding a coloring of  $H_x$  with a given property (to be defined later). The main idea of the algorithm is that the number of subproblems considered at a node can be reduced to a function of  $k$ .

Before presenting the algorithm, we introduce some technical tools that will be useful. For each node  $x$  of the nice tree decomposition, the graph  $H_x^*$  is defined by adding a clique of  $|C| - |K_x|$  vertices  $u_1, u_2, \dots, u_{|C|-|K_x|}$  to the graph  $H_x$ , and connecting each new vertex with each vertex of  $K_x$ . The clique  $K_x$  together with the new vertices form a clique of size  $|C|$ , this clique will be called  $K_x^*$ . Instead of the colorings of  $H_x$ , we will consider the colorings of  $H_x^*$ . Although  $H_x^*$  is a supergraph of  $H_x$ , it is  $C$ -colorable if and only if  $H_x$  is  $C$ -colorable: the new vertices are only connected to  $K_x$ , hence in every coloring of  $H_x$  there remains  $|C| - |K_x|$  colors from  $C$  to color these vertices. However, considering the colorings of  $H_x^*$  instead of the colorings of  $H_x$  will make the arguments cleaner. The reason for this is that in every  $C$ -coloring of  $H_x^*$  every color of  $C$  appears on the clique  $K_x^*$  exactly once, which makes the description of the colorings more uniform.

Another technical trick is that we will assume that every special vertex is contained in exactly one special edge (recall that a vertex is called special if it is the end point of a special edge.) We show how the problem can be converted to a form where this assumption holds. If  $w$  is a special vertex with more than one special edges incident to it, then add a new vertex

$w'$  to the graph. Add also a clique  $K$  of  $|C| - 1$  vertices to the graph, and connect  $w$  and  $w'$  with every vertex of  $K$ . Let  $vw$  be one of the special edges, delete this edge, and add the edge  $vw'$  to the graph instead. It is easy to see that this does not change the  $C$ -colorability of the graph, as in every  $C$ -coloring vertices  $w$  and  $w'$  receive the same color (they are adjacent to the same size  $|C| - 1$  clique). Moreover, the modified graph is also in  $\text{chordal}+ke$ . Repeating the transformation an appropriate number of times, we can ensure, with only a polynomial increase in the size of the graph, that the special edges are independent.

Each special vertex is contained in only one special edge, thus each special vertex  $w$  has a unique *pair*  $w'$ , which is the other vertex of the special edge incident to  $w$ .

## 7.1 Set systems

Now we define the subproblems associated with a node  $x$  of the tree decomposition. A set system is defined where each set corresponds to a type of coloring that is possible on  $H_x^*$ . Let  $W$  be the set of special vertices, we have  $|W| \leq 2k$ . Let  $W_x$  be the special vertices contained in the subgraph  $H_x^*$ . In the following, we consider sets over  $K_x^* \times W$ : each element of the set is a pair  $(v, w)$  with  $v \in K_x^*$ ,  $w \in W$ .

**Definition 7.1.** *To each  $C$ -coloring  $\psi$  of  $H_x^*$ , we associate a set  $S_x(\psi) \subseteq K_x^* \times W$  such that  $(v, w) \in S_x(\psi)$  ( $v \in K_x^*$ ,  $w \in W_x$ ) if and only if  $\psi(v) = \psi(w)$ . The set system  $\mathcal{S}_x$  over  $K_x^* \times W$  contains a set  $S$  if and only if there is a coloring  $\psi$  of  $H_x^*$  such that  $S = S_x(\psi)$ .*

The set  $S_x(\psi)$  describes  $\psi$  on  $H_x^*$  as it is seen from the “outside,” i.e., from  $H \setminus H_x^*$ . In  $H_x^*$  only  $K_x^*$  and  $W_x$  are connected to the outside. Since  $K_x^*$  is a clique of size  $|C|$ , every color appears on exactly one vertex, this is the same for every coloring. Seen from the outside, the only difference between the colorings is how the colors are assigned to  $W_x$ . The set  $S_x(\psi)$  captures this information.

Subgraph  $H_x^*$  (hence  $H_x$ ) is  $C$ -colorable if and only if the set system  $\mathcal{S}_x$  is not empty. Therefore, to decide the  $C$ -colorability of  $H$ , we have to check whether  $\mathcal{S}_r$  is empty, where  $r$  is the root of the nice tree decomposition.

Let us demonstrate Definition 7.1 with the graph shown in Figure 2. Assume that  $x$  is a join node in the tree decomposition and nodes  $y$  and  $z$  are the children of  $x$ . Figures 2b and 2c show the subgraphs  $H_y$  and  $H_z$  (the special edges incident to  $w_1, w_2, w_5, w_6$  are not present in these subgraphs, they appear in the figure only for illustrative purposes). If these two graphs are joined at the clique  $K_y = K_z$ , then we obtain the graph  $H_x$  shown in Figure 2a. Let  $|C| = 3$ , in this case  $H_x = H_x^*$  (since  $|C| = |K_x|$ ).

The graph  $H_y$  has four essentially different colorings. Special vertex  $w_1$  has to receive the same color as either  $v_2$  or  $v_3$ , while vertex  $w_2$  has to receive the same color as either  $v_1$  or  $v_2$ . Vertices  $w_1$  and  $w_2$  are not neighbors, hence any of the four combinations is possible. Therefore, the set system  $\mathcal{S}_y$  contains the following 4 sets:

$$\begin{aligned} S_y(\psi_{y,1}) &= \{(v_2, w_1), (v_1, w_2)\} \\ S_y(\psi_{y,2}) &= \{(v_3, w_1), (v_1, w_2)\} \\ S_y(\psi_{y,3}) &= \{(v_2, w_1), (v_2, w_2)\} \\ S_y(\psi_{y,4}) &= \{(v_3, w_1), (v_2, w_2)\} \end{aligned}$$

In graph  $H_z$ , the special vertices  $w_5$  and  $w_6$  are neighbors, hence either  $w_5$  receives the color of  $v_3$  and  $w_6$  receives the color of  $v_1$ , or vice versa. Vertices  $w_3$  and  $w_4$  has to receive different colors (recall that Definition 7.1 considers the coloring of  $H_x$  and not the chordal

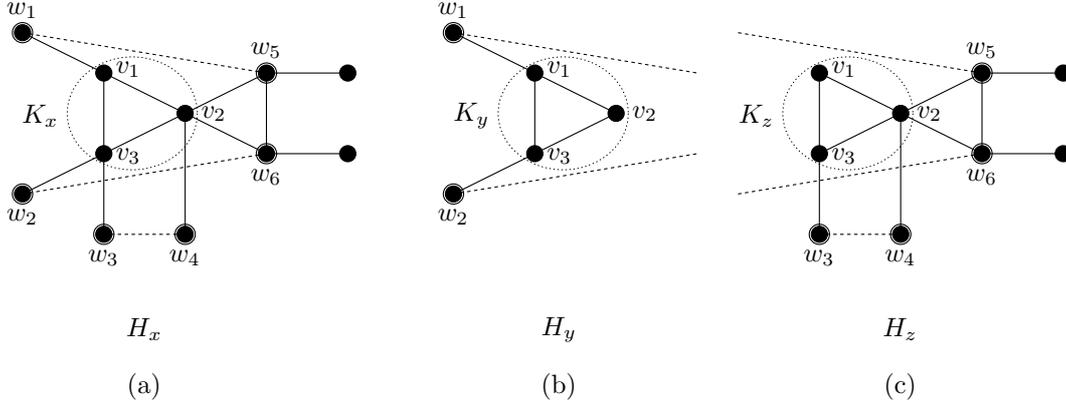


Figure 2: Example graph for defining the set systems.

graph  $G_x$ ), thus there are three different possibilities for coloring them. This means that there are 6 sets in  $\mathcal{S}_x$ :

$$\begin{aligned}
S_z(\psi_{z,1}) &= \{(v_1, w_3), (v_3, w_4), (v_3, w_5), (v_1, w_6)\} \\
S_z(\psi_{z,2}) &= \{(v_1, w_3), (v_3, w_4), (v_1, w_5), (v_3, w_6)\} \\
S_z(\psi_{z,3}) &= \{(v_2, w_3), (v_1, w_4), (v_3, w_5), (v_1, w_6)\} \\
S_z(\psi_{z,4}) &= \{(v_2, w_3), (v_1, w_4), (v_1, w_5), (v_3, w_6)\} \\
S_z(\psi_{z,5}) &= \{(v_2, w_3), (v_3, w_4), (v_3, w_5), (v_1, w_6)\} \\
S_z(\psi_{z,6}) &= \{(v_2, w_3), (v_3, w_4), (v_1, w_5), (v_3, w_6)\}
\end{aligned}$$

The set  $S_x(\psi)$  in Definition 7.1 cannot be an arbitrary subset of  $K_x^* \times W$ , there are certain trivial properties that  $S_x(\psi)$  should satisfy:

**Definition 7.2.** A set  $S \subseteq K_x^* \times W$  is regular, if for every  $w \in W$ , there is at most one element of the form  $(v, w)$  in  $S$ . Moreover, we also require that if  $v \in K_x^* \cap W$  then  $(v, v) \in S$ . The set  $S$  contains special vertex  $w$ , if there is an element  $(v, w)$  in  $S$  for some  $v \in K_x^*$ .

Note that for a coloring  $\psi$  of  $H_x^*$ , the set  $S_x(\psi)$  is regular and contains exactly the vertices in  $W_x$ . In  $S_x(\psi)$ , it is possible that the pairs  $(v, w_1)$  and  $(v, w_2)$  appear with  $w_1 \neq w_2$  (this means that special vertices  $w_1$  and  $w_2$  have the same color as  $v \in K_x^*$ ), but it is not possible that  $(v_1, w)$  and  $(v_2, w)$  appear with  $v_1 \neq v_2$  (that would mean that  $v_1$  and  $v_2$  both have the same color as special vertex  $w$ ).

The definition of  $\mathcal{S}_x$  might seem somewhat technical, but it precisely captures all the information we need from subgraph  $H_x$ . It turns out that the set system for a node can be constructed based on the set systems of its children. In Lemma 7.5, we will prove this in the case of join nodes. But before that we need some further definitions.

**Definition 7.3.** For a set  $S \in K_x^* \times W$ , its blocker  $B(S)$  is a subset of  $K_x^* \times W$  such that  $(v, w) \in B(S)$  if and only if  $(v, w') \in S$  for the pair  $w'$  of  $w$ . We say that sets  $S_1$  and  $S_2$  form a non-blocking pair if  $B(S_1) \cap S_2 = \emptyset$  and  $S_1 \cap B(S_2) = \emptyset$ .

If  $\psi$  is a coloring of  $H_x^*$ , then the set  $B(S_x(\psi))$  describes the requirements that have to be satisfied if we want to extend  $\psi$  to the whole graph. For example, let  $w$  be a special vertex in  $H_x^*$ , whose pair is outside  $H_x^*$ . If  $(v, w) \in S_x(\psi)$ , then this means that  $v \in K_x^*$  has the same

color as special vertex  $w$ . Now  $(v, w') \in B(S_x(\psi))$  for the pair  $w'$  of  $w$ . This tells us that we *should not* color  $w'$  with the same color as  $v$ , because in this case the pairs  $w$  and  $w'$  would have the same color.

For the set system  $\mathcal{S}_y$ , the blockers of the 4 sets are the following:

$$\begin{aligned} B(\{(v_2, w_1), (v_1, w_2)\}) &= \{(v_2, w_5), (v_1, w_6)\} \\ B(\{(v_3, w_1), (v_1, w_2)\}) &= \{(v_3, w_5), (v_1, w_6)\} \\ B(\{(v_2, w_1), (v_2, w_2)\}) &= \{(v_2, w_5), (v_2, w_6)\} \\ B(\{(v_3, w_1), (v_2, w_2)\}) &= \{(v_3, w_5), (v_2, w_6)\} \end{aligned}$$

Similarly, the blockers of the sets in  $\mathcal{S}_z$  are

$$\begin{aligned} B(\{(v_1, w_3), (v_3, w_4), (v_3, w_5), (v_1, w_6)\}) &= \{(v_1, w_4), (v_3, w_3), (v_3, w_1), (v_1, w_2)\} \\ B(\{(v_1, w_3), (v_3, w_4), (v_1, w_5), (v_3, w_6)\}) &= \{(v_1, w_4), (v_3, w_3), (v_1, w_1), (v_3, w_2)\} \\ B(\{(v_2, w_3), (v_1, w_4), (v_3, w_5), (v_1, w_6)\}) &= \{(v_2, w_4), (v_1, w_3), (v_3, w_1), (v_1, w_2)\} \\ B(\{(v_2, w_3), (v_1, w_4), (v_1, w_5), (v_3, w_6)\}) &= \{(v_2, w_4), (v_1, w_3), (v_1, w_1), (v_3, w_2)\} \\ B(\{(v_2, w_3), (v_3, w_4), (v_3, w_5), (v_1, w_6)\}) &= \{(v_2, w_4), (v_3, w_3), (v_3, w_1), (v_1, w_2)\} \\ B(\{(v_2, w_3), (v_3, w_4), (v_1, w_5), (v_3, w_6)\}) &= \{(v_2, w_4), (v_3, w_3), (v_1, w_1), (v_3, w_2)\} \end{aligned}$$

We can see that sets  $S_y(\psi_{y,1}) \in \mathcal{S}_y$  and  $S_z(\psi_{z,1}) \in \mathcal{S}_z$  block each other. This means that the colorings  $\psi_{y,1}$  and  $\psi_{z,1}$  are not compatible:  $\psi_{y,1}$  assigns the same color to  $v_1$  and  $w_2$ , while  $\psi_{z,1}$  assigns the same color  $v_1$  and  $w_6$ , which means that the same color appears on both ends of the special edge  $w_1w_6$ . The incompatibility of  $\psi_{y,1}$  and  $\psi_{z,1}$  is reflected by the fact that  $(v_1, w_2) \in S_y(\psi_{y,1}) \cap B(S_z(\psi_{z,1}))$  and  $(v_1, w_6) \in B(S_y(\psi_{y,1})) \cap S_z(\psi_{z,1})$ .

The sets  $S_y(\psi_{y,1})$  and  $S_z(\psi_{z,2})$  form a non-blocking pair in the example. To be a non-blocking pair, it is sufficient that one of  $B(S_1) \cap S_2$  and  $S_1 \cap B(S_2)$  is empty:

**Lemma 7.4.** *For two sets  $S_1, S_2 \in K_x \times W$ , we have that  $B(S_1) \cap S_2 = \emptyset$  if and only if  $S_1 \cap B(S_2) = \emptyset$ .*

*Proof.* Suppose that  $B(S_1) \cap S_2 = \emptyset$ , but  $(v, w) \in S_1 \cap B(S_2)$  (the other direction follows by symmetry). Since  $(v, w) \in B(S_2)$ , this means that  $(v, w') \in S_2$  where  $w'$  is the pair of  $w$ . But in this case  $(v, w) \in S_1$  implies that  $(v, w') \in B(S_1)$ , contradicting  $B(S_1) \cap S_2 = \emptyset$ . ■

The following lemma motivates the definition of the non-blocking pair. It turns out to be very relevant to our problem: two sets form a non-blocking pair if and only if the corresponding two colorings are compatible. If  $x$  is a join node, then this observation allows us to give a new characterization of  $\mathcal{S}_x$ , based on the set systems of its children.

**Lemma 7.5.** *If  $x$  is a join node with children  $y$  and  $z$ , then*

$$\mathcal{S}_x = \{S_y \cup S_z : S_y \in \mathcal{S}_y \text{ and } S_z \in \mathcal{S}_z \text{ form a non-blocking pair}\}.$$

*Proof.* If  $S \in \mathcal{S}_x$ , then there is a corresponding coloring  $\psi$  of  $H_x^*$ . Coloring  $\psi$  induces a coloring  $\psi_y$  (resp.,  $\psi_z$ ) of  $H_y^*$  (resp.,  $H_z^*$ ). Let  $S_y$  (resp.,  $S_z$ ) be the set that corresponds to coloring  $\psi_y$  (resp.,  $\psi_z$ ). We show that  $S_y$  and  $S_z$  form a non-blocking pair, and  $S = S_y \cup S_z$ . By Lemma 7.4, it is enough to show that  $S_y \cap B(S_z) = \emptyset$ . Suppose that  $S_y \cap B(S_z)$  contains the element  $(v, w)$  for some  $v \in K_y^* = K_z^*$  and  $w \in W_y$ . By the definition of  $S_y$ , this means that  $\psi_y(v) = \psi_y(w)$ . Since  $(v, w) \in B(S_z)$ , thus  $(v, w') \in S_z$  for the pair  $w' \in W$  of  $w$ . Therefore,  $\psi_z(v) = \psi_z(w')$  follows. However,  $\psi_y(v) = \psi_z(v)$ , hence  $\psi_y(w) = \psi_z(w')$ , which

is a contradiction, since  $w$  and  $w'$  are neighbors, and  $\psi$  is a proper coloring of  $H_x^*$ . Now we show that  $S = S_y \cup S_z$ . It is clear that  $(v, w) \in S_y$  implies  $(v, w) \in S$ , hence  $S_y \cup S_z \subseteq S$ . Moreover, suppose that  $(v, w) \in S$ . Without loss of generality, it can be assumed that  $w$  is contained in  $H_y^*$ . This implies that  $(v, w) \in S_y$ , as required.

Now let  $S_y \in \mathcal{S}_y$  and  $S_z \in \mathcal{S}_z$  be a non-blocking pair, it has to be shown that  $S = S_y \cup S_z$  is in  $\mathcal{S}_x$ . Let  $\psi_y$  (resp.,  $\psi_z$ ) be the coloring corresponding to  $S_y$  (resp.,  $S_z$ ). In general,  $\psi_y$  and  $\psi_z$  might assign different colors to the vertices of  $K_x^* = K_y^* = K_z^*$ . However, since  $K_x^*$  is a clique and every color appears exactly once on it, by permuting the colors in  $\psi_y$ , we can ensure that  $\psi_y$  and  $\psi_z$  agree on  $K_x^*$ . We claim that if we merge  $\psi_y$  and  $\psi_z$ , then the resulting coloring  $\psi$  is a proper coloring of  $H_x^*$ . The only thing that has to be verified is whether  $\psi$  assigns different colors to the end vertices of those special edges that are contained completely neither in  $H_y^*$  nor  $H_z^*$ . Suppose that special vertices  $w \in W_y \setminus W_z$  and  $w' \in W_z \setminus W_y$  are pairs, but  $\psi(w) = \psi(w')$ . We know that  $(v, w) \in S_y$  for some  $v \in K_y^*$ , and similarly  $(v', w') \in S_z$  for some  $v' \in K_z^*$ . By definition, this means that  $\psi_y(v) = \psi_y(w)$  and  $\psi_z(v') = \psi_z(w')$ . Since  $\psi_y$  and  $\psi_z$  assign the same colors to the vertices of the clique  $K_x^*$ , this is only possible if  $v = v'$ , implying  $(v, w') \in S_z$ . However, from  $(v, w) \in S_y$  it follows that  $B(S_y)$  also contains  $(v, w')$  contradicting the assumption that  $B(S_y) \cap S_z = \emptyset$ . Now it is straightforward to verify that the set corresponding to  $\psi$  is  $S = S_y \cup S_z$ , proving that  $S \in \mathcal{S}_x$ . ■

Lemma 7.5 gives us a way to obtain the system  $\mathcal{S}_x$  if  $x$  is a join node and the systems for the children are known. It can be shown for introduce nodes and forget nodes as well that their set systems can be constructed if the set systems for their children are given. However, this observation does not lead to a uniformly polynomial algorithm. The problem is that the size of  $\mathcal{S}_x$  can be  $O(n^k)$ , therefore it cannot be represented explicitly. In the following we show that it is not necessary to represent the whole set system, most of the sets can be thrown away, and it is enough to retain only a subsystem whose size can be bounded by a function of  $k$ .

## 7.2 Representative systems

We will replace  $\mathcal{S}_x$  with a system  $\mathcal{S}_x^*$  representative for  $\mathcal{S}_x$  that has size bounded by a function of  $k$ . Representative systems and their use in finding disjoint sets were introduced by Monien [18] (and subsequently used also in [1]).

**Definition 7.6.** *A set system  $\mathcal{S}' \subseteq \mathcal{S}$  is  $q$ -representative for  $\mathcal{S}$  if the following holds: for every set  $B$  of size at most  $q$ , there is a set  $A \in \mathcal{S}$  with  $A \cap B = \emptyset$  if and only if there is a set  $A' \in \mathcal{S}'$  with  $A' \cap B = \emptyset$ . The set system  $\mathcal{S}'$  is minimally representative for  $\mathcal{S}$  if it is representative for  $\mathcal{S}$ , but it is not representative after deleting any of the sets from  $\mathcal{S}'$ .*

For example, if we have the following sets:

$$\begin{array}{cccc} \{a_1, b_1, c_1\} & \{a_1, b_2, c_2\} & \{a_1, b_3, c_3\} & \{a_1, b_4, c_4\} \\ \{a_2, b_1, d_1\} & \{a_2, b_2, d_2\} & \{a_2, b_3, d_3\} & \{a_2, b_4, d_4\} \end{array}$$

then the subsystem

$$\{a_1, b_1, c_1\} \{a_2, b_2, d_2\}$$

is 1-representative. The following subsystem is 2-representative:

$$A_1 = \{a_1, b_1, c_1\} \quad A_2 = \{a_1, b_2, c_2\} \quad A_3 = \{a_2, b_3, d_3\} \quad A_4 = \{a_2, b_4, d_4\}$$

Furthermore, this is a minimally 2-representative subsystem. Set  $A_i$  cannot be thrown away, since there is a set  $B_i$  such that only  $A_i$  is disjoint from  $B_i$ :

$$B_1 = \{a_2, b_2\} \quad B_2 = \{a_2, b_1\} \quad B_3 = \{a_1, b_4\} \quad B_4 = \{a_1, b_3\}$$

The crucial idea is that the size of a minimally  $q$ -representative system can be bounded by a function of  $q$  and the maximum size of the sets  $\mathcal{S}$ . This is a consequence of the following version of Bollobás' inequality:

**Theorem 7.7 (Bollobás [3]).** *Let  $(A_1, B_1), (A_2, B_2), \dots, (A_m, B_m)$  be a sequence of pairs of sets over a common ground set  $X$  such that  $A_i \cap B_j = \emptyset$  if and only if  $i = j$ . Then*

$$\sum_{i=1}^m \binom{|A_i| + |B_i|}{|A_i|}^{-1} \leq 1.$$

**Lemma 7.8.** *If  $\mathcal{S}$  contains sets of size at most  $p$ , and  $\mathcal{S}' \subseteq \mathcal{S}$  is minimally  $q$ -representative for  $\mathcal{S}$ , then  $|\mathcal{S}'| \leq 2^{p+q}$ .*

*Proof.* Let  $\mathcal{S}' = \{A_1, A_2, \dots, A_m\}$ . Subsystem  $\mathcal{S}'$  is minimally representative for  $\mathcal{S}$ , thus for every  $1 \leq i \leq m$ , there is a set  $B_i$  of size at most  $q$  such that  $A_i$  is the only set in  $\mathcal{S}'$  disjoint from  $B_i$  (otherwise  $A_i$  could be safely removed from  $\mathcal{S}'$ ). This means that  $A_i \cap B_i = \emptyset$  for every  $1 \leq i \leq m$ , and  $A_j \cap B_i \neq \emptyset$  for every  $i \neq j$ . Therefore,  $(A_1, B_1), (A_2, B_2), \dots, (A_m, B_m)$  satisfy the requirements of Theorem 7.7, hence

$$1 \geq \sum_{i=1}^m \binom{|A_i| + |B_i|}{|A_i|}^{-1} \geq \sum_{i=1}^m 2^{-(|A_i| + |B_i|)} \geq m 2^{-(p+q)}.$$

Thus  $m \leq 2^{-(p+q)}$ , and the lemma follows. ■

Lemma 7.8 shows that a representative system of size bounded by  $k$  can be obtained by throwing away sets until the system becomes a minimally representative system. However, it is not completely trivial how to check whether a set can be thrown away.

**Lemma 7.9.** *Given a set system  $\mathcal{S}$  containing  $n$  sets of size at most  $p$ , a minimally  $q$ -representative subsystem of  $\mathcal{S}'$  can be found in  $O(p^q \cdot n^2)$  time.*

*Proof.* In the beginning, set  $\mathcal{S}' := \mathcal{S}$ . For each set  $S \in \mathcal{S}'$ , we check whether  $\mathcal{S}'$  remains  $q$ -representative for  $\mathcal{S}$  if  $S$  is removed. If yes, then we remove  $S$  from  $\mathcal{S}'$ . We repeat this until there is no set in  $\mathcal{S}$  that can be removed, in this case  $\mathcal{S}'$  is minimally  $q$ -representative.

Set  $S$  cannot be removed if there is a set  $B$  of size at most  $q$  such that  $S \cap B = \emptyset$ , but  $B$  intersects every other set in  $\mathcal{S}'$ . This question is exactly the HITTING SET problem, which is to find a set of size  $s$  that intersects every set in the given collection of sets. In the parameterized version of the problem the parameter is the size  $s$  of the required set. In general, the HITTING SET problem is W[2]-complete, but fixed-parameter tractable if we have a bound on the size of the sets in the collection. To solve the HITTING SET problem in the case when every set has size at most  $d$ , we use the method of bounded search trees. Let the sets in the collection be ordered in an arbitrary order. At each step of the algorithm, we select the first set that is not already hit by the selected elements. We try to hit this set by adding a new element to the selected elements. Since the set has size at most  $d$ , there are at most  $d$  different possibilities for hitting this set. The algorithm branches off into at most  $d$  directions, by trying all the possibilities. The algorithm has to stop after selecting  $s$  elements, hence the search tree has depth at most  $s$ . At each step we branch off into at most

$d$  directions, therefore the search tree has size  $O(d^s)$ , which is independent of the size of the input. The work to be done at each node of the search tree is linear in the size of the input, hence the HITTING SET problem can be solved in  $O(d^s n)$  time.

In our case, when we check whether the set  $S$  can be thrown away, the size of each set is at most  $p$  and we look for a set of size at most  $q$ . Hence the algorithm described above determines in  $O(p^q |\mathcal{S}'|)$  time whether  $S$  can be removed. To obtain the minimally  $q$ -representative system  $\mathcal{S}'$  for  $\mathcal{S}$ , we have to repeat this procedure at most as many times as the number of sets in the initial set system  $\mathcal{S}$ .  $\blacksquare$

Another way of obtaining a small representative system is to use the data structure of Monien [18] for finding and storing representative systems. Here the size of the resulting set system is somewhat larger, but the running time is linear:

**Lemma 7.10 ([18]).** *Given a set system  $\mathcal{S}$  containing  $n$  sets of size at most  $p$ , a  $q$ -representative subsystem  $\mathcal{S}' \subseteq \mathcal{S}$  of size at most  $\sum_{i=0}^q p^i$  can be found in  $O(pq \cdot \sum_{i=0}^q p^i \cdot n)$  time.*

In our algorithm we adapt the definition of representative systems to our problem:

**Definition 7.11.** *A subsystem  $\mathcal{S}_x^* \subseteq \mathcal{S}_x$  is representative for  $\mathcal{S}_x$  if the following holds: for each regular set  $U \subseteq K_x \times W$  that does not contain vertices from  $W_x \setminus K_x^*$ , if  $\mathcal{S}_x$  contains a set  $S$  disjoint from  $B(U)$ , then  $\mathcal{S}_x^*$  also contains a set  $S'$  disjoint from  $B(U)$ .*

A pair  $(v, w)$  in  $B(U)$  can be interpreted as a requirement that vertex  $v$  should not receive the same color as  $w$ . Therefore, if  $\mathcal{S}_x^*$  is representative for  $\mathcal{S}_x$ , and  $\mathcal{S}_x$  can present a member satisfying all the requirements in  $B(U)$ , then  $\mathcal{S}_x^*$  can present such a member as well. For technical reasons, we are interested only in requirements  $B(U)$  with  $U$  as described above. Since  $B(U)$  has size at most  $2k$ , if  $\mathcal{S}_x^*$  is a  $2k$ -representative for  $\mathcal{S}$ , then it is representative for  $\mathcal{S}$  in the sense of Definition 7.11. Therefore, the algorithms of Lemma 7.9 and 7.10 can be used to find a representative  $\mathcal{S}_x^*$  for a set system  $\mathcal{S}_x$ .

Let us return to our example based on Figure 2. The set system  $\mathcal{S}_y$  is minimally representative of itself. For example, to see that  $S_y(\psi_{y,1}) = \{(v_2, w_1), (v_1, w_2)\}$  cannot be thrown away, consider the set  $U = \{(v_3, w_5), (v_2, w_6)\}$ . Set  $U$  satisfies the requirements of Definition 7.11, and  $S_y(\psi_{y,1})$  is the only set in  $\mathcal{S}_y$  that is disjoint from  $B(U) = \{(v_3, w_1), (v_2, w_2)\}$ . As another example, consider the set  $S_y(\psi_{y,3}) = \{(v_2, w_1), (v_2, w_2)\}$  from  $\mathcal{S}_y$ . To see that it cannot be thrown away, let  $U$  be  $S_z(\psi_{z,1}) = \{(v_1, w_3), (v_3, w_4), (v_3, w_5), (v_1, w_6)\}$ . Now  $S_y(\psi_{y,3})$  is the only set in  $\mathcal{S}_y$  disjoint from  $B(S_z(\psi_{z,1}))$ . Intuitively, this means that if we want to extend coloring  $\psi_{z,1}$  of  $H_z$  to the whole  $H_x$ , then  $\psi_{y,3}$  is the only coloring of  $H_y$  that is compatible with  $\psi_{z,1}$ . Therefore, when we consider the possible colorings of  $H_y$ , then  $\psi_{y,3}$  cannot be thrown away, since it can be essential in some cases.

The set system  $\mathcal{S}_z$  is not minimally representative, the following two sets are representative of  $\mathcal{S}_z$ :

$$\begin{aligned} S_z(\psi_{z,1}) &= \{(v_1, w_3), (v_3, w_4), (v_3, w_5), (v_1, w_6)\} \\ S_z(\psi_{z,2}) &= \{(v_1, w_3), (v_3, w_4), (v_1, w_5), (v_3, w_6)\} \end{aligned}$$

If the set  $U$  satisfies the requirements of Definition 7.11, then  $B(U)$  cannot contain  $w_3$  or  $w_4$ . Therefore, there is no use of adding another set from  $\mathcal{S}_z$  to the two sets above: if we disregard  $w_3$  and  $w_4$ , then every other set in  $\mathcal{S}_z$  is equivalent to  $S_z(\psi_{z,1})$  and  $S_z(\psi_{z,2})$ .

We show that instead of determining the set system  $\mathcal{S}_x$  for each node, it is sufficient to find a set system  $\mathcal{S}_x^*$  representative for  $\mathcal{S}_x$ . That is, if for each child  $y$  of  $x$  we are given a system  $\mathcal{S}_y^*$  representative for  $\mathcal{S}_y$ , then we can construct a system  $\mathcal{S}_x^*$  representative for  $\mathcal{S}_x$ .

For a join node  $x$ , one can find a set system  $\mathcal{S}_x^*$  representative for  $\mathcal{S}_x$  by a characterization analogous to Lemma 7.5:

**Lemma 7.12.** *Let  $x$  be a join node with children  $y$  and  $z$ , and let  $\mathcal{S}_y^*$  be representative for  $\mathcal{S}_y$ , and  $\mathcal{S}_z^*$  representative for  $\mathcal{S}_z$ . Then the system*

$$\mathcal{S}_x^* = \{S_y \cup S_z : S_y \in \mathcal{S}_y^* \text{ and } S_z \in \mathcal{S}_z^* \text{ form a non-blocking pair}\}$$

*is representative for  $\mathcal{S}_x$ .*

*Proof.* Since  $\mathcal{S}_y^* \subseteq \mathcal{S}_y$  and  $\mathcal{S}_z^* \subseteq \mathcal{S}_z$ , by Lemma 7.5 it follows that  $\mathcal{S}_x^* \subseteq \mathcal{S}_x$ . Therefore, we have to show that for every regular set  $U$  not containing vertices from  $W_x \setminus K_x^*$ , if there is a set  $S \in \mathcal{S}_x$  disjoint from  $B(U)$ , then there is a set  $S' \in \mathcal{S}_x^*$  also disjoint from  $B(U)$ . Let  $\psi$  be the coloring of  $H_x^*$  corresponding to set  $S$ , and let  $\psi_y$  (resp.,  $\psi_z$ ) be the coloring of  $H_y^*$  (resp.,  $H_z^*$ ) induced by  $\psi$ . Let  $S_y \in \mathcal{S}_y$  and  $S_z \in \mathcal{S}_z$  be the sets corresponding to  $\psi_y$  and  $\psi_z$ . We have seen in the proof of Lemma 7.5 that  $S_y$  and  $S_z$  form a non-blocking pair and  $S = S_y \cup S_z$ , hence  $S_y$  is disjoint from  $B(U) \cup B(S_z) = B(U \cup S_z)$ . Note that  $U$  does not contain vertices from  $W_x \setminus K_x^*$ , and  $S_z$  contains vertices only from  $W_z$ , hence  $U \cup S_z$  is regular, and does not contain vertices from  $W_y \setminus K_y^*$ . Since  $\mathcal{S}_y^*$  is representative for  $\mathcal{S}_y$ , there is a set  $S'_y \in \mathcal{S}_y^*$  that is also disjoint from  $B(U \cup S_z)$ . By Lemma 7.4,  $S'_y \cap B(S_z) = \emptyset$  implies that  $B(S'_y) \cap S_z = \emptyset$ , hence  $S_z$  is disjoint from  $U \cup B(S'_y) = B(U \cup S'_y)$ . Since  $\mathcal{S}_z^*$  is representative for  $\mathcal{S}_z$ , there is a set  $S'_z \in \mathcal{S}_z^*$  that is also disjoint from  $B(U \cup S'_y)$ . Applying again Lemma 7.4, we get that  $S'_y$  and  $S'_z$  form a non-blocking pair, hence  $S' = S'_y \cup S'_z$  is in  $\mathcal{S}_x^*$ . The set  $S'$  is disjoint from  $B(U)$ , thus  $\mathcal{S}_x^*$  contains a set disjoint from  $B(U)$ . ■

### 7.3 The algorithm

Now we are ready to prove the main result of the section. In Theorem 7.13 we put the pieces together to obtain a uniformly polynomial algorithm for coloring chordal+ke graphs.

**Theorem 7.13.** *Coloring chordal+ke graphs is in FPT if the modulator of the graph is given in the input.*

*Proof.* The first step of the algorithm is to find a nice tree decomposition of the chordal graph  $G$ , where  $G$  is the input graph  $H$  minus the special edges. For each node  $x$  of the nice tree decomposition, we construct a set system  $\mathcal{S}_x^*$  over  $K_x^* \times W$  that is minimally representative for  $\mathcal{S}_x$ . Clearly,  $H$  is  $C$ -colorable if and only if  $\mathcal{S}_r^*$  is non-empty for the root  $r$  (note that if  $\mathcal{S}_r$  is non-empty, then  $\mathcal{S}_r^*$  cannot be empty).

The systems  $\mathcal{S}_x^*$  are constructed using bottom-up dynamic programming. First we construct the systems for the leaves. For every non-leaf node  $x$ , we assume that the children of  $x$  are ready when we construct  $\mathcal{S}_x^*$ . Below we describe what has to be done for the different types of nodes. For join nodes the construction follows easily from Lemma 7.12. The case of introduce nodes and forget nodes are conceptually not difficult, but requires a tedious discussion.

**Leaf node  $x$ .** If  $x$  is a leaf node, then  $H_x$  has only one vertex, and  $H_x^*$  is a complete graph. Therefore,  $H_x^*$  has only one coloring (up to permutation of colors), and  $\mathcal{S}_x$  has only one set. More precisely, if vertex  $v$  of  $H_x$  is special vertex, then  $\mathcal{S}_x$  contains only the set  $(v, v)$ ; if  $v$  is not a special vertex, then  $\mathcal{S}_x$  contains only the empty set.

**Introduce node  $x$  of vertex  $v$ .** Let  $y$  be the child of  $x$ . The only difference between the set  $K_x^*$  and  $K_y^*$  is that  $K_x^*$  contains  $v$ , but does not contain  $u_{|C|-|K_y|}$ . Therefore, a subset  $S \subseteq K_y^* \times W$  can be mapped to a subset  $S' \subseteq K_x^* \times W$  by mapping  $u_{|C|-|K_y|}$  to  $v$ . That is, let  $(v, w) \in S'$  if and only if  $(u_{|C|-|K_y|}, w) \in S$ . In the following, when mapping sets between  $K_y^* \times W$  and  $K_x^* \times W$ , then we will mean this mapping.

We consider three cases:

*Case 1.* If  $v \neq W$ , then  $\mathcal{S}_x$  and  $\mathcal{S}_y$  are essentially the same,  $\mathcal{S}_y$  can be obtained by mapping  $u_{|C|-|K_y|}$  to  $v$  in each set of  $\mathcal{S}_y$ . To see this, notice that  $H_x^*$  and  $H_y^*$  are in fact the same graph:  $H_x$  contains one vertex more than  $H_y$ , but we attach to  $H_x$  one vertex less than to  $H_y$ , since  $K_x$  is greater than  $K_y$  by one. Thus the colorings of  $H_x^*$  and  $H_y^*$  are in one-to-one correspondence, vertex  $v$  in  $H_x^*$  corresponds to vertex  $u_{|C|-|K_y|}$  in  $H_y^*$ . Therefore, each set in  $\mathcal{S}_x$  corresponds to a set in  $\mathcal{S}_y$ , via the mapping described above. Moreover, this also means that  $\mathcal{S}_x^*$  can be obtained in a similar way, by mapping each set in  $\mathcal{S}_y^*$ .

*Case 2.* If  $v \in W$  and the pair of  $v$  is not in  $W_x$ , then  $H_x^*$  and  $H_y^*$  are still the same, but  $\mathcal{S}_x$  cannot be obtained from  $\mathcal{S}_y$  as in the previous case. The reason for this is that  $W_x = W_y \cup \{v\}$ , hence different sets correspond to the same coloring in  $\mathcal{S}_x$  and  $\mathcal{S}_y$ . More precisely, each set in  $\mathcal{S}_x$  contains  $(v, v)$ . The set system  $\mathcal{S}_x$  can be obtained from  $\mathcal{S}_y$  by mapping each set of  $\mathcal{S}_y$  to  $K_x^* \times W$ , and by adding the pair  $(v, v)$  to each set. It is easy to show that  $\mathcal{S}_x^*$  can be obtained from  $\mathcal{S}_y^*$  the same way.

*Case 3.* If  $v \in W$ , and the pair  $v'$  of the special vertex  $v$  is in  $W_x$ , then  $H_x^*$  and  $H_y^*$  are not the same:  $H_x^*$  contains one edge more, namely the special edge  $vv'$ . This means that only those colorings of  $H_y^*$  yield a coloring of  $H_x^*$  that assign different colors to  $u_{|C|-|K_y|} \in K_y^*$  and  $v' \in W$ . Therefore, first we throw away those sets of  $\mathcal{S}_y^*$  that contain  $(u_{|C|-|K_y|}, v')$ , and we proceed with the remaining sets as in the previous case. To see that the resulting system  $\mathcal{S}_x^*$  is indeed representative of  $\mathcal{S}_x$ , consider a set  $U$  as in Definition 7.11, and let  $S \in \mathcal{S}_x$  be disjoint from  $B(U)$ . Let  $\psi$  be the coloring of  $H_x^*$  corresponding to  $S$ , clearly  $\psi(v) \neq \psi(v')$ . Coloring  $\psi$  can be used to define a coloring  $\psi'$  of  $H_y^*$  by setting  $\psi'(u_{|C|-|K_y|}) = \psi(v)$ . If  $R$  is the set corresponding to  $\psi'$ , then  $R$  does not contain the pair  $(u_{|C|-|K_y|}, v')$  since  $\psi'(u_{|C|-|K_y|}) \neq \psi'(v')$ . Therefore,  $R \in \mathcal{S}_y$  is disjoint from  $B(U')$  where  $U' := U \cup \{(u_{|C|-|K_y|}, v)\}$ . Set system  $\mathcal{S}_y^*$  is assumed to be representative for  $\mathcal{S}_y$ , hence  $\mathcal{S}_y^*$  contains a set  $R'$  disjoint from  $B(U')$  (notice the  $U'$  satisfies the requirements of Definition 7.11:  $v \notin W_y \setminus K_y^*$ ). This implies that during the construction of  $\mathcal{S}_x^*$ , the set  $R' \in \mathcal{S}_y^*$  contributes a set to  $\mathcal{S}_x^*$  that is disjoint from  $B(U)$ , as required.

**Forget node  $x$  of vertex  $v$ .** Let  $\psi$  be a coloring of  $H_x^*$  and let  $u_i$  ( $1 \leq i \leq |C| - |K_x|$ ) be a vertex of  $K_x^* \setminus K_x$ . Define  $\psi[u_i]$  to be the coloring where the colors of  $u_i$  and  $u_{|C|-|K_x|}$  are exchanged. Notice that  $\psi[u_i]$  is a proper coloring since the neighborhoods of  $u_i$  and  $u_{|C|-|K_x|}$  are the same (the clique  $K_x^*$ ). If  $S \subseteq K_x^* \times W$  is the set corresponding to  $\psi$ , then the set  $S[u_i]$  corresponding to  $\psi[u_i]$  can be obtained by interchanging the role of  $u_i$  and  $u_{|C|-|K_x|}$ , that is,  $(u_i, w) \in S[u_i]$  if and only if  $(u_{|C|-|K_x|}, w) \in S$ ; and  $(u_{|C|-|K_x|}, w) \in S[u_i]$  if and only if  $(u_i, w) \in S$ .

Given a set system  $\mathcal{S}_y^*$  representative for  $\mathcal{S}_y$ , we construct  $\mathcal{S}_x^*$  as follows. The only difference between the set  $K_x^*$  and  $K_y^*$  is that  $K_x^*$  contains  $u_{|C|-|K_x|}$ , but it does not contain  $v$ . As in the case of introduce nodes, we give a mapping from  $K_y^* \times W$  to  $K_x^* \times W$ , but now we map  $v \in K_y^*$  to  $u_{|C|-|K_x|} \in K_x^*$ . The set system  $\mathcal{S}_x^*$  is constructed as follows: for each  $S \in \mathcal{S}_y^*$ , take the corresponding set  $S' \subseteq K_x^* \times W$ , and add to  $\mathcal{S}_x^*$  the sets  $S'[u_i]$  for  $1 \leq i \leq |C| - |K_x|$ .

To prove that the resulting system  $\mathcal{S}_x^*$  is representative for  $\mathcal{S}_x$ , we show first that  $\mathcal{S}_x^* \subseteq \mathcal{S}_x$ . Let  $S[u_i]$  be a set from  $\mathcal{S}_x^*$ , where  $S \subseteq K_y \times W$  is obtained from some set  $S' \in \mathcal{S}_y^*$ .

Therefore, there is a coloring  $\psi$  of  $H_y^*$  that corresponds to  $S'$ . Coloring  $\psi$  can be extended to a coloring  $\psi'$  of  $H_x^*$  by setting  $\psi'(u_{|C|-|K_x|}) = \psi(v)$ . Now the set  $S[u_i]$  corresponds to coloring  $\psi'[u_i]$  of  $H_x^*$ , hence  $S[u_i] \in \mathcal{S}_x$ .

Assume that for some  $U \subseteq K_x^* \times W$  satisfying Definition 7.11, there is a set  $S \in \mathcal{S}_x$  that is disjoint from  $B(U)$ . It has to be shown that  $\mathcal{S}_x^*$  also has such a set. Let  $\psi$  be the coloring corresponding to set  $S$ . In coloring  $\psi$ , one of the vertices  $u_1, \dots, u_{|C|-|K_x|}$  has the same color as  $v$  (the vertices in  $K_x$  cannot have this color, since they are adjacent to  $v$ ), assume that  $\psi(u_i) = \psi(v)$ . Consider the coloring  $\psi[u_i]$ , clearly the set corresponding to it is  $S[u_i]$ , which is disjoint from  $B(U[u_i])$ . Moreover,  $\psi[u_i]$  induces a proper coloring of  $H_y^*$  (since the color of  $v$  does not appear on  $K_x^* \setminus \{u_{|C|-|K_x|}\}$ ). The set corresponding to this coloring is  $S'[u_i]$  (which is obtained by mapping  $S[u_i]$  from  $K_x^* \times W$  to  $K_y^* \times W$ ), and this set is disjoint from  $B(U'[u_i])$  (where  $U'[u_i]$  is obtained by mapping  $U[u_i]$ ). Therefore,  $\mathcal{S}_y$  has a set disjoint from  $B(U'[u_i])$ . By assumption,  $\mathcal{S}_y^*$  is representative for  $\mathcal{S}_y$ , hence there is a set  $R' \in \mathcal{S}_y^*$  also disjoint from  $B(U'[u_i])$ . Mapping  $R'$  from  $K_y^* \times W$  to  $K_x^* \times W$  yields a set  $R$  disjoint from  $B(U[u_i])$ , hence the set  $R[u_i]$  is disjoint from  $B(U)$ . When  $\mathcal{S}_x^*$  is constructed, we add the set  $R[u_i]$  to the system, thus  $\mathcal{S}_x^*$  also has a set disjoint from  $B(U)$ , what we had to prove.

**Join node  $x$ .** Let  $y$  and  $z$  be the children of  $x$ . By Lemma 7.12, the set system

$$\mathcal{S}_x^* = \{S_y \cup S_z : S_y \in \mathcal{S}_y^* \text{ and } S_z \in \mathcal{S}_z^* \text{ form a non-blocking pair}\}$$

is representative for  $\mathcal{S}_x$ . This set system can be easily constructed if we consider each pair  $S_y \in \mathcal{S}_y^*$  and  $S_z \in \mathcal{S}_z^*$ , and add their union into  $\mathcal{S}_x^*$  only if they form a non-blocking pair.

**Running time.** For each type of node  $x$ , we have shown how to obtain a set system  $\mathcal{S}_x^*$  representative for  $\mathcal{S}_x$ . After constructing  $\mathcal{S}_x^*$ , the algorithm of Lemma 7.9 or Lemma 7.10 can be used to reduce the size of the set system to a function of  $k$ . This will ensure that we have to work with small set systems at each step. Therefore, the algorithm takes uniformly polynomial time at each node, and it follows that coloring chordal+ $ke$  graphs is fixed-parameter tractable.  $\blacksquare$

## 8 Conclusions

We have considered two different types of coloring problems in this paper. Precoloring extension already received a lot of attention in the literature. Here we investigated two possible parameterizations of the problem. Following Cai [5], we also investigated the parameterized complexity of coloring if the parameter is the distance of the graph from some “nice” class of graphs. The paper demonstrated that these two types problems are intimately related and should be studied together.

Besides presenting general connections between the different problems, we obtained concrete results for chordal and interval graphs. We have determined the complexity of the two parameterizations of PREXT on chordal and interval graphs, and the parameterized complexity of coloring chordal+ $ke$ , chordal+ $kv$ , interval+ $ke$ , interval+ $kv$  graphs. It would be interesting to study the analogous problems for proper interval, comparability, co-comparability, and permutation graphs.

The problem of finding modulators also seems to be worth studying. The “iterative compression” technique, introduced recently in [21], could be very useful for designing such algorithms.

## Acknowledgments

I'm grateful to Ildi Schlotter for her careful reading of the manuscript.

## References

- [1] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
- [2] M. Biró, M. Hujter, and Zs. Tuza. Precoloring extension. I. Interval graphs. *Discrete Math.*, 100(1-3):267–279, 1992.
- [3] B. Bollobás. On generalized graphs. *Acta Math. Acad. Sci. Hungar.*, 16:447–452, 1965.
- [4] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inform. Process. Lett.*, 58(4):171–176, 1996.
- [5] L. Cai. Parameterized complexity of vertex colouring. *Discrete Appl. Math.*, 127:415–429, 2003.
- [6] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. In *Proceedings of 19th Annual IEEE Conference on Computational Complexity*, pages 150–160, 2004.
- [7] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.
- [8] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980.
- [9] M. Grohe and J. Flum. Parameterized complexity and subexponential time. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, (84):71–100, 2004.
- [10] J. Guo, F. Hüffner, and R. Niedermeier. A structural view on parameterizing problems: distance from triviality. In *Proceedings of the International Workshop on Parameterized and Exact Computation (IWPEC 2004)*, volume 3162 of *Lecture Notes in Comput. Sci.*, pages 162–173, Berlin, 2004. Springer.
- [11] M. Hujter and Zs. Tuza. Precoloring extension. II. Graph classes related to bipartite graphs. *Acta Mathematica Universitatis Comenianae*, 62(1):1–11, 1993.
- [12] M. Hujter and Zs. Tuza. Precoloring extension. III. Classes of perfect graphs. *Combin. Probab. Comput.*, 5(1):35–56, 1996.
- [13] H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28(5):1906–1922, 1999.
- [14] T. Kloks. *Treewidth*, volume 842 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1994.
- [15] D. König. Über graphen und ihre andwendung auf determinantentheorie und mengenlehre. *Math. Ann.*, 77:453–465, 1916.

- [16] D. Marx. Precoloring extension on chordal graphs, 2004. Manuscript. Preliminary version presented at Graph Theory 2004, Paris.
- [17] D. Marx. Precoloring extension on unit interval graphs, 2004. Submitted.
- [18] B. Monien. How to find long paths efficiently. In *Analysis and design of algorithms for combinatorial problems (Udine, 1982)*, volume 109 of *North-Holland Math. Stud.*, pages 239–254. North-Holland, Amsterdam, 1985.
- [19] R. Niedermeier. Invitation to fixed-parameter algorithms, 2002. Habilitation thesis, Universität Tübingen.
- [20] B. Reed. Edge coloring nearly bipartite graphs. *Oper. Res. Lett.*, 24(1-2):11–14, 1999.
- [21] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.
- [22] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976.
- [23] A. Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. In *Algorithms – ESA 2003, 11th Annual European Symposium*, volume 2832 of *Lecture Notes in Comput. Sci.*, pages 482–493. Springer, Berlin, 2003.