HAL
open science

Bridge-Depth Characterizes which Minor-Closed Structural Parameterizations of Vertex Cover Admit a Polynomial Kernel

Marin Bougeret, Bart M P Jansen, Ignasi Sau

▶ **To cite this version:**

Marin Bougeret, Bart M P Jansen, Ignasi Sau. Bridge-Depth Characterizes which Minor-Closed Structural Parameterizations of Vertex Cover Admit a Polynomial Kernel. SIAM Journal on Discrete Mathematics, 2022, 36 (4), pp.2737-2773. 10.1137/21m1400766 . lirmm-04028294

HAL Id: lirmm-04028294

https://hal-lirmm.ccsd.cnrs.fr/lirmm-04028294v1

Submitted on 14 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bridge-Depth Characterizes which Minor-Closed Structural Parameterizations of Vertex Cover Admit a Polynomial Kernel

Marin Bougeret
LIRMM, Université de Montpellier, France
marin.bougeret@lirmm.fr

Bart M. P. Jansen
Eindhoven University of Technology, The Netherlands
b.m.p.jansen@tue.nl

Ignasi Sau
LIRMM, Université de Montpellier, CNRS, France
ignasi.sau@lirmm.fr

Abstract

We study the kernelization complexity of structural parameterizations of the Vertex Cover problem. Here, the goal is to find a polynomial-time preprocessing algorithm that can reduce any instance (G, k) of the Vertex Cover problem to an equivalent one, whose size is polynomial in the size of a pre-determined complexity parameter of G. A long line of previous research deals with parameterizations based on the number of vertex deletions needed to reduce G to a member of a simple graph class $\mathcal{F}$, such as forests, graphs of bounded tree-depth, and graphs of maximum degree two. We set out to find the most general graph classes $\mathcal{F}$ for which Vertex Cover parameterized by the vertex-deletion distance of the input graph to $\mathcal{F}$, admits a polynomial kernelization. We give a complete characterization of the minor-closed graph families $\mathcal{F}$ for which such a kernelization exists. We introduce a new graph parameter called *bridge-depth*, and prove that a polynomial kernelization exists if and only if $\mathcal{F}$ has bounded bridge-depth. The proof is based on an interesting connection between bridge-depth and the size of minimal blocking sets in graphs, which are vertex sets whose removal decreases the independence number.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases vertex cover, parameterized complexity, polynomial kernel, structural parameterization, bridge-depth.

Related Version A conference version of this paper appeared in the *Proceedings of the 47th International Colloquium on Automata, Languages and Programming (**ICALP**), volume 168 of LIPIcs, pages 16:1-16:19, July **2020**.*

Funding *Bart M. P. Jansen*: Supported by ERC Starting Grant ReduceSearch (Grant Agreement No 803421).
Ignasi Sau: Supported by French projects DEMOGRAPH (ANR-16-CE40-0028), ESIGMA (ANR-17-CE23-0010), and ELIT (ANR-20-CE48-0008-01).

1 Introduction

Background and motivation. The NP-complete Vertex Cover problem is one of the most prominent problems in the field of kernelization [3, 7, 12, 16, 26], which investigates provably efficient and effective preprocessing for parameterized problems. A *parameterized problem* is a decision problem in which a positive integer k, called the *parameter*, is associated with every instance x. A *kernelization* for a parameterized problem is a polynomial-time algorithm that reduces any parameterized instance (x, k) to an equivalent instance (x', k') of the same problem whose size is bounded by $f(k)$ for some function f, which is the *size* of the kernelization. Hence a kernelization guarantees that instances that are large compared to their parameter, can be efficiently reduced without changing their answer. Of particular interest are *polynomial kernelizations*, whose size bound f is polynomial.

An instance (G, k) of Vertex Cover asks whether the undirected graph G has a vertex set S of size at most k that contains at least one endpoint of every edge. Using the classic Nemhauser-Trotter theorem [28], one can reduce (G, k) in polynomial time to an instance (G', k') with the same answer, such that $|V(G')| \leq 2k$. Hence when using the size of the desired solution as the parameter, Vertex Cover has a kernelization that reduces to instances of $2k$ vertices, which can be encoded in $\mathcal{O}(k^2)$ bits. While the bitsize of this kernelization is known to be essentially optimal [9] assuming the established conjecture $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$, this result does not guarantee any effect of the preprocessing for instances whose solution has size at least $|V(G)|/2$. In particular, it does not promise any size reduction when G is simply a path.

To be able to give better preprocessing guarantees, one can use *structural parameters* that take on smaller values than the size of a minimum vertex cover, a quantity henceforth called the *vertex cover number*. Such structural parameterizations can conveniently be described in terms of the vertex-deletion distance to certain graph families $\mathcal{F}$. Note that the vertex cover number $\textsc{vc}(G)$ of G can be defined as the minimum number of vertex deletions needed to reduce G to an edgeless graph. Hence this number will always be at least as large as the *feedback vertex number* $\textsc{fvs}(G)$ of G, which is the vertex-deletion distance of G to a forest. In 2011, it was shown that Vertex Cover even admits a polynomial kernelization when parameterized by the feedback vertex number [21, 22]. This triggered a long line of follow-up research, which aimed to find the most general graph families $\mathcal{F}$ such that Vertex Cover admits a polynomial kernelization when parameterized by vertex-deletion distance to $\mathcal{F}$. Polynomial kernelizations were obtained for the families $\mathcal{F}$ of graphs of maximum degree two [27], of graphs of constant *tree-depth* [4, 23], of the *pseudo-forests* where each connected component has at most one cycle [17], and for *d-pseudo-forests* in which each connected component has a feedback vertex set of size at most $d \in \mathcal{O}(1)$ [18]. Note that all these target graph classes are *closed under taking minors*. Using *randomized algorithms* with a small error probability, polynomial kernelizations are also known for several parameterizations by vertex-deletion distance to graph classes that are not minor-closed, such as Kőnig graphs [25], bipartite graphs [25], and parameterizations based on the linear-programming relaxation of Vertex Cover [19, 24]. On the negative side, it is known that Vertex Cover parameterized by the vertex-deletion distance to a graph of treewidth two [8] does *not* have a polynomial kernel, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. This long line of research into kernelization for structural parameterizations raises the following question:

> How can we characterize the graph families $\mathcal{F}$ for which Vertex Cover parameterized by vertex-deletion distance to $\mathcal{F}$ admits a polynomial kernel?

We answer this question for all minor-closed families $\mathcal{F}$, by introducing a new graph parameter.

Our results. We introduce a new graph parameter that we call *bridge-depth*. It has a recursive definition similar to that of tree-depth [30] (full definitions follow in Section 3), but deals with bridges in a special way. A graph without vertices has bridge-depth zero. The bridge-depth $\mathsf{bd}(G)$ of a disconnected graph G is simply the maximum bridge-depth of its connected components. The bridge-depth of a connected nonempty graph G is defined as follows. Let G_{CB} denote the graph obtained from G by contracting each edge that is a bridge in G; the order does not matter. Then $\mathsf{bd}(G) := 1 + \min_{v \in V(G_{\text{CB}})} \mathsf{bd}(G_{\text{CB}} \setminus v)$. Intuitively, the bridge-depth of G is given by the depth of an elimination process [5] that reduces G to the empty graph. One step consists of contracting all bridges and removing a vertex; each of the remaining connected components is then recursively eliminated in parallel. From this definition, it is not difficult to see that $\mathsf{bd}(G)$ is at least as large as the *tree-width* of G, but never larger than the tree-depth or feedback vertex number of G. In particular, any forest has bridge-depth one.

Using the notion of bridge-depth, we characterize the minor-closed families $\mathcal{F}$ for which Vertex Cover parameterized by vertex-deletion distance to $\mathcal{F}$ admits a polynomial kernel.

▶ **Theorem 1.1.** *Let $\mathcal{F}$ be a minor-closed family of graphs, and assume $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$. Vertex Cover parameterized by vertex-deletion distance to $\mathcal{F}$ has a polynomial kernelization if and only if $\mathcal{F}$ has bounded bridge-depth.*

Theorem 1.1 gives a clean and unified explanation for all the minor-closed families $\mathcal{F}$ that were previously considered individually [4, 17, 18, 22, 27], and generalizes these results as far as possible. To the best of our knowledge, Theorem 1.1 captures all known (deterministic) kernelizations for structural parameterizations of Vertex Cover. (There are *randomized* kernelizations [19, 24, 25] that apply for distance to classes $\mathcal{F}$ that are *not* minor-closed, such as bipartite graphs.) For example, we capture the case of $\mathcal{F}$ being a forest [22] since forests have bridge-depth one, and the case of $\mathcal{F}$ being graphs of constant tree-depth [4, 23] since bridge-depth does not exceed tree-depth. In this sense, bridge-depth can be seen as the ultimate common generalization of feedback vertex number and tree-depth (which are incomparable parameters) in the context of polynomial kernels for Vertex Cover.

We consider it one of our main contributions to identify the graph parameter bridge-depth as the right way to capture the kernelization complexity of Vertex Cover parameterizations.

Techniques. To describe our techniques, we introduce some terminology. Let $\alpha(G)$ denote the independence number of graph G, i.e., the maximum size of a set of pairwise nonadjacent vertices. A *blocking set* in a graph G is a vertex set $Y \subseteq V(G)$ such that $\alpha(G \setminus Y) < \alpha(G)$. Hence if Y is a blocking set, then every maximum independent set in G contains a vertex from Y. The connection between blocking sets and the kernelization of Vertex Cover has been already exploited in previous work. Namely, earlier kernelizations for Vertex Cover parameterized by distance to a graph class $\mathcal{F}$, starting with the work of Jansen and Bodlaender [22], all rely, either implicitly or explicitly, on having upper-bounds on the size of (inclusion-)minimal blocking sets for graphs in $\mathcal{F}$ [4, 17, 18, 22, 27]. For example, it is known that minimal blocking sets in a bipartite graph have size at most two [18, Cor. 11], while minimal blocking sets in graphs of tree-depth c have size at most 2^c [4, Lemma 1]. Similarly, all the existing superpolynomial kernelization lower bounds for parameterizations by distance to $\mathcal{F}$, rely on $\mathcal{F}$ having minimal blocking sets of arbitrarily large size. Indeed, if $\mathcal{F}$ is closed under disjoint union and has arbitrarily large blocking sets, it is easy to prove a superpolynomial lower bound (cf. [19, Thm. 1]).

Since all positive cases for kernelization are when minimal blocking sets of graphs in $\mathcal{F}$ have bounded size, while one easily obtains lower bounds when the size of minimal blocking

sets of graphs in $\mathcal{F}$ is unbounded, the question rises whether a bound on the size of minimal blocking sets is a necessary and sufficient condition for the existence of polynomial kernels. To our initial surprise, we show that for minor-closed families $\mathcal{F}$, this is indeed the case: the purely structural property of having bounded-size minimal blocking sets can always be leveraged into preprocessing algorithms.

For an insight into our techniques, consider an instance (G, k) of Vertex Cover, together with a vertex set $X \subseteq V(G)$ such that $G \setminus X \in \mathcal{F}$ for some minor-closed family $\mathcal{F}$ that has bounded-size minimal blocking sets. The goal of the kernelization is then to reduce to an equivalent instance of size $|X|^{\mathcal{O}(1)}$ in polynomial time. Using ideas of the previous kernelizations [4,22], it is quite simple to reduce the *number of connected components* of $G \setminus X$ to $|X|^{\mathcal{O}(1)}$. To obtain a polynomial kernel, the challenge is therefore to bound the size of each such component C of $G \setminus X$ to $|X|^{\mathcal{O}(1)}$, so that the overall instance size becomes polynomial in $|X|$. However, the non-existence of large minimal blocking sets does not seem to offer any handle for reducing the size of individual components of $G \setminus X$. The route to the kernelization therefore goes via the detour of bridge-depth. We prove the following relation between the sizes of minimal blocking sets and bridge-depth.

▶ **Theorem 1.2.** *Let $\mathcal{F}$ be a minor-closed family of graphs. Then $\mathcal{F}$ has bounded bridge-depth if and only if the size of minimal blocking sets of graphs in $\mathcal{F}$ is bounded.*

Using this equivalence, we can exploit the fact that all minimal blocking sets of $\mathcal{F}$ are of bounded size, through the fact that the bridge-depth of $G \setminus X \in \mathcal{F}$ is small. This means that there is a bounded-depth elimination process to reduce $G \setminus X$ to the empty graph. We use this bounded-depth process in a technical kernelization algorithm following a recursive scheme, inspired by the earlier kernelization for the parameterization by distance to bounded tree-depth [4].

Let us now discuss the ideas behind the equivalence of Theorem 1.2. We prove that the bridge-depth of graphs in a minor-closed family $\mathcal{F}$ is upper-bounded in terms of the maximum size of minimal blocking sets for graphs in $\mathcal{F}$, by exploiting the Erdős-Pósa property in an interesting way. We analyze an elementary graph structure called *necklace of length t*, which is essentially the multigraph formed by a path of t double-edges. If a simple graph $G \in \mathcal{F}$ contains a necklace of length t as a minor, then there is a minor G' of G (which therefore also belongs to $\mathcal{F}$) that has a minimal blocking set of size $\Omega(t)$. Hence to show that bridge-depth is upper-bounded in terms of the size of minimal blocking sets of graphs in $\mathcal{F}$, it suffices to show that bridge-depth is upper-bounded by the maximum length of a necklace minor of graphs in $\mathcal{F}$. Since the definition of bridge-depth allows for the contraction of all bridges in a single step, it suffices to consider bridgeless graphs. Then we argue that in a bridgeless graph, any pair of maximum-length necklace minor models intersects at a vertex (cf. Lemma 4.6). By the Erdős-Pósa property, this implies that there is a constant-size vertex set that hits all maximum-length necklace minor models, and whose removal therefore strictly decreases the maximum length of a necklace minor. If the length of necklace minor models is bounded, then after a bounded number of steps of this process (interleaved with contracting all bridges) we reduce the maximum length of necklace minor models to zero, which is equivalent to breaking all cycles of the graph. At that point, the bridge-depth is one by definition, and we have obtained the desired upper-bound on the bridge-depth in terms of the length of the longest necklace minor, and therefore blocking set size.

For the other direction of Theorem 1.2, we prove (cf. Theorem 5.4) the tight bound that a minimal blocking set in a graph G has size at most $2^{\mathsf{bd}(G)}$. We use induction to prove this statement, together with an analysis of the structure of a tree of bridges whose removal

decreases the bridge-depth. The fact that bipartite graphs have minimal blocking sets of size at most two, allows for an elegant induction step.

Related work. In a recent paper, Hols, Kratsch, and Pieterse [19] also analyze the role of blocking sets in the existence of polynomial kernels for structural parameterizations of Vertex Cover. Note that our paper is independent from, and orthogonal to [19]: we consider the setting of *deterministic* kernelization algorithms for parameterizations to *minor-closed* families $\mathcal{F}$, and obtain an exact characterization of which $\mathcal{F}$ allow for a polynomial kernelization. Hols et al. [19] consider hereditary families $\mathcal{F}$ and give kernelizations for several such parameterizations, without arriving at a complete characterization. Some of the randomized kernelizations they provide do not fit into our framework, but all the deterministic kernelizations they present are captured by Theorem 1.1. Another contribution of [19] is to prove that there is a class $\mathcal{F}$ with minimal blocking sets of size one where Vertex Cover cannot be solved in polynomial time. In particular, there is no polynomial kernel parameterized by the distance to this family $\mathcal{F}$, and thus bounded minimal blocking set size is not sufficient to get a polynomial kernel. This implies that our minor-closed assumption of Theorem 1.1 cannot be dropped.

We refer to the survey by Fellows et al. [12] for an overview of classic results and new research lines concerning kernelization for Vertex Cover. Additional relevant work includes the work by Kratsch [24] on a randomized polynomial kernel for a parameterization related to the difference between twice the cost of the linear-programming relaxation of Vertex Cover and the size of a maximum matching.

Organization. Preliminaries on graphs and complexity are presented in Section 2. Section 3 introduces bridge-depth and its properties. In Section 4 we prove one direction of Theorem 1.2, showing that large bridge-depth implies the existence of large minimal blocking sets. In Section 5 we handle the other direction, proving a tight upper-bound on the size of minimal blocking sets in terms of the bridge-depth. We present the kernelization algorithm exploiting bridge-depth in Section 6, and we conclude the article in Section 7.

2 Preliminaries

Graphs. We use standard graph-theoretic notation, and we refer the reader to Diestel [10] for any undefined terms. All graphs we consider are finite and undirected. Graphs are simple, unless specifically stated otherwise, such as when dealing with necklaces (see Definition 4.1). A graph G has vertex set $V(G)$ and edge set $E(G)$. Given a graph G and a subset $S \subseteq V(G)$, we say that S is *connected* if $G[S]$ is connected, and we use the shorthand $G \setminus S$ to denote $G[V(G) \setminus S]$. For a single vertex $v \in V(G)$, we use $G \setminus v$ as a shorthand for $G \setminus \{v\}$. Similarly, for a set of edges $T \subseteq E(G)$ we denote by $G \setminus T$ the graph on vertex set $V(G)$ with edge set $E(G) \setminus T$. A cycle on three vertices is called a *triangle*. For two positive integers i, j with $i \leq j$, we denote by $[i, j]$ the set of all integers ℓ such that $i \leq \ell \leq j$, and by $[i]$ the set $[1, i]$. Given $v \in V(G)$, we denote $N_G(v) = \{u \mid \{u, v\} \in E(G)\}$, $d_G(v) = |N_G(v)|$ and given $X \subseteq V(G)$, we denote $N_G(X) = \bigcup_{v \in X} N_G(v) \setminus X$. Given $X, Y \subseteq V(G)$, we denote by $N_G^Y(X) = N_G(X) \cap Y$. We may omit the subscript G when it is clear from the context. For distinct vertices u and v of a graph G, the graph G' obtained by *identifying* u and v is defined by removing vertices u and v from G, adding a new vertex uv with $N_{G'}(uv) = N_G(\{u, v\})$, and keeping the other vertices and edges unchanged. Given two adjacent vertices u and v, we define the *contraction* of the edge $\{u, v\}$ as the identification of u and v.

Given a graph G, we denote by $\alpha(G)$ the size of a maximum independent set in G, by $\#\text{CC}(G)$ the number of connected components of G, by $\mathsf{diam}(G)$ the diameter of G, and by $\Delta(G)$ the maximum degree of G. Given an independent set $S \subseteq V(G)$ and a set $V' \subseteq V(G)$, we denote $S_{V'} = S \cap V'$. Given a graph G and a set $S \subseteq V(G)$, we say that S is a *blocking set* in G if $\alpha(G \setminus S) < \alpha(G)$. The maximum size of an inclusion-wise minimal blocking set of a graph G is denoted by $\mathsf{mbs}(G)$.

A graph H is a minor of graph G if H can be obtained from G by a sequence of edge deletions, edge contractions, and removals of isolated vertices. Let us also recall the definition of minor in the context of multigraphs. Let H be a loopless multigraph. An *H-model* M in a simple graph G is a collection $\{S_x^M \mid x \in V(H)\}$ of pairwise disjoint subsets of $V(G)$ such that $G[S_x^M]$ is connected for every $x \in V(H)$, and such that for every pair of distinct vertices x, y of H, the quantity $|\{\{u, v\} \in E(G) \mid u \in S_x^M, v \in S_y^M\}|$ is at least the number of edges in H between x and y. The vertex set $V(M)$ of M is the union of the vertex sets of the subgraphs in the collection. We say that a graph G contains a loopless multigraph H as a *minor* if G has an H-model.

For the following definitions, we refer the reader to [29] for more details and we only recall here some basic notations and facts. The *tree-depth* of a graph G, denoted by $\mathsf{td}(G)$, is defined recursively. The empty graph without vertices has tree-depth zero. The tree-depth of a disconnected graph is the maximum tree-depth of its connected components. Finally, if G is a nonempty connected graph then $\mathsf{td}(G) = 1 + \min_{v \in V(G)} \mathsf{td}(G \setminus v)$. Equivalent definitions exist in terms of the minimum height of a rooted forest whose closure is a supergraph of G. The tree-width of G is denoted $\mathsf{tw}(G)$ (cf. [2]).

Given a graph family $\mathcal{F}$, an *$\mathcal{F}$-modulator* in a graph G is a subset of vertices $X \subseteq V(G)$ such that $G \setminus X \in \mathcal{F}$. We denote by $\mathsf{dist\text{-}to\text{-}}\mathcal{F}(G)$ the size of a smallest $\mathcal{F}$-modulator in G. For a graph measure f that associates an integer with each graph, and an integer c, a *c-f-modulator* is a modulator to $\mathcal{F}_c^{\mathsf{f}} := \{G \mid \mathsf{f}(G) \leq c\}$. We denote by $c\text{-}\mathsf{f}\text{-}\mathsf{mod}(G) := \mathsf{dist\text{-}to\text{-}}\mathcal{F}_c^{\mathsf{f}}(G)$, that is, the size of a smallest c-f-modulator of G. Typical measures f that we consider here are tree-width, tree-depth, and bridge-depth. Notice that $0\text{-}\mathsf{tw}\text{-}\mathsf{mod}(G)$ corresponds to the minimum size of a vertex cover of G, and $1\text{-}\mathsf{tw}\text{-}\mathsf{mod}(G)$ corresponds to the minimum size of a feedback vertex set of G. Finally, IS (resp. VC) denotes the Maximum Independent Set (resp. Minimum Vertex Cover) problem.

Parameterized complexity. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, for some finite alphabet Σ. For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, the value k is called the *parameter*. For a computable function $g \colon \mathbb{N} \to \mathbb{N}$, a *kernelization algorithm* (or simply a *kernel*) for a parameterized problem L of *size* g is an algorithm A that given any instance (x, k) of L, runs in polynomial time and returns an instance (x', k') such that $(x, k) \in L \Leftrightarrow (x', k') \in L$ with $|x'|, k' \leq g(k)$. Consult [7, 11, 14, 16, 31] for background on parameterized complexity.

3 An introduction to bridge-depth

Let G be a graph. An edge $e \in E(G)$ is a *bridge* if its removal increases the number of connected components of G. We define G_{CB} as the simple graph obtained from G by contracting all bridges of G (the order does not matter.) Observe that, as contracting an edge cannot create a new bridge, G_{CB} has no bridges, implying that $(G_{\text{CB}})_{\text{CB}} = G_{\text{CB}}$. Given a subgraph T of a graph G, we say that T is a *tree of bridges* if T is a tree and, for every $e \in E(T)$, e is a bridge in G. Note that a single vertex is, by definition, a tree of bridges. Note also that with any vertex $v \in V(G_{\text{CB}})$ we can associate, in a bijective way, an inclusion-wise

maximal tree of bridges T_v of G. The set $\{T_v \mid v \in V(G_{\text{CB}})\}$ is a minor model of G_{CB} in G (a G_{CB}-model, from now on). For any $u, v \in V(G_{\text{CB}})$ such that $\{u, v\} \in E(G_{\text{CB}})$, there is exactly one edge $\{u', v'\} \in E(G)$ with $u' \in T_u$ and $v' \in T_v$. The latter claim can be easily verified by supposing that there are two such edges, implying that some edge in T_u or T_v is involved in a cycle, which contradicts the fact that all the edges in T_u and T_v are bridges.

▶ **Definition 3.1.** *The* bridge-depth $\mathsf{bd}(G)$ *of a graph G is recursively defined as follows:*

- *If G is the empty graph without any vertices, then* $\mathsf{bd}(G) = 0$.
- *If G has $\ell > 1$ connected components $\{G_i \mid i \in [\ell]\}$, then* $\mathsf{bd}(G) = \max_{i \in [\ell]} \mathsf{bd}(G_i)$.
- *If G is connected, then* $\mathsf{bd}(G) = 1 + \min_{v \in V(G_{CB})} \mathsf{bd}(G_{CB} \setminus v)$.

Informally, bd behaves like tree-depth except that at each step of the recursive definition we are allowed to delete trees of bridges instead of just single vertices, as proved in Item 4 of the following proposition. The following properties of bridge-depth follow from the definitions in an elementary way, often exploiting the fact that if e is a bridge in G, then e is also a bridge in any minor of G that still contains e.

▶ **Proposition 3.2.** *For any graph G the following claims hold:*

1. $\mathsf{bd}(G) = 1$ *if and only if G is a forest with at least one vertex.*
2. $\mathsf{bd}(G_{CB}) = \mathsf{bd}(G)$.
3. *The parameter* bd *is minor-closed: if G' is a minor of G then* $\mathsf{bd}(G') \leq \mathsf{bd}(G)$.
4. *If G is connected, then* $\mathsf{bd}(G) = 1 + \min_T \mathsf{bd}(G \setminus V(T))$, *where the minimum is taken over all trees of bridges T of G.*
5. *For any $X \subseteq V(G)$, we have* $\mathsf{bd}(G) \leq |X| + \mathsf{bd}(G \setminus X)$.
6. $\mathsf{tw}(G) \leq \mathsf{bd}(G)$.

Proof. The first item follows easily from the definition, while the second one uses that $(G_{\text{CB}})_{\text{CB}} = G_{\text{CB}}$.

Proof of 3: We prove the claim by induction on $|V(G)| + |E(G)|$. Suppose that G has multiple connected components $\{G_i \mid i \in [\#\text{CC}(G)]\}$, and let $\{G'_i \mid i \in [\#\text{CC}(G')]\}$ be the connected components of the minor G' of G. Then each connected component G'_j is a minor of some component G_i of G on fewer than $|V(G)|$ vertices, which gives $\mathsf{bd}(G'_j) \leq \mathsf{bd}(G_i)$ by induction. Hence we have $\mathsf{bd}(G') = \max_{j \in [\#\text{CC}(G')]} \mathsf{bd}(G'_j) \leq \max_{i \in [\#\text{CC}(G)]} \mathsf{bd}(G_i) = \mathsf{bd}(G)$.

We now deal with the case that G is connected. In general, if some graph G^* is a minor of G, then G^* is a minor of a graph G' obtained from G by removing an edge, contracting an edge, or removing an isolated vertex. Since G is assumed to be connected, the third case cannot occur here. Then by induction, we have $\mathsf{bd}(G^*) \leq \mathsf{bd}(G')$, so it suffices to prove that $\mathsf{bd}(G') \leq \mathsf{bd}(G)$ for any graph G' obtained by removing or contracting an edge. Let us first prove that if G'_{CB} is a minor of G_{CB}, then $\mathsf{bd}(G') \leq \mathsf{bd}(G)$. Indeed, let $v^* \in V(G_{\text{CB}})$ such that $\mathsf{bd}(G) = 1 + \mathsf{bd}(G_{\text{CB}} \setminus v^*)$, and consider an arbitrary component G'_i of G'. Note that $(G'_i)_{\text{CB}}$ is a component of $(G')_{\text{CB}}$, and therefore a minor of G_{CB} by hypothesis. If $(G'_i)_{\text{CB}}$ is a minor of the graph $G_{\text{CB}} \setminus v^*$, then by induction and Item 2 we have $\mathsf{bd}(G'_i) = \mathsf{bd}((G'_i)_{\text{CB}}) \leq \mathsf{bd}(G_{\text{CB}} \setminus v^*) < \mathsf{bd}(G)$. Otherwise, any minor model $\{S_x \mid x \in V((G'_i)_{\text{CB}})\}$ of $(G'_i)_{\text{CB}}$ in G_{CB} contains a branch set S_{x^*} with $v^* \in S_{x^*}$. But then $\mathsf{bd}((G'_i)_{\text{CB}}) \leq 1 + \mathsf{bd}((G'_i)_{\text{CB}} \setminus x^*)$ by definition, and $(G'_i)_{\text{CB}} \setminus x^*$ is a minor of $G_{\text{CB}} \setminus v^*$, and therefore has bridge-depth at most $\mathsf{bd}(G) - 1$, so that $\mathsf{bd}((G'_i)_{\text{CB}}) \leq \mathsf{bd}(G)$. Hence for each component G'_i of G' we have $\mathsf{bd}(G'_i) = \mathsf{bd}((G'_i)_{\text{CB}}) \leq \mathsf{bd}(G)$, implying $\mathsf{bd}(G') \leq \mathsf{bd}(G)$.

Thus, it only remains to prove that G'_{CB} is a minor of G_{CB}. Let us first assume that G' is obtained from G by removing an edge e. Let $\{T_v \mid v \in V(G_{\text{CB}})\}$ be the G_{CB}-model in G

given by the trees of bridges. If e is not a bridge, then e is an edge between T_u and T_v for some vertices $u, v \in V(G_{\text{CB}})$. To obtain G'_{CB} as a minor, we start from G_{CB}, remove edge $\{u, v\}$, and for any edge e' between $T_{u'}$ and $T_{v'}$ (for any $u', v' \in V(G_{\text{CB}})$) that has become a bridge in G' because of the removal of e, we contract $\{u', v'\}$. This implies that G'_{CB} is a minor of G_{CB}. Otherwise, if e is a bridge, then there exists $u \in V(G_{\text{CB}})$ such that $e \in E(T_u)$, and G' has two connected components G'_1 and G'_2. To obtain $(G'_i)_{\text{CB}}$ as a minor, for $i \in [2]$, we start from G_{CB} and remove any vertex v such that $T_v \cap V(G'_i) = \emptyset$ (notice that u appears both in $(G'_1)_{\text{CB}}$ and $(G'_2)_{\text{CB}}$). Thus, both $(G'_1)_{\text{CB}}$ and $(G'_2)_{\text{CB}}$ are minors of G_{CB}, hence G'_{CB} as well. The case where G' is obtained from G by contracting an edge e can be proved using similar but simpler arguments. Indeed, if e is a bridge in G, then we have that $G'_{\text{CB}} = G_{\text{CB}}$, and if it is not, it suffices to contract in G_{CB} the edge $\{u, v\}$ with $u, v \in V(G_{\text{CB}})$ such that e is an edge between T_u and T_v.

Proof of 4: Let $v \in V(G_{\text{CB}})$ and T_v be its associated tree of bridges in G. Observe first that we may have $(G \setminus V(T_v))_{\text{CB}} \neq G_{\text{CB}} \setminus v$. Indeed, if for example we consider G composed of two vertex-disjoint triangles $\{a, b, c\}$, $\{a', b', c'\}$ and an edge $e = \{a, a'\}$, and if we consider $T_v = \{e\}$, then $G_{\text{CB}} \setminus v$ is composed of two disjoint edges, whereas $(G \setminus V(T_v))_{\text{CB}}$ is composed of two isolated vertices. However, it is easy to verify that $(G \setminus V(T_v))_{\text{CB}} = (G_{\text{CB}} \setminus v)_{\text{CB}}$. Let us now prove that $\min_T \mathsf{bd}(G \setminus V(T)) \leq \min_{v \in V(G_{\text{CB}})} \mathsf{bd}(G_{\text{CB}} \setminus v)$. Let v^* be a vertex minimizing $\mathsf{bd}(G_{\text{CB}} \setminus v)$. We have $\min_T \mathsf{bd}(G \setminus V(T)) \leq \mathsf{bd}(G \setminus V(T_{v^*})) = \mathsf{bd}((G \setminus V(T_{v^*})_{\text{CB}})$ using Item 2 in the last equality, and $\mathsf{bd}((G \setminus V(T_{v^*})_{\text{CB}}) = \mathsf{bd}((G_{\text{CB}} \setminus v^*)_{\text{CB}}) = \mathsf{bd}(G_{\text{CB}} \setminus v^*)$ using again Item 2.

For the other inequality, let T^0 be a tree of bridges that minimizes $\mathsf{bd}(G \setminus V(T))$. If T^0 is not inclusion-wise maximal, let T^* be any inclusion-wise maximal tree of bridges containing T^0. Note that as $G \setminus V(T^*)$ is a subgraph of $G \setminus V(T^0)$, by Item 3 we get that $\mathsf{bd}(G \setminus V(T^*)) \leq \mathsf{bd}(G \setminus V(T^0))$, implying that T^* also minimizes $\mathsf{bd}(G \setminus V(T))$. Let $v^* \in V(G_{\text{CB}})$ such that $T_{v^*} = T^*$. We have $\min_{v \in V(G_{\text{CB}})} \mathsf{bd}(G_{\text{CB}} \setminus v) \leq \mathsf{bd}(G_{\text{CB}} \setminus v^*) = \mathsf{bd}((G \setminus T_{v^*})_{\text{CB}}) = \mathsf{bd}(G \setminus T_{v^*})$.

Proof of 5. We use induction on $|X|$, the base case $X = \emptyset$ being trivial. For the induction step, pick an arbitrary $v \in X$, let $X' := X \setminus \{v\}$, and $G' := G \setminus X'$. By induction we have $\mathsf{bd}(G) \leq |X'| + \mathsf{bd}(G')$. Let G'_i be the connected component of G' containing v. Using v as a singleton tree of bridges in G'_i, Item 4 shows that $\mathsf{bd}(G'_i) \leq 1 + \mathsf{bd}(G'_i \setminus v) \leq 1 + \mathsf{bd}(G' \setminus v)$. Since all other components G'_j of G' also occur as components of $G' \setminus v$, it follows that $\mathsf{bd}(G'_j) \leq \mathsf{bd}(G' \setminus v)$, implying $\mathsf{bd}(G') \leq 1 + \mathsf{bd}(G' \setminus v) = 1 + \mathsf{bd}(G \setminus X)$ since $G' \setminus v = G \setminus X$. Hence $\mathsf{bd}(G) \leq |X'| + 1 + \mathsf{bd}(G \setminus X)$.

Proof of 6: We use induction on $|V(G)|$; the base case follows directly from the definitions. It is well-known (cf. [2, Lemma 6]) that the tree-width of G is the maximum tree-width of its biconnected components. Hence it suffices to prove that for an arbitrary biconnected component G' of G, we have $\mathsf{tw}(G') \leq \mathsf{bd}(G')$. If G' consists of a single edge, then $\mathsf{tw}(G') = \mathsf{bd}(G') = 1$. Otherwise, G' is a connected bridgeless graph. This implies $(G')_{\text{CB}} = G'$, so by Definition 3.1 there is a vertex $v \in V(G')$ such that $\mathsf{bd}(G') = 1 + \mathsf{bd}(G' \setminus v)$. Since G' is a minor of G, we have $\mathsf{bd}(G') \leq \mathsf{bd}(G)$ by Item 3. By induction, the tree-width of $G' \setminus v$ is at most $\mathsf{bd}(G' \setminus v) \leq \mathsf{bd}(G) - 1$. Adding vertex v to all bags of a tree decomposition of this width, gives a valid tree decomposition of G' of width at most $\mathsf{bd}(G' \setminus v) + 1 \leq \mathsf{bd}(G')$. Hence $\mathsf{tw}(G') \leq \mathsf{bd}(G')$ for all biconnected components of G. ◀

A $(c+1) \times (c+1)$-grid is a planar graph of tree-width exactly $c+1$ [2, Cor. 89], which implies by Item 6 of Proposition 3.2 that its bridge-depth is larger than c. This gives the following consequence of Proposition 3.2, which will be useful when invoking algorithmic meta-theorems.

▶ **Observation 3.3.** *For each $c \in \mathbb{N}$, the graphs of bridge-depth at most c form a minor-closed family that excludes a planar graph. By the Graph Minor Theorem [35], there is a finite set of forbidden minors $\mathcal{H}_c$ such that $\mathsf{bd}(G) \leq c$ if and only if G excludes all graphs of $\mathcal{H}_c$ as a minor. The set $\mathcal{H}_c$ contains a planar graph, since some planar graphs have bridge-depth $> c$.*

Observation 3.3, together with known results on minor testing, imply the following.

▶ **Proposition 3.4** (Follows from [1, Thm. 7.1])**.** *For each constant $c \in \mathbb{N}$, there is a linear-time algorithm to test whether the bridge-depth of a given graph G is at most c.*

Fomin et al. [15, Thm. 1.3] gave a generic approximation algorithm for finding a small vertex set that hits forbidden minors from a finite forbidden set containing a planar graph. By Observation 3.3, deleting vertices to obtain a graph of bounded bridge-depth fits into their framework.

▶ **Proposition 3.5** (Follows from [15, Thm. 1.3])**.** *For each fixed $c \in \mathbb{N}$ there is a polynomial-time algorithm that, given a graph G, outputs a set $X \subseteq V(G)$ such that $\mathsf{bd}(G \setminus X) \leq c$ and $|X| \leq \mathcal{O}(|X_{\text{opt}}| \log^{2/3} |X_{\text{opt}}|)$, where $|X_{\text{opt}}|$ is the minimum size of such a set.*

The following concept will be crucial to facilitate a recursive approach for reducing graphs of bounded bridge-depth.

▶ **Definition 3.6.** *A* lowering tree *T of a graph G is a tree of bridges (possibly consisting of a single vertex and no bridges) such that $\mathsf{bd}(G \setminus V(T)) = \mathsf{bd}(G) - 1$.*

Item 4 of Proposition 3.2 implies that any connected graph G has a lowering tree.

▶ **Proposition 3.7.** *For each fixed $c \in \mathbb{N}$ there is an algorithm that, given a connected graph G on n vertices of bridge-depth c, computes a lowering tree in $\mathcal{O}(n^2)$ time.*

Proof. Given G, we compute its decomposition into biconnected components, which can be done in linear time taking into account that having bounded bridge-depth implies a linear number of edges [20]. From this decomposition, it is straightforward to identify the inclusion-maximal trees of bridges in G. For each tree of bridges T in G, we can test whether $\mathsf{bd}(G \setminus V(T)) < c = \mathsf{bd}(G)$ in linear time using Proposition 3.4, and we output T if this is the case. By Proposition 3.2, such a tree T exists. Since G is decomposed into at most n trees of bridges, and we need a linear-time computation for each T, this results in an $\mathcal{O}(n^2)$-time algorithm. ◀

4 Bounded minimal blocking sets imply bounded bridge-depth

The goal of this section is to prove one direction of Theorem 1.2, showing that if $\mathcal{F}$ has bounded-size minimal blocking sets, then $\mathcal{F}$ has bounded bridge-depth. As explained in Section 1, we prove this via the intermediate structure of *necklace minors* and show that the bridge-depth of a graph G can be upper-bounded in terms of the longest necklace contained in it as a minor.

This result can be seen as an analog to the fact that the tree-depth of a graph can be bounded in terms of the length of the longest simple path it contains (as a subgraph or as a minor, which is equivalent for paths). A classical proof of this fact (see [29]) is to consider a depth-first search tree of G, bounding the tree-depth of G by the depth of this tree. However, it does not seem immediate to find a similar bound for bridge-depth.

We therefore follow another approach, inspired by the following alternative proof that the tree-depth is upper-bounded by the length of the longest path (which gives a worse

bound). Observe that in a connected graph G, any two longest paths intersect at a vertex. (If they did not, one could combine them to make an even longer path.) Given a connected graph G whose longest path has t vertices, we can bound its tree-depth by $f(t) := \sum_{i=1}^{t} i$ as follows. Let P be a longest path in G. Then the longest path in $G \setminus V(P)$ has strictly fewer than t vertices, and by induction the tree-depth of $G \setminus V(P)$ is at most $f(t-1)$. From the definition of tree-depth, it follows that the tree-depth of G is at most $|V(P)| = t$ larger than that of $G \setminus V(P)$, so the tree-depth of G is at most $f(t)$.

In the case of bridge-depth, where paths are replaced with necklaces contained as minors, we cannot afford to remove the entire set of vertices of the corresponding model of a longest necklace, as the size of this set cannot be bounded in terms of the length t of the necklace. To overcome this problem, we will prove in Lemma 4.6, similarly to the case of paths, that there cannot be two vertex-disjoint longest necklaces. Then we resort to the Erdős-Pósa property, which gives us a set of vertices of size $f(t)$ whose removal decreases the maximum length of a longest necklace. We now formalize these ideas.

▶ **Definition 4.1.** *For $t \in \mathbb{N}$, the* necklace of length t*, denoted by N_t, is the multigraph having $t+1$ vertices $\{v_i \mid i \in [t+1]\}$ and two parallel edges between v_i and v_{i+1} for $i \in [t]$.*

▶ **Observation 4.2.** *A simple graph G contains N_t as a minor if and only if G contains $t+1$ vertex-disjoint sets $S_i \subseteq V(G)$ such that each S_i is connected and, for $i \in [t]$, there are at least two edges between S_i and S_{i+1}.*

▶ **Definition 4.3.** *The* necklace-minor length *of a graph G, denoted by $\mathsf{nm}(G)$, is the largest length of a necklace contained in G as a minor, or zero if G contains no such minor.*

We need to introduce the Erdős-Pósa property for packing and covering minor models. Let $\mathcal{F}$ be a finite collection of simple graphs. An *$\mathcal{F}$-model* is an H-model for some $H \in \mathcal{F}$. Two $\mathcal{F}$-models M_1 and M_2 are *disjoint* if $V(M_1) \cap V(M_2) = \emptyset$. Let $\nu_{\mathcal{F}}(G)$ be the maximum cardinality of a packing of pairwise disjoint $\mathcal{F}$-models in G, and let $\tau_{\mathcal{F}}(G)$ be the minimum size of a subset $X \subseteq V(G)$ such that $G \setminus X$ has no $\mathcal{F}$-model. Clearly, $\nu_{\mathcal{F}}(G) \leq \tau_{\mathcal{F}}(G)$. We say that the *Erdős-Pósa property* holds for $\mathcal{F}$-models if there exists a bounding function $f\colon \mathbb{N} \to \mathbb{N}$ such that, for every graph G, $\tau_{\mathcal{F}}(G) \leq f(\nu_{\mathcal{F}}(G))$.

In the case where $\mathcal{F} = \{H\}$ contains a single connected graph H, Robertson and Seymour [34] proved the following result.

▶ **Theorem 4.4** (Robertson and Seymour [34])**.** *Let H be a connected graph. The Erdős-Pósa property holds for H-models if and only if H is planar.*

It is worth mentioning that a tight bounding function when H is planar has been recently obtained by van Batenburg et al. [36]. Theorem 4.4 easily implies the following corollary.

▶ **Corollary 4.5.** *For every $t \geq 1$, the Erdős-Pósa property holds for N_t-models.*

Proof. For $t \geq 1$, let $\mathcal{F}_t$ be the set containing all minor-minimal *simple* graphs that contain the necklace N_t as a minor. By definition, a simple graph G contains an N_t-model if and only if it contains an $\mathcal{F}_t$-model. Clearly, all the graphs in $\mathcal{F}_t$ are connected and planar, and it is easy to see that $|\mathcal{F}_t|$ is bounded by a function of t. For each $F \in \mathcal{F}_t$, by Theorem 4.4 there is a function f_F such that if G does not contain k vertex-disjoint models of F, then all the F-models of G can be hit by at most $f_F(k)$ vertices. This implies that if G does not contain k models of any graph in $\mathcal{F}_t$, then the union of all hitting sets has size bounded by $\sum_{F \in \mathcal{F}_t} f_F(k)$, and since $\mathcal{F}_t$ is finite this is a valid bounding function for N_t-models. ◀

We denote by f_{N_t} the bounding function for N_t-models given by Corollary 4.5. In a connected bridgeless graph, each pair of maximum-length necklace models intersect at a vertex:

▶ **Lemma 4.6.** *If G is a connected bridgeless simple graph with $\mathsf{nm}(G) = t$, then $\nu_{N_t}(G) = 1$.*

Proof. Suppose for a contradiction that G contains two disjoint models M^1 and M^2 of N_t. For $i \in [t+1]$ and $\ell \in [2]$, let S_i^ℓ be the vertex set of M^ℓ given by Observation 4.2. Note that these $2t+2$ subsets of vertices of G are pairwise disjoint, and that for any $i \in [t]$, there are at least two edges between S_i^ℓ and S_{i+1}^ℓ. Since G is bridgeless and connected, it is 2-edge-connected and by Menger's theorem [10, § 3.3] G contains two edge-disjoint paths between any pair of vertices. Pick two arbitrary vertices $x_1 \in M^1, x_2 \in M^2$, and let P^1, P^2 be two edge-disjoint paths between them. Consider the subpath Q^ℓ of P^ℓ between the last vertex of M^1 that is visited, until the first vertex of M^2. Let $Q^\ell = (v_1^\ell, \ldots, v_{q_\ell}^\ell)$ where $v_1^\ell \in M^1$ and $v_{q_\ell}^\ell \in M^2$. Let a_ℓ such that $v_1^\ell \in S_{a_\ell}^1$ and b_ℓ such that $v_{q_\ell}^\ell \in S_{b_\ell}^2$.

Let us first show that if t is odd, then we can use Q^1 to find an $N_{t'}$-model M' for some $t' > t$ by "gluing" M^1 and M^2, leading to a contradiction. Let $S = S_{a_1}^1 \cup V(Q^1) \cup S_{b_1}^2$. If $a_1 > \frac{t+1}{2}$ define $A = \{S_1^1, \ldots, S_{a_1-1}^1\}$, and otherwise define $A = \{S_{a_1+1}^1, \ldots, S_{t+1}^1\}$. Similarly, if $b_1 > \frac{t+1}{2}$ define $B = \{S_1^2, \ldots, S_{b_1-1}^2\}$, and otherwise define $B = \{S_{b_1+1}^2, \ldots, S_{t+1}^2\}$. Note that the sets in A, S, B are pairwise disjoint. Since t is odd, it can be easily checked that $M' = A \cup \{S\} \cup B$ is an $N_{t'}$-model in G for some $t' > t$; see Figure 1(a) for an illustration.

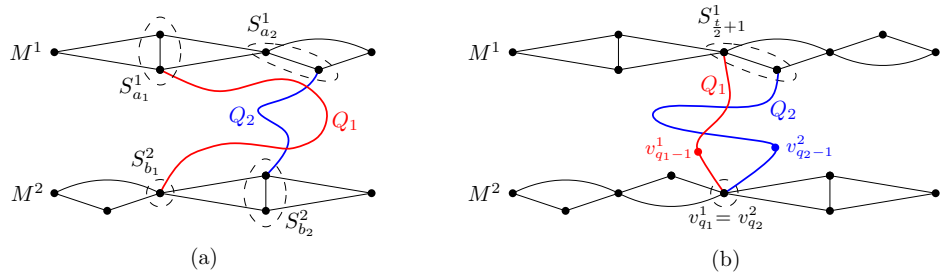

Figure 1 (a) Example with $t = 3$ and $a_1 = b_1 = 2$. (b) Example with $t = 4$.

Let us now consider the case where t is even. Note first that if there exists $\ell \in [2]$ such that $a_\ell \neq \frac{t}{2} + 1$ or $b_\ell \neq \frac{t}{2} + 1$, then we can use Q^ℓ to find an $N_{t'}$-model for some $t' > t$ as in the previous case. Hence, it only remains to consider the case where $a_1 = b_1 = a_2 = b_2 = \frac{t}{2} + 1$, meaning that Q^1 and Q^2 are two edge-disjoint paths, both between $S_{\frac{t}{2}+1}^1$ and $S_{\frac{t}{2}+1}^2$. Let $A = \{S_1^1, \ldots, S_{\frac{t}{2}}^1\}$, $B = \{S_1^2, \ldots, S_{\frac{t}{2}}^2\}$, and $S = S_{a_1}^1 \cup (V(Q^1) \setminus \{v_{q_1}^1\}) \cup (V(Q^2) \setminus \{v_{q_2}^2\})$. We claim that $M' = A \cup \{S, S_{\frac{t}{2}+1}^2\} \cup B$ is an N_{t+1}-model. Indeed, note in particular there are two edges between S and $S_{\frac{t}{2}+1}^2$ as we cannot have $v_{q_1-1}^1 = v_{q_2-1}^2$ and $v_{q_1}^1 = v_{q_2}^2$ because Q^1 and Q^2 are edge-disjoint and G is a simple graph; see Figure 1(b) for an illustration. ◀

By combining Corollary 4.5 with Lemma 4.6 we easily get the following corollary.

▶ **Corollary 4.7.** *Let G be a connected bridgeless graph and $t = \mathsf{nm}(G)$. Then G contains a set of vertices X with $|X| \leq f_{N_t}(1)$ such that $\mathsf{nm}(G \setminus X) < \mathsf{nm}(G)$, where $f_{N_t} : \mathbb{N} \to \mathbb{N}$ is the bounding function given by Corollary 4.5.*

Proof. By Lemma 4.6, it follows that $\nu_{N_t}(G) = 1$, and therefore by Corollary 4.5

$$\tau_{N_t}(G) \;\leq\; f_{N_t}(\nu_{N_t}(G)) \;=\; f_{N_t}(1).$$

Thus, there exists a set $X \subseteq V(G)$ with $|X| \leq f_{N_t}(1)$ such that $G \setminus X$ has no N_t-model, implying that $\mathsf{nm}(G \setminus X) < t$. ◀

We are finally in position to prove the following theorem.

▶ **Theorem 4.8.** *There is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that $\mathsf{bd}(G) \leq f(\mathsf{nm}(G))$ for all graphs G.*

Proof. We prove the statement by induction on $\mathsf{nm}(G)$, for the function f defined by $f(t) := 1 + \sum_{i=1}^{t} f_{N_i}(1)$. If $\mathsf{nm}(G) = 0$, then G is a forest, and by definition of bridge-depth we get $\mathsf{bd}(G) = 1 = f(0)$. Suppose now that $\mathsf{nm}(G) = t$ with $t > 0$.

Consider the case that G is connected. Then G_{CB} is also connected and has no bridge, and thus we can apply Corollary 4.7 and get a set $X \subseteq V(G_{\text{CB}})$ with $|X| \leq f_{N_t}(1)$ such that $\mathsf{nm}(G') < t$, where $G' = G_{\text{CB}} \setminus X$. By Item 5 of Proposition 3.2, we get that $\mathsf{bd}(G) = \mathsf{bd}(G_{\text{CB}}) \leq |X| + \mathsf{bd}(G')$. Let $G'_1, \ldots, G'_\ell$ be the connected components of G'. As $\mathsf{nm}(G') < t$, we get that $\mathsf{nm}(G'_i) < t$ for every $i \in [\ell]$. Then, by induction hypothesis it follows that, for every $i \in [\ell]$, $\mathsf{bd}(G'_i) \leq f(t-1) = 1 + \sum_{i=1}^{t-1} f_{N_i}(1)$. Thus, as $\mathsf{bd}(G') = \max_{i\in[\ell]} \mathsf{bd}(G'_i) \leq 1 + \sum_{i=1}^{t-1} f_{N_i}(1)$, we get that

$$\mathsf{bd}(G) \;\leq\; |X| + \mathsf{bd}(G') \;\leq\; f_{N_t}(1) + 1 + \sum_{i=1}^{t-1} f_{N_i}(1) \;=\; 1 + \sum_{i=1}^{t} f_{N_i}(1) \;=\; f(\mathsf{nm}(G)).$$

Finally, if G is disconnected, let $G_1, \ldots, G_\ell$ be its connected components, and note that $\mathsf{bd}(G) = \max_{i\in[\ell]} \mathsf{bd}(G_i)$. Since for every $i \in [\ell]$ it holds that $\mathsf{nm}(G_i) \leq \mathsf{nm}(G)$, and since the function f is non-decreasing, by applying the above case to each connected component of G we get that

$$\mathsf{bd}(G) \;=\; \max_{i\in[\ell]} \mathsf{bd}(G_i) \;\leq\; \max_{i\in[\ell]} f(\mathsf{nm}(G_i)) \;\leq\; \max_{i\in[\ell]} f(\mathsf{nm}(G)) \;=\; f(\mathsf{nm}(G)). \quad ◀$$

Now that we established a relation between bridge-depth and necklace minors, our next step is to relate necklace minors to blocking sets. For this purpose, we use the known triangle-path gadget.

▶ **Definition 4.9.** *A* triangle-path of length t *is the graph consisting of t vertex-disjoint triangles, with vertex sets $\{\{a_i, b_i, c_i\} \mid i \in [t]\}$, together with the $t-1$ edges $\{\{b_i, a_{i+1}\} \mid i \in [t-1]\}$. The* triangle-path-minor length *of a graph G, denoted by $\mathsf{tpm}(G)$, is the largest length of a triangle-path contained in G as a minor, or zero if no such minor exists.*

A slight variation of this gadget was used by Fomin and Strømme [17, Def. 6]. We observe the following (cf. [17, Obs. 3–5]).

▶ **Observation 4.10.** *Let G be a triangle-path of length $t \geq 2$. Then $\mathsf{mbs}(G) \geq t + 2$, as $\{a_1, c_1\} \cup \{b_t, c_t\} \cup \{c_i \mid i \in [2, t-1]\}$ is a minimal blocking set.*

▶ **Lemma 4.11.** *For any graph G, $\mathsf{tpm}(G) \geq \lfloor \frac{\mathsf{nm}(G)+1}{2} \rfloor$.*

Proof. Let $t = \mathsf{nm}(G)$, and let $\{S_i \mid i \in [t+1]\}$ be an N_t-model in G. Let $i \in [\lfloor \frac{t+1}{2} \rfloor]$ and let $e_1 = \{u_1, v_1\}$ and $e_2 = \{u_2, v_2\}$ be the two edges between S_{2i-1} and S_{2i}, with $u_\ell \in S_{2i-1}$ and $v_\ell \in S_{2i}$. If $u_1 \neq u_2$ then there is a partition A_1, A_2 of S_{2i-1} such that $u_i \in A_i$ and A_i is connected for $i \in [2]$, and we define $L_i = \{A_1, A_2\}$, $R_i = \{S_{2i}\}$. Otherwise, if $u_1 = u_2$, then necessarily $v_1 \neq v_2$, and we define symmetrically $L_i = \{S_{2i-1}\}$ and $R_i = \{A_1, A_2\}$. In both cases we get that $L_i \cup R_i$ is a model of a triangle, and moreover there is an edge between a vertex in R_i and a vertex in L_{i+1} for every $i \in [\lfloor \frac{t+1}{2} \rfloor - 1]$. This implies that $\bigcup_{i\in[\lfloor \frac{t+1}{2} \rfloor]} (L_i \cup R_i)$ is a model of a triangle-path of length $\lfloor \frac{t+1}{2} \rfloor$ in G. ◀

▶ **Corollary 4.12.** *There is a function $g\colon \mathbb{N} \to \mathbb{N}$ such that $\mathsf{bd}(G) \leq g(\mathsf{tpm}(G))$ for all graphs G.*

Proof. By Lemma 4.11, we have that $\mathsf{tpm}(G) \geq \mathsf{nm}(G)/2$. By letting $g(t) := f(2t)$, where f is the function given by Theorem 4.8, we get the desired result. ◀

Note that in the next two results we need the assumption that $\mathcal{F}$ is minor-closed. The following immediate corollary of Corollary 4.12 is critically used in proof of the lower bound given in Theorem 6.24.

▶ **Corollary 4.13.** *Let $\mathcal{F}$ be a minor-closed family of graphs. If $\mathcal{F}$ has unbounded bridge-depth then it contains the family $\mathcal{F}^{\mathsf{tp}}$ of all triangle-paths.*

Using this corollary, we can prove one direction of Theorem 1.2.

▶ **Theorem 4.14.** *Let $\mathcal{F}$ be a minor-closed family of graphs of unbounded bridge-depth. Then there are graphs in $\mathcal{F}$ which have arbitrarily large minimal blocking sets.*

Proof. By Corollary 4.13, $\mathcal{F}$ contains all triangle-paths. Since a triangle-path of length t contains a minimal blocking set of size $t + 2$ by Observation 4.10, the theorem follows. ◀

Theorem 4.14 is phrased for graph families, rather than individual graphs. There is no function h such that $\mathsf{bd}(G) \leq h(\mathsf{mbs}(G))$ for all G: a bipartite grid graph can have arbitrarily large tree-width and therefore bridge-depth, but its minimal blocking sets have size at most two (cf. Lemma 5.2).

5 Bounded bridge-depth implies bounded-size blocking sets

In this section we prove the other direction of Theorem 1.2: minimal blocking sets in a graph G have size at most $2^{\mathsf{bd}(G)}$. We need the following consequence of Kőnig's theorem.

▶ **Lemma 5.1.** *Let G be a bipartite graph and let M be a maximum matching in G. Every maximum independent set of G contains all vertices that are not saturated by M, and exactly one endpoint of each edge in M.*

Proof. Consider a maximum independent set S in G. Then $\overline{S} := V(G) \setminus S$ is a minimum vertex cover of G. By Kőnig's theorem (cf. [10, Thm. 2.1.1]) we have $|\overline{S}| = |M|$. Since $\overline{S}$ is a vertex cover it contains at least one endpoint of each edge of M; since $|\overline{S}| = |M|$ it contains exactly one endpoint of each edge of M, and no other vertices of G. So the complement S contains all vertices that are not saturated by M, and exactly one endpoint of each edge in M. ◀

The next lemma shows that minimal blocking sets in a bipartite graph have at most two vertices. This was known before, see [18, Thm. 14]. Our self-contained proof highlights an additional property of such minimal blocking sets: the two vertices of minimal blocking sets of size two belong to opposite partite sets. This will be crucial later on.

▶ **Lemma 5.2.** *Let G be a bipartite graph with partite sets A and B. If $Y \subseteq V(G)$ is a blocking set in G, then there is a blocking set $Y' \subseteq Y$ in G such that one of the following holds:*
- *$|Y'| = 1$, or*
- *$Y' = \{a, b\}$ for some $a \in A$ and $b \in B$.*

Proof. Let M be a maximum matching in G, let $V(M)$ be the saturated vertices, and let $U := V(G) \setminus V(M)$ be the unsaturated vertices. Let $R_{A \cap U}$ be the vertices that can be reached by a possibly empty M-alternating path from $A \cap U$ (which necessarily starts with

a non-matching edge). Let $R_{B\cap Y}$ be the vertices that can be reached by a possibly empty M-alternating path that starts with a *matching edge* from a vertex of $B \cap Y$. Note that both types of alternating paths move from A to B over non-matching edges, and move from B to A over matching edges.

We first deal with some cases in which we easily obtain a blocking set Y' as desired.

(**Case 1:** $A \cap Y \cap R_{A\cap U} \neq \emptyset$) Let $a \in A \cap Y \cap R_{A\cap U}$. Then $a \in A$ can be reached by an M-alternating path P that starts in an unsaturated vertex in the same partite set, implying that P has even length and ends with a matching edge into a. Hence $M' := M \oplus E(P)$, where $\oplus$ denotes the symmetric difference, is a new maximum matching (which is equal to M if P is empty), and it does not saturate $a \in A \cap Y$. Lemma 5.1 applied to M' implies that all maximum independent sets of G contain a, showing that $Y' := \{a\}$ is a blocking set of size one.

(**Case 2:** $B \cap U \cap R_{B\cap Y} \neq \emptyset$) By definition, some $u \in B \cap U$ can be reached by an M-alternating path P that starts in some vertex $b \in B \cap Y$ that belongs to the same partite set. Similarly as in the previous case, $M' := M \oplus E(P)$ is a new maximum matching (which is equal to M if P is empty) that does not saturate b, so by Lemma 5.1 applied to M' we conclude that $Y' := \{b\}$ is a blocking set of size one.

(**Case 3:** $A \cap Y \cap R_{B\cap Y} \neq \emptyset$) By definition, some $a \in A \cap Y$ is reachable by an M-alternating path P from some $b \in B \cap Y$, and P starts with a matching edge. Since it ends in the other partite set, it ends with a matching edge as well; hence both a and b are saturated. We claim that $Y' := \{a, b\}$ is a blocking set in G, as desired. Let $a = a_1, b_1, \ldots, a_k, b_k = b$ be the vertices on P, so that $\{a_i, b_i\} \in M$ for all $i \in [k]$ and $\{b_i, a_{i+1}\} \in E(G) \setminus M$ for $i \in [k-1]$. By Lemma 5.1, a maximum independent set in G contains one endpoint of each of the edges $\{a_i, b_i\} \in M$. A maximum independent set avoiding a_1 therefore has to contain b_1, preventing it from containing a_2, forcing it to contain b_2, and so on. Hence a maximum independent set avoiding a_1 contains b_k, proving that $Y' := \{a, b\} = \{a_1, b_k\}$ is a blocking set in G.

(**Case 4:** $B \cap U \cap R_{A\cap U} \neq \emptyset$) Then some unsaturated vertex of A can reach an unsaturated vertex of B by an M-alternating path P. But then M is not a maximum matching since $M \oplus E(P)$ is larger; a contradiction. Hence this case cannot occur.

Assume now that none of the cases above hold. We will conclude the proof of the lemma by deriving a contradiction. Let $R := R_{A\cap U} \cup R_{B\cap Y}$. The following will be useful.

▷ Claim 5.3. If $a \in A \cap R$ and $\{a, b\} \in E(G)$, then $b \in B \cap R$.

Proof. By definition, $a \in A \cap R$ implies a is reachable by some M-alternating path P that moves to A over matching edges and moves to B over non-matching edges, such that P starts in a vertex $v \in (A \cup U) \cup (B \cap Y)$. But then b is also reachable by such an alternating path from v: if $\{a, b\} \in M$ then, since P ends at a, edge $\{a, b\}$ must be the last edge of P, so a prefix of P is an M-alternating path reaching b; if $\{a, b\} \notin M$ then appending $\{a, b\}$ to P yields such an M-alternating path. Hence $b \in R$, and $b \in B$ follows since G is bipartite. ◁

Now consider the following set: $S := (A \cap R) \cup (B \setminus R)$.

We will prove that S is a maximum independent set of G disjoint from Y, contradicting the assumption that Y is a blocking set. To see that S is indeed an independent set, consider any vertex from $A \cap S$, which belongs to $A \cap R$. By Claim 5.3 all neighbors of a belong to $B \cap R$, and are therefore not contained in S. Hence S is indeed an independent set. To see that it is maximum, by Lemma 5.1 it suffices to argue it contains all of U and one endpoint of each edge in M.

To see that S contains all vertices of $A \cap U$, note that all such vertices are trivially in $R_{A\cap U}$ and therefore in R, implying their presence in $A \cap R$ and therefore in S. To see that S contains all vertices of $B \cap U$, it suffices to show that $B \cap U \cap R = \emptyset$, which follows from the fact that neither Case 2 nor Case 4 is applicable. Hence S contains all vertices of U.

To see that S contains an endpoint of each edge of M, let $\{a, b\} \in M$ be arbitrary with $a \in A$ and $b \in B$. If $b \notin R$ then clearly $b \in S$, as desired. If $b \in R$, then this is witnessed by an alternating path P that reaches b and ends with a non-matching edge. Extending P with the edge $\{a, b\} \in M$ then shows that $a \in R$, so that $a \in A \cap R$ is an endpoint of the edge contained in S.

Hence S is a maximum independent set in G. Since Case 1 and Case 3 do not apply, it follows that $A \cap R \cap Y = \emptyset$, so that $S \cap A$ contains no vertex from Y. Since all vertices of $B \cap Y$ are trivially in $R_{B\cap Y}$ and therefore in R, it follows that $B \setminus R$ contains no vertex from Y. Hence S is a maximum independent set in G disjoint from Y, contradicting the assumption that Y is a blocking set. ◀

We will use Lemma 5.2 to power the induction step in the proof of the next theorem, which gives the desired upper-bound on the size of minimal blocking sets in terms of bridge-depth. The main idea in the induction step is as follows. For a connected graph G, we consider a tree of bridges T for which $\mathsf{bd}(G \setminus V(T)) < \mathsf{bd}(G)$. We can summarize the relevant ways in which a maximum independent set in G can be composed out of maximum independent sets for the connected components of $G \setminus E(T)$, into a weighted tree T' that is obtained from T by adding a pendant leaf to each vertex. In turn, maximum-weight independent sets in T' correspond to maximum independent sets in a bipartite graph obtained from T' by replacing each vertex by a set of false twins. Applying Lemma 5.2 to this bipartite graph points to two vertices that form a blocking set. We can translate this back into two components of $G \setminus E(T)$ which are sufficient for constructing a blocking set in G, and apply induction using the fact that $\mathsf{bd}(G \setminus V(T)) < \mathsf{bd}(G)$.

▶ **Theorem 5.4.** *Let G be a graph and $Y_G \subseteq V(G)$ a blocking set in G. There is a blocking set $Y'_G \subseteq Y_G$ in G of size at most $2^{\mathsf{bd}(G)}$.*

Proof. We use induction on $\mathsf{bd}(G) + |V(G)|$. If $\mathsf{bd}(G) = 1$ then G is bipartite (it is even a forest, by Proposition 3.2) and therefore the claim follows from Lemma 5.2. For the induction step, assume $\mathsf{bd}(G) > 1$.

If G is disconnected, then since a maximum independent set in G contains a maximum independent set from each connected component, there is a connected component C such that $Y_G \cap V(C)$ is a blocking set of C. By induction on C and $Y_G \cap V(C)$ we obtain a blocking set $Y'_G \subseteq Y_G$ for C of size at most $2^{\mathsf{bd}(C)} \leq 2^{\mathsf{bd}(G)}$, and Y'_G is also a blocking set in G.

It remains to deal with the case that G is connected. If there is a vertex $v \in V(G)$ such that no maximum independent set in G contains v, then any blocking set for $G \setminus v$ is a blocking set in G. Hence we may invoke induction on $G \setminus v$ and $Y_G \setminus v$ and output the resulting blocking set Y'_G.

From now on, we assume that each vertex of G is contained in at least one maximum independent set, and that G is connected. By Proposition 3.2, there is a tree of bridges T in G such that $\mathsf{bd}(G \setminus V(T)) < \mathsf{bd}(G)$. Let $C_1, \ldots, C_k$ be the connected components of the graph $G \setminus E(T)$, that is, the graph we obtain by removing all bridges of T but keeping the vertices incident on them. (We may have $k = 1$ if T consists of a single vertex.) Since T is a tree of bridges, every connected component C_i contains exactly one vertex of T; denote this vertex by t_i.

(**Constructing an auxiliary graph**) To identify a bounded-size blocking set in G, we build an auxiliary bipartite graph H and vertex subset Y_H, as follows.

1. For each $i \in [k]$, do the following.
   a. Add a vertex set Z_i^+ of size $\alpha(C_i)$ to H. Select $\alpha(C_i) - \alpha(C_i \setminus Y_G)$ vertices from Z_i^+ arbitrarily, and add them to Y_H.
   b. Add a vertex set Z_i^- of size $\alpha(C_i \setminus t_i)$ to H. (Z_i^- may be empty.) Select $\alpha(C_i \setminus t_i) - \alpha(C_i \setminus t_i \setminus Y_G)$ vertices from Z_i^- arbitrarily, and add them to Y_H.
   c. Add all possible edges between Z_i^+ and Z_i^-.
2. For each edge $\{t_i, t_j\}$ of T, add all possible edges between Z_i^+ and Z_j^+.

One can think of H as being obtained from the tree T by attaching a degree-1 pendant leaf to each vertex, and then blowing each vertex up into a set of false twins (vertices with the same open neighborhoods), whose size is determined by an independence number. From this interpretation, it is easy to verify that H is indeed bipartite. Note that for each $i \in [k]$, all vertices of Z_i^+ belong to the same partite set. Similarly, all vertices of Z_i^- belong to the same (but opposite) partite set. Intuitively, the bipartite graph H captures the structure of independent sets in G: for each component C_i you can choose whether or not to include the attachment point t_i in your independent set or not. If you do, then you can get the corresponding $\alpha(C_i)$ vertices from set Z_i^+, but the adjacencies to sets Z_j^+ for $t_j \in N_T(t_i)$ then prevent you from picking an independent set of size $\alpha(C_j)$ from Z_j^+ of the neighboring components. If you do not use t_i in the independent set, you can pick the $\alpha(C_i \setminus t_i)$ vertices from Z_i^- instead, which does not impose any restrictions on what you choose for neighboring components. The set Y_H is chosen in such a way that the loss of having to avoid Y_G in an independent set in G, corresponds to the loss of having to avoid Y_H in an independent set for H. We now formalize these ideas.

▷ Claim 5.5. $\alpha(G) = \alpha(H)$ and $\alpha(G \setminus Y_G) = \alpha(H \setminus Y_H)$.

Proof. Consider an independent set S_G in G (respectively, in $G \setminus Y_G$). Initialize S_H as an empty set. For each $t_i \in V(T)$, if $t_i \in S_G$ then add Z_i^+ to S_H (respectively, add $Z_i^+ \setminus Y_H$ to S_H). If $t_i \notin S_G$, then add Z_i^- (respectively, $Z_i^- \setminus Y_H$) to S_H. Observe that by Steps 1a and 1b, in either case we add at least $|V(C_i) \cap S_G|$ vertices to S_H, therefore $|S_H| \geq |S_G|$. To see that S_H is independent in H, observe that if $t_i \in S_G$ and we use Z_i^+, then all $t_j \in N_T(t_i)$ do not belong to S_G since T is a subgraph of G, and hence we use the sets Z_j^- for such t_j. Hence $\alpha(H) \geq \alpha(G)$. Since the set S_H avoids Y_H if S_G avoids Y_G, we also have $\alpha(H \setminus Y_H) \geq \alpha(G \setminus Y_G)$.

Consider an independent set S_H in H (respectively, in $H \setminus Y_H$). For each $t_i \in V(T)$, by Step 1c of the construction, the set S_H cannot contain vertices from both Z_i^+ and Z_i^-. If $S_H \cap Z_i^+ \neq \emptyset$, then add a maximum independent set of C_i (respectively, of $C_i \setminus Y_G$) to S_G, which has size at least $|Z_i^+ \cap S_H|$. Otherwise, add a maximum independent set of $C_i \setminus t_i$ (respectively, of $(C_i \setminus t_i) \setminus Y_G$) to S_G, which has size at least $|Z_i^- \cap S_H|$. Hence $|S_G| \geq |S_H|$, and S_G avoids Y_G if S_H avoids Y_H. Using the fact that the components C_i arose by deleting the edges of the tree T, it is easy to verify that S_G is independent in G. Hence $\alpha(G) \geq \alpha(H)$ and $\alpha(G \setminus Y_G) \geq \alpha(H \setminus Y_H)$. ◁

Since Y_G is a blocking set in G, Claim 5.5 shows that Y_H is a blocking set in H. Since H is bipartite, by Lemma 5.2 there is a blocking set $Y'_H \subseteq Y_H$ in H of size at most two, with the guarantee that if it has size two then its vertices belong to opposite partite sets. Before using Y'_H, we establish a structural claim that will be useful later.

▷ Claim 5.6. For each $t_i \in V(T)$ we have $\alpha(C_i) = \alpha(C_i \setminus N_{C_i}[t_i]) + 1$.

Proof. Since each vertex of G is contained in a maximum independent set by the assumption in the beginning of the proof, there is a maximum independent set S_G of G containing t_i. Then $S_G \cap V(C_i)$ is a maximum independent set of C_i which contains t_i: since t_i is the only vertex that has neighbors in G outside of C_i, if there was an independent set in C_i larger than $S_G \cap V(C_i)$ then we could substitute it into S_G to obtain a larger independent set of G, which is impossible. So there is a maximum independent set of C_i that contains t_i, implying that removing t_i and its closed neighborhood from C_i decreases the independence number by exactly one. ◁

(**Building a blocking set**) Now we build a small blocking set $Y'_G \subseteq Y_G$ for G, as follows. Initialize Y'_G as an empty set, and do the following for each vertex $y \in Y'_H \subseteq Y_H$.

(I) If $y \in Z_i^+$ for some i and $t_i \in Y_G$, then add t_i to Y'_G.

(II) If $y \in Z_i^+$ for some i but $t_i \notin Y_G$, then let $C'_i := C_i \setminus N_{C_i}[t_i] = C_i \setminus N_G[t_i]$. Since C'_i is a subgraph of $G \setminus V(T)$, we have $\mathsf{bd}(C'_i) < \mathsf{bd}(G)$. We claim that $\alpha(C'_i \setminus Y_G) < \alpha(C'_i)$. To see that, note that $y \in Z_i^+$ implies that $Z_i^+ \cap Y_H \neq \emptyset$, which implies by Step 1a that $\alpha(C_i) > \alpha(C_i \setminus Y_G)$. By Claim 5.6, we know $\alpha(C'_i) = \alpha(C_i \setminus N_{C_i}[t_i]) = \alpha(C_i) - 1$. If $\alpha(C'_i \setminus Y_G) = \alpha(C'_i)$, then any independent set of this size in $C'_i \setminus Y_G$ combines with t_i to form a maximum independent set in C_i that is disjoint from Y_G, contradicting the fact that $\alpha(C_i) > \alpha(C_i \setminus Y_G)$. Hence Y_G is indeed a blocking set for C'_i and we may invoke induction on C'_i and $Y_G \cap V(C'_i)$ to obtain a blocking set $Y_G^y \subseteq Y_G$ of C'_i of size at most $2^{\mathsf{bd}(C'_i)} \leq 2^{\mathsf{bd}(G)-1}$. We add Y_G^y to Y'_G.

(III) If $y \in Z_i^-$ for some i, then $Z_i^- \cap Y_H \neq \emptyset$ which implies by Step 1b that $\alpha(C_i \setminus t_i) > \alpha((C_i \setminus t_i) \setminus Y_G)$, hence $Y_G \cap V(C_i \setminus t_i)$ is a blocking set for $C_i \setminus t_i$. Since $C_i \setminus t_i$ is a subgraph of $G \setminus V(T)$, we have $\mathsf{bd}(C'_i) < \mathsf{bd}(G)$. Hence we may invoke induction on $C_i \setminus t_i$ and $Y_G \cap V(C_i \setminus t_i)$ to obtain a blocking set Y_G^y for $C_i \setminus t_i$ of size at most $2^{\mathsf{bd}(G)-1}$. We add Y_G^y to Y'_G.

Since $|Y'_H| \leq 2$, the process above results in a set $Y'_G \subseteq Y_G$ of size at most $2^{\mathsf{bd}(G)}$. To complete the proof, it suffices to show that Y'_G is indeed a blocking set in G.

▷ Claim 5.7. $\alpha(G \setminus Y'_G) < \alpha(G)$.

Proof. Assume for a contradiction that $\alpha(G \setminus Y'_G) = \alpha(G)$, and let S_G be a maximum independent set in G disjoint from Y'_G. We use a similar process as in the proof of Claim 5.5 to build a maximum independent set S_H in H disjoint from Y'_H, contradicting the fact that Y'_H is a blocking set in H.

Initialize S_H as the empty set. For each $t_i \in V(T)$, we do the following.

- If $t_i \in S_G$ and $Y'_H \cap Z_i^+ = \emptyset$, then add Z_i^+ to S_H. Note that $|Z_i^+| \geq |S_G \cap V(C_i)|$ by Step 1a.
- If $t_i \notin S_G$ and $Y'_H \cap Z_i^- = \emptyset$, then add Z_i^- to S_H. Note that $|Z_i^-| \geq |S_G \cap V(C_i)|$ by Step 1b.
- If $t_i \in S_G$ and $Y'_H \cap Z_i^+ \neq \emptyset$, then we claim that $|Y'_H \cap Z_i^+| = 1$. This follows from the fact that if Y'_H has size two, then by Lemma 5.2 its two vertices belong to opposite partite sets, while all of Z_i^+ belongs to the same partite set of H. Since $t_i \in S_G$ while S_G avoids Y'_G and $Y'_H \cap Z_i^+ \neq \emptyset$, Step I ensures that $t_i \notin Y_G$. Hence during the construction of Y'_G we executed Step II on account of the unique vertex in $Y'_H \cap Z_i^+$, which caused a blocking set for $C'_i := C_i \setminus N_{C_i}[t_i]$ to be added to Y'_G. Since S_G avoids Y'_G, it follows that $|S_G \cap V(C'_i)| < \alpha(C'_i)$. As $|S_G \cap N_{C_i}[t_i]| = 1$ since $t_i \in S_G$, together with Claim 5.6 this implies

$$|S_G \cap V(C_i)| = |S_G \cap V(C'_i)| + 1 < \alpha(C'_i) + 1 = \alpha(C_i) = |Z_i^+|.$$

Now add $Z_i^+ \setminus Y'_H$ to S_H, which has size at least $|S_G \cap V(C_i)|$ and is disjoint from Y'_H.
- If $t_i \notin S_G$ and $Y'_H \cap Z_i^- \neq \emptyset$, then similarly as in the previous case we have $|Y'_H \cap Z_i^-| = 1$, and on account of this vertex we executed Step III when constructing Y'_G. Hence Y'_G contains a blocking set for $C_i \setminus t_i$, and together with the assumption $t_i \notin S_G$ this implies $|S_G \cap V(C_i)| \leq \alpha(C_i \setminus t_i) - 1 = |Z_i^-| - 1$. Now add $Z_i^- \setminus Y'_H$ to S_H, which has size at least $|S_G \cap V(C_i)|$ and is disjoint from Y'_H.

It follows directly from the construction that S_H is at least as large as S_G and is disjoint from Y'_H. The fact that S_H is independent follows for the same reasons as in Claim 5.5. Since S_G is a maximum independent set in G, Claim 5.5 then implies that S_H is a maximum independent set in H. But this contradicts the fact that Y'_H is a blocking set in H. ◁

Claim 5.7 shows that Y'_G is a blocking set in G, which concludes the proof of Theorem 5.4. ◀

Note that Theorem 5.4 and Theorem 4.14 together prove Theorem 1.2. We finish the section by showing that the upper-bound of $2^{\mathsf{bd}(G)}$ on the size of minimal blocking sets is tight.

▶ **Theorem 5.8.** *For every $c \in \mathbb{N}$, there is a graph G with $\mathsf{bd}(G) \leq c$ that contains a minimal blocking set of size 2^c.*

Proof. Recall the notion of triangle-path from Definition 4.9. For $t \geq 2$, let a *truncated triangle-path of length t* be the graph U_t obtained from a triangle-path of length t by removing vertices a_1 and b_t; see Figure 2. Analogously to Observation 4.10, we show that $Y_t := \{c_i \mid i \in [t]\}$ is a minimal blocking set in U_t. Since Y_t is an independent set of size t, while (the remainders of) the triangles in U_t partition the vertices of U_t into t cliques, it follows that $\alpha(U_t) = t$. The set Y_t is a blocking set, since $U_t \setminus Y_t$ is a path on $2(t-1)$ vertices, whose independence number is only $t-1$. Finally, it is easy to see that for any $y \in Y_t$, there is a size-t independent set in $U_t \setminus (Y_t \setminus y)$ that consists of the vertex y and, for every (remainder of a) triangle in U_t, the vertex closest to y.

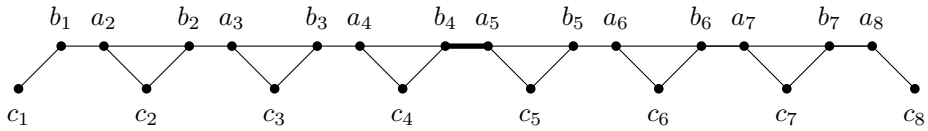

Figure 2 Truncated triangle path U_8 of length 8, illustrating Theorem 5.8. Removing the fat middle bridge and its incident vertices, leaves two connected components isomorphic to U_4.

Hence U_t has a minimal blocking set of size t, for all $t \geq 2$. To prove the theorem, it therefore suffices to show that $\mathsf{bd}(U_{2^c}) \leq c$ for all $c \in \mathbb{N}$. We prove this by induction on c. For $c = 1$, note that the graph U_2 is just the four-vertex path. Hence it is a forest, implying $\mathsf{bd}(U_2) = 1$ by Proposition 3.2. For $c > 1$, consider the graph U_{2^c}. By construction, the middle edge $e = \{b_{2^{c-1}}, a_{2^{c-1}+1}\}$ is a bridge in U_{2^c}. Let T be the tree in U_{2^c} consisting of the single bridge e. Note that removing $V(T)$ splits U_{2^c} evenly, into two connected components that are both isomorphic to $U_{2^{c-1}}$. By induction, $\mathsf{bd}(U_{2^{c-1}}) \leq c-1$. Then Proposition 3.2 shows that $\mathsf{bd}(U_{2^c}) \leq 1 + \mathsf{bd}(U_t \setminus V(T)) = 1 + (c-1) = c$. ◀

6 Kernelization for modulators to bounded bridge-depth

To establish the positive direction of Theorem 1.1, we develop a polynomial kernel for Vertex Cover parameterized by the size of a modulator X whose removal leaves a graph

of constant bridge-depth; an approximately optimal such set X can be computed using Proposition 3.5. As the kernelization is technical and consists of many different reduction rules, with a nontrivial size analysis, we first present below the high-level idea behind the kernelization and the role of bridge-depth.

High-level ideas of the kernelization algorithm. Consider an instance (G, k) of Vertex Cover with a modulator X such that $\mathsf{bd}(G \setminus X)$ is bounded by the constant given by the graph class $\mathcal{F}$. As explained in the introduction, using the fact that minimal blocking sets for the components C of $G \setminus X$ have bounded size, the number of such components can easily be bounded by $|X|^{\mathcal{O}(1)}$. To bound the size of individual components, the definition of bridge-depth ensures that in each connected component C of $G \setminus X$ there is a tree of bridges $T \subseteq E(C)$ (called a *lowering tree*) such that removing the vertex set $V(T)$ from C decreases the bridge-depth of C. By designing new problem-specific reduction rules, we shrink the tree of bridges to size polynomial in the parameter. This is where the main technical work of the kernelization step lies. It properly subsumes the earlier kernelization for the parameterization by distance to a forest, which is imported as a black box in several previous works [4, 17–19, 27]. Having bounded the number of components of $G \setminus X$, together with the size of a lowering tree of bridges in each component, we now proceed as follows: in each component C of $G \setminus X$ we move the vertices from a lowering tree of bridges into the set X. This blows up $|X|$ by a polynomial factor, but strictly decreases the bridge-depth of the graph $G \setminus X$. We then recursively kernelize the resulting instance. When the bridge-depth of $G \setminus X$ reaches zero, the graph $G \setminus X$ is empty and the kernelization is completed. Full details are given below. We start with a formal description of the algorithm in Section 6.1 (cf. Algorithm 1), where we also explain how the remainder of this section is organized.

The negative direction of Theorem 1.1, presented in Section 6.5, is much easier to establish. Using the fact that a minor-closed family $\mathcal{F}$ of unbounded bridge-depth contains all triangle paths, a kernelization lower bound for modulators to such $\mathcal{F}$ follows easily using known gadgets.

6.1 Formal description of the kernelization algorithm

For each integer $c \in \mathbb{N}$, we will obtain a kernel for the following parameterized problem.

Independent Set with c-bridge-depth modulator (IS/c-bd-mod) **Parameter:** $\lvert X\rvert$.
Input: Undirected graph G, integer k, and a set $X \subseteq V(G)$ such that $\mathsf{bd}(G \setminus X) \leq c$.
Question: Does G have an independent set of size k?

Note that, for structural graph parameterizations that do not refer to the solution size, the IS and VC problems are equivalent: an instance (G, X, k) of IS with a structural parameter $|X|$ is equivalent to the similarly-parameterized instance $(G, X, |V(G)| - k)$ of VC, and this reduction preserves the parameter. Similarly as in previous work [4, 22], we therefore present the kernelization algorithm for IS, since it makes some arguments more intuitive.

We may assume that a modulator X is given in the input, since one may use the polynomial-time approximation algorithm given by Proposition 3.5 to obtain a modulator that can be used to compute the kernel. This is a standard assumption also used in related work; see [13, §2.2] for a detailed discussion.

Given an input (G, X, k) of IS/c-bd-mod, we henceforth denote by $R := V(G) \setminus X$ the *remaining* bounded-bridge-depth graph that results from removing the modulator. The

kernelization algorithm is defined in Algorithm 1; throughout this section we present a series of definitions and results that will eventually lead to a proof that this algorithm indeed computes a polynomial kernel for IS/c-bd-mod.

Let us first explain the roadmap of the proof, keeping in mind the outline given in the beginning of this section. Section 6.2 corresponds to the easy part of the kernel, as we invoke the classical machinery to bound the number of connected components of R by a polynomial in the parameter $|X|$. We employ the notions of *conflict* and *chunk* that were introduced in [22] for analyzing how a choice of an independent set from the modulator X affects the number of additional vertices from R that can be added to the independent set. The fact that having bounded minimal blocking set size (in R) allows for an efficient reduction of the number of connected components of R (cf. Lemma 6.6) is implicit in previous work [22, §3.1] [4, Rule 3]; see [19, Thm. 3] for an explicit argument. We use the same kind of arguments in our Rules 1 and 2. While some earlier work [4] uses an *annotated* version of the problem, in this work we introduce the notion of *almost-free* set (cf. Definition 6.2) to avoid having annotations. This allows us to work in a conceptually cleaner setting.

Let us now turn to Section 6.3. When working with a modulator to bounded *tree-depth* [4], once the number of connected components of R is bounded by a polynomial in $|X|$, we can move a tree-depth decreasing vertex from each connected component into the modulator, and get a $(c-1)$-tree-depth-modulator X', with $|X'|$ being still polynomial in $|X|$. This facilitates a recursion on c presented in Section 6.4, immediately leading to a polynomial kernel. In our case, we cannot move an entire lowering tree (cf. Definition 3.6) of a connected component of R into the modulator, as its size may be unbounded. Thus, the main challenge to obtain the polynomial kernel is addressed in Section 6.3, where we define new rules to shrink the size of lowering trees. Notice that shrinking these lowering trees generalizes the reduction rules for Independent Set parameterized by distance to a forest [22], since for any component in R that is a tree, its unique lowering tree consists of the entire component. This explains why we cannot simply use the kernelization for the parameterization by feedback vertex set [22] as a black box, as it was the case in previous work [4, 17–19, 27].

Let us now formally define the above notions.

▶ **Definition 6.1.** *For a graph G and disjoint vertex sets $X', R' \subseteq V(G)$, we define the number of* conflicts induced on R' by X' *as*

$$\mathsf{conf}^G_{R'}(X') = \alpha(G[R']) - \alpha(G[R' \setminus N_G(X')]).$$

When G is clear from the context, $\mathsf{conf}^G_{R'}(X')$ will be simply denoted by $\mathsf{conf}_{R'}(X')$.

Intuitively, $\mathsf{conf}_{R'}(X')$ measures how much smaller the independence number of $G[R']$ becomes when one is forbidden from picking vertices that are adjacent in G to X'. This allows us to reason about which subsets X' are viable candidates for occurring in a maximum independent set of G.

▶ **Definition 6.2.** *Let (G, X, k) be an input of* IS/c-bd-mod *and $R := G \setminus X$.*

- *A* chunk *is an independent set $X' \subseteq X$ of size at most 2^c in G. We denote by $\mathcal{X}$ the set of all chunks.*
- *The* degree *of a chunk X', denoted $d(X')$, is defined as the number of connected components R' of R for which $\mathsf{conf}_{R'}(X') \neq 0$.*
- *A set $Z \subseteq V(R)$ is* free *if $\mathsf{conf}_Z(X') = 0$ for all $X' \in \mathcal{X}$.*
- *A set $Z \subseteq V(R)$ is* x-almost-free *for $x \in \mathbb{N}$, if for any $X' \in \mathcal{X}$ such that $\mathsf{conf}_Z(X') \neq 0$, it holds that $\mathsf{conf}_R(X') \geq x$.*

Slight abusing notation, for a subgraph R' of R we will sometimes write $\mathsf{conf}_{R'}(X')$ as a shorthand for $\mathsf{conf}_{V(R')}(X')$.

Note the subtle difference in the subscripts of conf for the definition of almost-free. Intuitively, if Z is x-almost-free, then any chunk that makes a conflict on the subgraph induced by Z, will make at least x conflicts on the entire bounded bridge-depth graph R. For $x \geq |X|$, this will allow us to infer that there is a maximum independent set of G that does not contain any chunk that makes a conflict on Z.

▶ **Lemma 6.3.** *Let (G, X, k) be an instance of* IS/c-bd-mod*. There is a polynomial-time algorithm to compute the function* conf *for any $R' \subseteq V(R)$, to compute the degree d of a chunk, and to decide if a subset $Z \subseteq V(R)$ is free or x-almost free.*

Proof. For any $R' \subseteq V(R)$ we have $\mathsf{tw}(G[R']) \leq \mathsf{bd}(G[R']) \leq c$, by Item 6 of Proposition 3.2. Since IS is fixed-parameter tractable parameterized by treewidth (cf. [7, §7.3.1]), and a tree decomposition of constant width c can be computed in linear time [1], this means independence numbers of subgraphs of $G \setminus X$ can be computed in polynomial time. Since the number of potential chunks is polynomial in $|X|$ since c is a constant, this allows all mentioned quantities to be efficiently computed. ◀

The following lemma shows why having small blocking sets is useful to characterize the interaction between maximum independent sets in R and chunks.

▶ **Lemma 6.4.** *Let (G, X, k) be an instance of* IS/c-bd-mod *and let $R' \subseteq V(R)$. For every independent set $S_X \subseteq X$ such that $\mathsf{conf}_{R'}(S_X) \neq 0$ there exists a chunk $X' \in \mathcal{X}$, with $X' \subseteq S_X$, such that $\mathsf{conf}_{R'}(X') \neq 0$.*

Proof. Let $Z = N_G(S_X) \cap R'$. The fact that $\mathsf{conf}_{R'}(S_X) \neq 0$ implies that Z is a blocking set of $G[R']$. As $\mathsf{bd}(R') \leq c$, which implies that $\mathsf{mbs}(R') \leq 2^c$ by Theorem 5.4, there exists $Z' \subseteq Z$ with $|Z'| \leq 2^c$ such that Z' is a blocking set of R'. Thus, with any $z \in Z'$ we associate a vertex $v_z \in S_X$ such that v_z is adjacent to z, and we get that $X' := \{v_z \mid z \in Z'\}$ is the desired chunk. ◀

▶ **Corollary 6.5.** *Let (G, X, k) be an instance of* IS/c-bd-mod*. A set $Z \subseteq V(R)$ is free if and only if for any independent set $S \subseteq X$, $\mathsf{conf}_Z(S) = 0$ (or equivalently $\alpha(G[Z \setminus N_G(S)]) = \alpha(G[Z])$).*

In Algorithm 1 we present the pseudo-code of the kernelization algorithm. The reduction rules employed by the algorithm will be presented in the subsequent sections.

6.2 Bounding the number of connected components

Let us first define some rules that will be used in Algorithm 1. In these rules, we let k' denote the desired independent set size of the resulting instance.

▶ **Rule 1.** *Let (G, X, k) be an input of* IS/c-bd-mod*. If there exists a connected component R' of R such that $V(R')$ is free, then delete $V(R')$ from G and define $k' = k - \alpha(R')$.*

Proof of safeness. Let G' be the graph obtained after applying the rule. We just prove that $\alpha(G') \geq k'$ implies $\alpha(G) \geq k$, as the other implication is straightforward. Let S' be an independent set of G' with $|S'| \geq k'$. Recall that for any subset V' we use $S'_{V'}$ to denote $S' \cap V'$. As $V(R')$ is free, according to Corollary 6.5 we get that $\mathsf{conf}_{R'}(S'_X) = 0$, implying that there exists a maximum independent set Z of R' such that $S' \cup Z$ is independent in G, implying the desired inequality. ◀

Algorithm 1 A polynomial kernel for IS/c-bd-mod

Input: an input (G, X, k, c) of IS/c-bd-mod
Output: an equivalent instance of size polynomial (for fixed c) in $|X|$ (see Theorem 6.22)

1 **if** $c = 0$ **then**
2 ⌊ return G
3 If Rule 1 can be applied to (G, X, k), apply it and restart from Line 1
4 If Rule 2 can be applied to (G, X, k), apply it and restart from Line 1
5 /* #cc($G[R]$) is now bounded according to Lemma 6.6 */
6 Define $X_1 = \emptyset$
7 **for** *each connected component* R' *of* R **do**
8 Compute a lowering tree T /* done in polynomial time by Proposition 3.7 */
9 Compute a longest path T' in T
10 If Meta-Rule 1 can be applied to $((G, X, k), T')$, apply it and restart from Line 1
11 If Meta-Rule 2 can be applied to $((G, X, k), T')$, apply it and restart from Line 1
12 /* T now has bounded diameter diam(T) according to Lemma 6.13 */
13 If Meta-Rule 3 can be applied to $((G, X, k), T)$, apply it and restart from Line 1
14 /* T now has bounded max. degree $\Delta(T)$ according to Lemma 6.15 */
15 /* It now remains to bound the number of leaves of T */
16 Define T_1 by removing from T any A-leaf that has a B-parent
17 /* T_1 has no A-leaf with B-parent, $\Delta(T_1) \leq \Delta(T)$ and diam(T_1) $\leq$ diam(T) according to Lemma 6.16 */
18 If Meta-Rule 1 can be applied to $((G, X, k), T_1)$, apply it and restart from Line 1
19 /* $n_{*\mathsf{A}}(T_1)$ is now bounded according to Lemma 6.18 */
20 If Meta-Rule 4 can be applied to $((G, X, k), T_1)$, apply it and restart from Line 1
21 /* $n_{\mathsf{BBB}}(T_1)$ is now bounded according to Lemma 6.19 */
22 If Meta-Rule 5 can be applied to $((G, X, k), T_1)$, apply it and restart from Line 1
23 /* $n_{\mathsf{BB2}}(T_1)$ is now bounded according to Lemma 6.20 */
24 /* $|V(T)|$ is now bounded according to Lemma 6.21 */
25 ⌊ $X_1 := X_1 \cup V(T)$
26 Define $X_2 = X \cup X_1$
27 /* $|X_2|$ is bounded and is a $(c-1)$-bd-modulator */
28 Return Algorithm 1$(G, X_2, k, c-1)$

▶ **Rule 2.** *Let (G, X, k) be an input of* IS/c-bd-mod*. If there exists a connected component R' of R such that for any chunk X' with $\mathsf{conf}^G_{R'}(X') \neq 0$ we have that $d(X') \geq |X| + 1$, then delete all edges from X to R' and define $k' = k$.*

Proof of safeness. Let G' be the graph obtained by applying the rule; then $\alpha(G) \geq k$ trivially implies $\alpha(G') \geq k$. Let us prove that $\alpha(G') \geq k$ implies that $\alpha(G) \geq k$. Let S' be an independent set of G' of size at least k. If $\mathsf{conf}^G_{R'}(S'_X) = 0$ then there exists a maximum independent set $S^*_{R'}$ of $G[R']$ such that $S'_X \cup S^*_{R'}$ is still independent in G. Thus, in this case we can replace $S'_{R'}$ by $S^*_{R'}$ in S' and get an independent set of G of size at least $|S'|$. Otherwise, $\mathsf{conf}^G_{R'}(S'_X) \neq 0$ implies, by Lemma 6.4, that there exists a chunk $X' \subseteq S'_X$ such that $\mathsf{conf}^G_{R'}(X') \neq 0$. Since the rule applied, we know that $d(X') \geq |X| + 1$, implying that $\mathsf{conf}_{R \setminus R'}(S'_X) \geq \mathsf{conf}_{R \setminus R'}(X') \geq d(X') - 1 \geq |X|$. Thus, consider any maximum

independent set $S^*_{R\setminus R'}$ of $R \setminus R'$ and any maximum independent set $S^*_{R'}$ of R'. We have $|S^*_{R\setminus R'}| \geq |S'_X \cup S'_{R\setminus R'}|$, implying that $S^*_{R\setminus R'} \cup S^*_{R'}$ is an independent set of G of size at least $|S'|$. ◀

Notice that applying Rule 2 may trigger the application of Rule 1. Observe also that applying Rule 2 does not guarantee that $d(X') \leq |X|$ for any chunk X'. However, as proved in the next lemma, applying Rules 1 and 2 exhaustively is sufficient to bound the number of connected components.

▶ **Lemma 6.6.** *Let (G, X, k) be an input of* IS/c-bd-mod*. Let (G', X', k') be the input of* IS/c-bd-mod *obtained after applying exhaustively Rule 1 and Rule 2. Recall that $R = V(G) \setminus X$, and let $\tilde{R} = V(G') \setminus X$. Then,*

- *$X' = X$ and $k' \leq k$,*
- *G' is a subgraph of G and $\tilde{R} \subseteq R$, and*
- *$\#\textsc{cc}(G'[\tilde{R}]) \leq |\mathcal{X}| \cdot |X|$.*

Proof. The claims in the first two items hold as, given an input (G, X, k) of IS/c-bd-mod, Rule 1 and Rule 2 either delete connected components of R, or delete edges between X and R. As $X' = X$, the set of chunks in (G', X, k) is also $\mathcal{X}$. Let us now turn to the last claim. Let R' be a connected component of G'. As Rule 1 cannot be applied, by Corollary 6.5 we have that $\{X' \in \mathcal{X} \mid \mathsf{conf}^{G'}_{R'}(X') \neq 0\} \neq \emptyset$. As Rule 2 cannot be applied, we can associate with R' a chunk $X^{R'} \in \mathcal{X}$ such that $\mathsf{conf}^{G'}_{R'}(X^{R'}) \neq 0$ and $d(X^{R'}) \leq |X|$. Thus, if $\#\textsc{cc}(G'[\tilde{R}]) > |\mathcal{X}| \cdot |X|$, by the pigeonhole principle more than $|X|$ different connected components will be associated with the same chunk X', contradicting the fact that $d(X') \leq |X|$. ◀

We remark that by using a more sophisticated marking scheme for connected components based on maximum matchings in an auxiliary bipartite graph [19, Thm 3], it is possible to reduce the number of connected components to $|\mathcal{X}|$. We have opted for a simpler approach since it suffices to obtain polynomial kernels that complete the dichotomy.

6.3 Bounding the size of a lowering tree of each connected component

Before formally defining the tools we need, let us explain the common ideas behind all the Meta-Rules (except Meta-Rule 3) of Sections 6.3.1 and 6.3.2, which are designed to shrink a lowering tree T of a connected component R' of R. Roughly speaking, a T-conflict structure C is formed by a subset of vertices of T together with the connected components of $R - E(T)$ that they are contained in (see Definition 6.8), which has a simple structure that allows us to shrink T locally provided that C does not interact too much with the modulator X. The condition of not interacting too much is captured by the notion of C being almost free in our formalism (more precisely $(|X| + \Delta)$-almost-free where Δ is a constant). This explains why all Meta-Rules below (except Meta-Rule 3) take the following form:

> "If there is a T-conflict structure C that is $(|X| + \Delta)$-almost-free, then reduce T (by performing local modification around C)."

Suppose now that such a rule does not apply anywhere on a lowering tree T of a connected component R' of R. The idea to bound the size of T is as follows. If by contradiction T is very large, then we can find a partition $\mathcal{P} = \{V_i^T \mid i \in [|\mathcal{P}|]\}$ of T (and thus a partition of R' by defining $V_i^{R'} = \bigcup_{v \in V_i^T} H_v^T$ and $\mathcal{P}^{R'} = \{V_i^{R'} \mid i \in [|\mathcal{P}|]\}$, where H_v^T is defined in Definition 6.7) such that many of the parts $V' \in V_i^{R'}$ are T-conflict structures. As the

Meta-Rule cannot be applied on any of these parts, it implies that none of them is $(|X|+\Delta)$-almost-free. Now we conclude by using the definition of almost-free as follows. If a part V' is not $(|X|+\Delta)$-almost-free, it means that V' has a "private" chunk $X'_{V'} \in \mathcal{X}$ such that $\mathsf{conf}_{V'}(X'_{V'}) \neq 0$ and $\mathsf{conf}_{R'}(X'_{V'}) < |X|+\Delta$. By a pigeonhole argument, if T is too large, then there will exist a chunk X' that is the private chunk of many (say $x \geq |X|+\Delta$) different V'. However, and this is where the notion of *α-additive partition* (see Definition 6.10) comes into play, as we will define $\mathcal{P}^{R'}$ such that it is an α-additive partition, such a chunk will have $\mathsf{conf}_{R'}(X') \geq x$, contradicting the fact that $\mathsf{conf}_{R'}(X') < |X|+\Delta$. The notion of *types* (see Definition 6.7) also plays a crucial role in our approach. Indeed, given a lowering tree T (or any tree of bridges) of a connected component R' of R, types allow us to capture in a simple way the condition we need in T and in the relations between T and $R' \setminus V(T)$, while hiding the complex structure of $G[R' \setminus V(T)]$. For example, a conflict structure of type 1 only needs two adjacent (in T) vertices u and v where at least one has type A, regardless of the exact structure of H_u^T and H_v^T.

▶ **Definition 6.7.** *Let R be a graph, let T be a tree of bridges of R, and let $R^* = R \setminus E(T)$. For a vertex $v \in V(T)$, the* pending component of v in T, *denoted by H_v^T, or simply H_v when clear from context, is the connected component of R^* containing v. We call v the* root *of H_v. We say that v has*
- type A in T *if $\alpha(R[H_v]) = \alpha(R[H_v \setminus \{v\}])$, and*
- type B in T *if $\alpha(R[H_v]) = \alpha(R[H_v \setminus \{v\}]) + 1$.*

Notice that for any distinct vertices $u, v \in V(T)$ we have $H_u \cap H_v = \emptyset$. Moreover, since the pending components arise by removing a tree of bridges, it follows that the only vertex of H_u that can have neighbors outside H_u is the root u itself.

When working with pending components, like in the following definition, we will sometimes refer to the vertex set of a component H_v simply as H_v, if there is no risk of confusion.

▶ **Definition 6.8.** *Let R be a graph and let T be a tree of bridges of R. We say that a subset of vertices $C \subseteq V(R)$ is a T-*conflict structure
- of type 1 *if $C = H_{v_1} \cup H_{v_2}$ where $\{v_1, v_2\} \in E(T)$, and v_1 or v_2 has type A in T,*
- of type 2 *if $C = H_{v_1} \cup H_{v_2}$ where*
  - *there exists $u_1, u_2 \in V(T)$ such that (u_2, v_1, v_2, u_1) is a path in T,*
  - *$d_T(v_1) = d_T(v_2) = 2$, and*
  - *v_1 and v_2 have type B in T,*
- of type 3 *if $C = H_u$ where u is a leaf of T and has type B,*
- of type 4 *if $C = H_{v_1} \cup H_{v_2}$ where*
  - *there exists $u \in V(T)$ such that (v_1, v_2, u) is a path in T,*
  - *v_1 is a leaf in T and $d_T(v_2) = 2$, and*
  - *v_1 and v_2 have type B in T.*

The following lemma holds by the same arguments used in the proof of Lemma 6.3.

▶ **Lemma 6.9.** *Let (G, X, k) be an instance of* IS/c-bd-mod *and T be a tree of bridges of a connected component R' of R. Computing the type of a vertex $v \in V(T)$, as well as deciding if there exists a T-conflict structure C of any fixed type, can done in polynomial time.*

The following definition will be useful to argue that a chunk makes a large number of conflicts on R, which will in turn allow us to argue that a reduction rule must be applicable if the instance is too large compared to the parameter. Namely, we will use Lemma 6.11 in the proofs of Lemmas 6.13, 6.18, and 6.19.

▶ **Definition 6.10.** *An* α-additive partition $\mathcal{P}$ *of a graph* R *is a partition* $\mathcal{P} = \{V_i \mid i \in [|\mathcal{P}|]\}$ *of* $V(R)$ *such that* $\alpha(R) = \sum_{i\in[|\mathcal{P}|]} \alpha(R[V_i])$.

Observe that if $\mathcal{P}$ is an α-additive partition of the graph $R = G\setminus X$ of an instance (G, X, k), then the number of conflicts induced on R by X' is at least as large as the number of blocks $V' \in \mathcal{P}$ of the partition for which $\mathsf{conf}_{V'}(X') \neq 0$.

▶ **Lemma 6.11.** *Let* (G, X, k) *be an input of* IS/c-bd-mod, R' *be a connected component of* R, *and* $\mathcal{P}$ *be an* α*-additive partition of* $G[R']$. *Suppose* $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$, *where each* $V' \in \mathcal{P}_1$ *is not* x*-almost-free. Then,* $|\mathcal{P}_1| < |\mathcal{X}| \cdot x$.

Proof. For any $V' \in \mathcal{P}_1$, as V' is not x-almost-free there exists $X^{V'} \in \mathcal{X}$ such that $\mathsf{conf}_{V'}(X^{V'}) \neq 0$ and $\mathsf{conf}_R(X^{V'}) < x$. Suppose for a contradiction that $|\mathcal{P}_1| \geq |\mathcal{X}| \cdot x$. By the pigeonhole principle, this implies that there exists $\mathcal{P}_1' \subseteq \mathcal{P}_1$ with $|\mathcal{P}_1'| \geq x$ and $X' \in \mathcal{X}$ such that $\mathsf{conf}_{V'}(X') \neq 0$ for all $V' \in \mathcal{P}_1'$, while $\mathsf{conf}_R(X') < x$. As $\mathcal{P}$ is an α-additive partition, this implies that $\mathsf{conf}_{R'}(X') \geq |\mathcal{P}_1'| \geq x$. Finally, as $\mathsf{conf}_R(X') \geq \mathsf{conf}_{R'}(X')$ since R' is a connected component of R, we get our contradiction. ◀

Finally, we need the following technical lemma. It will be used in the safeness proofs of our rules to say that if we have a chunk $X' \in \mathcal{X}$ that induces many conflicts on R in the instance (G, X, k) before the reduction, then X' also induces many conflicts in the instance (G', X, k) obtained after the reduction.

▶ **Lemma 6.12.** *Let* (G, X, k) *and* (G', X, k) *be two inputs of* IS/c-bd-mod *such that* $G[X] = G'[X]$ *(implying that* G *and* G' *have the same set of chunks* $\mathcal{X}$*). Suppose that there exist two non-negative integers* Δ_1, Δ_2 *such that*
1. $\alpha(G' \setminus X) \geq \alpha(G \setminus X) - \Delta_1$ *and*
2. *for any* $X' \in \mathcal{X}$ *we have* $\alpha((G \setminus X) \setminus N_G(X')) \geq \alpha((G' \setminus X) \setminus N_{G'}(X')) - \Delta_2$.

Then, for any $X' \in \mathcal{X}$ *it holds that* $\mathsf{conf}^{G'}_{V(G')\setminus X}(X') \geq \mathsf{conf}^{G}_{V(G)\setminus X}(X') - \Delta_1 - \Delta_2$.

Proof. Let $X' \in \mathcal{X}$. We have

$$\begin{aligned}
\alpha((G' \setminus X') \setminus N_{G'}(X')) &\leq \alpha((G \setminus X) \setminus N_G(X')) + \Delta_2 && \text{By (2)}\\
&= \alpha(G \setminus X) - \mathsf{conf}^{G}_{V(G)\setminus X}(X') + \Delta_2 && \text{Definition 6.1}\\
&\leq \alpha(G' \setminus X) + \Delta_1 - \mathsf{conf}^{G}_{V(G)\setminus X}(X') + \Delta_2. && \text{By (1)}
\end{aligned}$$

The desired bound follows. ◀

6.3.1 Bounding the diameter

In the following we say that a rule is a *meta-rule* if it takes as input, in addition to (G, X, k), a tree of bridges that will be computed in the course of the kernelization algorithm (see Algorithm 1). The two following meta-rules are designed to shrink the diameter of the lowering tree of a connected component of R. Whenever we refer to a pending component H_v in the statement of a meta-rule, these should be understood to be pending components for the graph $R = G \setminus X$, so that a pending component for a tree T of bridges in R is a connected component of $R \setminus E(T)$.

▶ **Meta-Rule 1.**
Input: *An input* (G, X, k) *of* IS/c-bd-mod *and a tree of bridges* T *of a connected component* R' *of* R.
Action: *If there exists a* T*-conflict structure of type 1, namely* $C = H_{v_1} \cup H_{v_2}$ *using the*

notation of Definition 6.8, such that C is $(|X|+2)$-almost-free, then remove edge $\{v_1, v_2\}$ and define $k' = k$.

Proof of safeness. Let G' be the graph obtained by removing $\{v_1, v_2\}$. Let us only prove that $\alpha(G') \geq k'$ implies that $\alpha(G) \geq k$ as the other direction is straightforward. Let S' be an independent set of G' with $|S'| \geq k$. If S' contains at most one endpoint of the removed edge $\{v_1, v_2\}$, then it is an independent set in G and we are done. Suppose $\{v_1, v_2\} \subseteq S'$. We distinguish two cases.

Case 1: $\mathsf{conf}_C(S'_X) = 0$. In this case there exists an independent set Z of C such that $S'_X \cup Z$ is still an independent set in G, and $|Z| = \alpha(G[C])$. Now observe that since at least one of v_1, v_2 is of type A, we have $\alpha(G[C]) = \alpha(G[H_{v_1}]) + \alpha(G[H_{v_2}])$: if $i^* \in [2]$ such that v_{i^*} is of type A, then a maximum independent set of $G[H_{v_{i^*}}]$ exists that does not use v_{i^*}, and it combines with a maximum independent set of $G[H_{v_{3-i^*}}]$ into a maximum independent set of $G[C]$. This implies that $|Z| \geq |S'_C|$. For each $i \in [2]$, since the pending components arise by removing the edges of a tree of bridges from $R = G \setminus X$, it follows that v_i is the only vertex of H_{v_i} that potentially has neighbors in R outside of H_{v_i}. Since $Z \cap \{v_1, v_2\} \subseteq S'_C \cap \{v_1, v_2\}$, we get that $Z \cup S'_{R' \setminus C}$ is also an independent set in G. Thus, $Z \cup S'_X \cup S'_{R' \setminus C} \cup S'_{R \setminus R'}$ is an independent set in G of size at least k.

Case 2: $\mathsf{conf}_C(S'_X) \neq 0$. According to Lemma 6.4, there exists $X' \in \mathcal{X}$ with $X' \subseteq S'_X$ such that $\mathsf{conf}_C(X') \neq 0$. As C is $(|X|+2)$-almost-free, this implies that $\mathsf{conf}^G_R(X') \geq |X|+2$. Observe that $\alpha(G' \setminus X) \geq \alpha(G \setminus X)$, and that for any $X' \in \mathcal{X}$ we have $\alpha((G \setminus X) \setminus N_G(X')) \geq \alpha((G' \setminus X) \setminus N_G(X')) - 1$. Thus, we can apply Lemma 6.12 with $\Delta_1 = 0$ and $\Delta_2 = 1$, and we get that $\mathsf{conf}^{G'}_{V(G') \setminus X}(X') \geq |X| + 1$. This implies that $|S'_{V(G') \setminus X}| \leq \alpha(G' \setminus X) - (|X| + 1) \leq \alpha(G \setminus X) - |X|$, implying in turn that $\alpha(G[R]) \geq |S'|$. ◀

▶ **Meta-Rule 2.**

Input: *An input (G, X, k) of* IS/c-bd-mod *and a tree of bridges T of a connected component R' of R.*

Action: *If there exists a T-conflict structure of type 2, namely $C = H_{v_1} \cup H_{v_2}$, with a path (u_2, v_1, v_2, u_1) in T using the notation of Definition 6.8, such that C is $(|X|+1)$-almost-free, then identify u_i and v_i for each $i \in [2]$, and denote by w_i the obtained vertex. Finally, define $k' = k - 1$.*

Proof of safeness. Let G' be the graph obtained after applying the rule and let $U = \{u_1, u_2\}$.

Let us first prove that $\alpha(G) \geq k$ implies that $\alpha(G') \geq k'$. Let S be an independent set of G with $|S| \geq k$. Recall that $S_X = S \cap X$. We distinguish several cases.

Case 1: $|S \cap U| = 0$. In this case, $S' = S \setminus \{v_1, v_2\}$ has size at least $k - 1$ and is still an independent set in G'.

Case 2: $|S \cap U| = 1$. Let $\{u_i\} = S \cap U$. If S also contains v_i then define $S' = (S \setminus \{u_i, v_i\}) \cup \{w_i\}$. Then S' is still an independent set in G' as, in particular, $N_{G'}(w_i) \cap S' = (N_G(u_i) \cup N_G(v_i)) \cap S = \emptyset$, as S contains $\{u_i, v_i\}$. Otherwise (S does not contain v_i), define $S' = S \setminus \{u_i\}$. *Case 3*: $|S \cap U| = 2$. We distinguish two subcases.

Case 3.1: $\mathsf{conf}_C(S_X) = 0$. This implies that there exists an independent set $Z \subseteq C$ such that $|Z| = \alpha(G[C])$ and such that $S_X \cup Z$ is still an independent set in G. As v_1 and v_2 have type B, we get that $\alpha(G[C]) = \alpha(G[H_{v_1}]) + \alpha(G[H_{v_2}]) - 1$. Moreover, we have $|Z \cap \{v_1, v_2\}| = 1$. Suppose $Z \cap \{v_1, v_2\} = \{v_1\}$; the other case follows symmetrically. For $i \in [2]$, recall that v_i is the only vertex of H_{v_i} that potentially has neighbors in R outside of H_{v_i}. Hence we get that $Z \cup Z'$ is also an independent set in G where $Z' = S_{R' \setminus C} \setminus \{u_2\}$. This implies that $\tilde{S} = Z \cup Z' \cup S_X \cup S_{R \setminus R'}$ is an independent set of G. Moreover, we get that $|\tilde{S}| \geq k$. Indeed, as S contains u_1 and u_2, it does not contain v_1 nor v_2, implying that

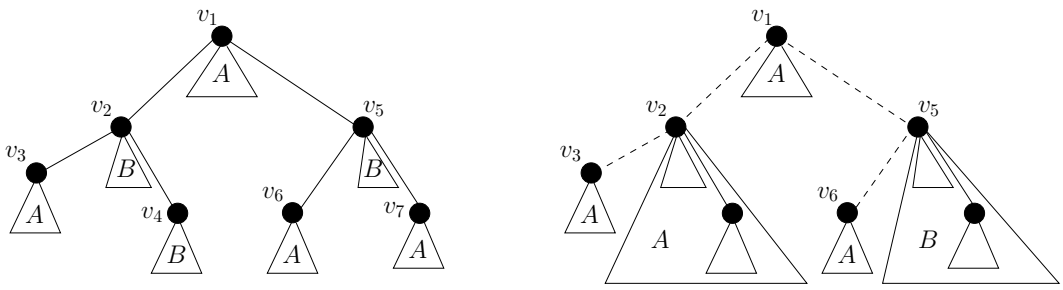

Figure 3 On the left: example of a lowering tree T (with $V(T) = \{v_i \mid i \in [8]\}$) computed in Line 8 of Algorithm 1. On the right: a longest path $T' = (v_3, v_2, v_1, v_5, v_6)$ computed in Line 9. The triangle below each vertex represents its pending component, and the type of each vertex is indicated inside its triangle. Notice that types of vertices in T' may be different from types in T. For example, $H^{T'}_{v_2} = H^T_{v_2} \cup H^T_{v_4}$, implying that v_2 has type A in T', whereas it has type B in T.

$|S_C| < \alpha(G[C])$ as both v_i have type B. This implies that $|Z \cup Z'| \geq |S_C \cup S_{R'\setminus C}|$, leading to $|\tilde{S}| \geq k$. Finally, we define $S' = (\tilde{S} \setminus \{u_1, v_1\}) \cup \{w_1\}$ and we get that S' is an independent set in G' of size at least $k-1$.

Case 3.2: $\mathsf{conf}_C(S_X) \neq 0$. According to Lemma 6.4, there exists $X' \in \mathcal{X}$ with $X' \subseteq S_X$ such that $\mathsf{conf}_C(X') \neq 0$. As C is $(|X|+1)$-almost-free, this implies that $\mathsf{conf}^G_R(X') \geq |X|+1$. This in turn implies that $\alpha(G[R]) \geq |S|+1$. As $\alpha(G[R']) \geq \alpha(G[R]) - 1$, we get the desired result.

Let us now prove that $\alpha(G') \geq k'$ implies that $\alpha(G) \geq k$. Let $W = \{w_1, w_2\}$. Let S' be an independent set of G' with $|S'| \geq k-1$. Suppose first that $|S' \cap W| = 1$ and let $w_i \in S' \cap W$. As w_i is the result of identifying the nonadjacent vertices u_i and v_i, we get that $(S' \setminus \{w_i\}) \cup \{u_i, v_i\}$ is an independent set of G of size at least k. Since the identification has made w_1 and w_2 adjacent in G', in the remainder we have $|S' \cap W| = 0$. Let $H'_{v_i} = H_{v_i} \setminus \{v_i\}$ for $i \in [2]$, and let V' such that we get a partition $R' = C \cup \{u_1, u_2\} \cup V'$ that satisfies $N_G(V') \cap C = \emptyset$. Let us partition $S' = S'_X \cup S'_{H'_{v_1}} \cup S'_{H'_{v_2}} \cup S'_{V'} \cup S'_{R\setminus R'}$. As $|S' \cap W| = 0$ and as both v_i are of type B, we get that $|S'_{H'_{v_1}} \cup S'_{H'_{v_2}}| \leq \alpha(G[C]) - 1$. We distinguish two cases.

Case 1: $\mathsf{conf}_C(S'_X) = 0$. This implies that there exists an independent set Z of $G[C]$ such that $|Z| = \alpha(G[C])$ and $Z \cup S'_X$ is an independent set in G. Moreover, $N_G(V') \cap C = \emptyset$ implies that $Z \cup S'_{V'}$ is an independent set in G. This implies that $S'_X \cup Z \cup S'_{V'} \cup S'_{R\setminus R'}$ is an independent set in G of size at least k.

Case 2: $\mathsf{conf}_C(S'_X) \neq 0$. According to Lemma 6.4, there exists $X' \in \mathcal{X}$ with $X' \subseteq S'_X$ such that $\mathsf{conf}_C(X') \neq 0$. As C is $(|X|+1)$-almost-free, this implies that $\mathsf{conf}^G_R(X') \geq |X|+1$. Observe that $\alpha(G' \setminus X) \geq \alpha(G \setminus X) - 1$, and that for any $X' \in \mathcal{X}$, $\alpha((G \setminus X) \setminus N_G(X')) \geq \alpha((G' \setminus X) \setminus N_G(X'))$. Thus, we can apply Lemma 6.12 with $\Delta_1 = 1$ and $\Delta_2 = 0$, and we get that $\mathsf{conf}^{G'}_{V(G')\setminus X}(X') \geq |X|$. This implies that $|S'_{V(G')\setminus X}| \leq \alpha(G' \setminus X) - |X| \leq \alpha(G \setminus X) - |X|$, implying in turn that $\alpha(G[R]) \geq |S'|$. ◀

Notice that in Algorithm 1 between Lines 8 and 11, we first compute a lowering tree T, but we try to apply Meta-Rules 1 and 2 on a longest path T' of T (and not directly on T). Observe also, as depicted in Figure 3, that types of vertices may change when considered in T or T'. Let us now prove that if neither Meta-Rule 1 nor Meta-Rule 2 can be applied to a longest path T' of a lowering tree T, then T' has bounded length.

▶ **Lemma 6.13.** *Let (G, X, k) be an input of* IS/c-bd-mod*, R' be a connected component of R, T be a lowering tree of R', and T' be a longest path of T. Suppose that neither Meta-Rule 1 nor Meta-Rule 2 can be applied to $((G, X, k), T')$. Then, $|V(T')| \leq \mathcal{O}(|\mathcal{X}| \cdot |X|)$. This implies that after Line 11 of Algorithm 1,* $\mathsf{diam}(T) = \mathcal{O}(|\mathcal{X}| \cdot |X|)$.

Proof. Let $T' = (v_1, \ldots, v_t)$. We consider types and pending components in T'. Let us define a *block* as an inclusion-maximal subpath of T' with only type B vertices. Our goal is to define an α-additive partition of R'. For any block $L = (v_f, \ldots, v_\ell)$, let M^L be the maximum matching using all vertices of L if $|L|$ is even, and all vertices except v_ℓ otherwise. Let $M_1 = \bigcup_{L \text{ is a block}} M^L$. For a block $L = (v_f, \ldots, v_\ell)$ such that $v_\ell \neq v_t$ and L has an odd number of vertices, let $e^L = \{v_\ell, v_{\ell+1}\}$, where $v_{\ell+1}$ has type A. Let M_2 be the union of all such e^L edges. Let M_3 be a maximum matching of type A vertices not belonging to $M_1 \cup M_2$. Finally, let $N = V(T') \setminus (V(M_1) \cup V(M_2) \cup V(M_3))$. Notice that vertices of N are either type A vertices between two blocks that were not matched in M_2, or v_t if the last block contains v_t and has an odd number of vertices.

For any $e \in M_1 \cup M_2 \cup M_3$, $e = \{v_i, v_{i+1}\}$, let $C_e = H_{v_i} \cup H_{v_{i+1}}$. Let $\mathcal{P}_1 = \{C_e \mid e \in M_1 \cup M_2 \cup M_3 \mid C_e \text{ is a } T'\text{-conflict structure (of type 1 or 2)}\}$. Observe that $|\mathcal{P}_1| \leq |M_1 \cup M_2 \cup M_3| - 2$, where we subtract two because if $\{v_1, v_2\}$ or $\{v_{t-1}, v_t\}$ are in M_1, then the corresponding set C_e is not a conflict structure of type 2, as we require that both vertices should be of degree two in T'. Let $\mathcal{P}_1' = \{C_e \mid e \in M_1 \cup M_2 \cup M_3\} \setminus \mathcal{P}_1$ and $\mathcal{P}_2 = \{H_v \mid v \in N\}$. Observe that $\mathcal{P} = \mathcal{P}_1' \cup \mathcal{P}_1 \cup \mathcal{P}_2$ is a partition of R'. Let us even prove that $\mathcal{P}$ is an α-additive partition by constructing a maximum independent set of R' of size $\sum_{V' \in \mathcal{P}} \alpha(G[V'])$. For any $C_e, e \in M_1$, $e = \{v_i, v_{i+1}\}$, both v_i and v_{i+1} have type B, implying that $\alpha(G[C_e]) = \alpha(H_{v_i}) + \alpha(H_{v_{i+1}}) - 1$, and thus we consider a maximum independent set S_e of $G[C_e]$ that uses v_i and not v_{i+1}. For any $C_e, e \in M_2$, $e = \{v_i, v_{i+1}\}$, v_i has type B and v_{i+1} has type A, implying that $\alpha(G[C_e]) = \alpha(H_{v_i}) + \alpha(H_{v_{i+1}})$, and thus we consider a maximum independent set S_e of $G[C_e]$ that uses v_i and not v_{i+1}. For any $C_e, e \in M_3$, we consider a maximum independent set S_e of $G[C_e]$ such that $S_e \cap V(T') = \emptyset$. For $H_v, v \in N$, we consider a maximum independent set S_v of $G[H_v]$ such that $S_v \cap V(T') = \emptyset$ if $v \neq v_t$, and $S_v \cap V(T') = \{v_t\}$ otherwise. Observe that by definition $\bigcup_{e \in M_1 \cup M_2 \cup M_3} S_e \cup \bigcup_{v \in N} S_v$ is an independent set of R' of the claimed size.

Now, as by hypothesis neither Meta-Rule 1 nor Meta-Rule 2 can be applied to $((G, X, k), T')$, none of the T'-conflict structures of type 1 of $\mathcal{P}_1$ is $(|X| + 2)$-almost-free, and none of the T'-conflict structures of type 2 of $\mathcal{P}_1$ is $(|X| + 1)$-almost-free, thus not $(|X| + 2))$-almost-free either. Thus, for any $C_e \in \mathcal{P}_1$, C_e is not $(|X| + 2)$-almost-free. By Lemma 6.11, this implies that $|\mathcal{P}_1| \leq |\mathcal{X}| \cdot (|X| + 2)$. As $|\mathcal{P}| = \mathcal{O}(|\mathcal{P}_1|)$ and $|V(T')| = \mathcal{O}(|\mathcal{P}|)$, we get the desired result. ◀

6.3.2 Bounding the degree and the number of leaves

We define another meta-rule based on the two previous ones. Informally, in this meta-rule we move v to the modulator, apply rules to decrease the number of connected components, and move v back to R.

▶ **Meta-Rule 3.**
Input: *An input (G, X, k) of* IS/c-bd-mod *and a tree of bridges T of a connected component R' of R.*
Action: *If there exists $v \in V(T)$ such that $d_T(v) > 3|\mathcal{X}| \cdot |X|$ then*

- *apply exhaustively Rule 1 and Rule 2 to $(G, X \cup \{v\}, k)$, getting an instance $(G', X \cup \{v\}, k')$, and*

- *define the new instance as* (G', X, k').

▶ **Lemma 6.14.** *The following hold:*
- *Meta-Rule 3 is safe,*
- X *is still a* c-bd-*modulator in* G' *(implying that* (G', X, k') *is indeed an instance of* IS/c-bd-mod*), and*
- $|V(G')| < |V(G)|$.

Proof. The safeness of this meta-rule follows directly from the safeness of Rule 1 and Rule 2.

Let us now prove the two other claims. Recall that, by definition of R, we have the partition $V(G) = R \cup X$. Observe that as $v \in V(T)$ and T is a tree of bridges, $\#\text{CC}(G[R' - \{v\}]) = d_T(v)$, implying by hypothesis that $\#\text{CC}(G[R - \{v\}]) \geq \#\text{CC}(G[R' - \{v\}]) > 3|\mathcal{X}| \cdot |X|$. Notice also that in $(G, X \cup \{v\}, k)$, $X \cup \{v\}$ is still a c-bd-modulator. Moreover, the set of chunks in this instance is $\mathcal{X}' = \{X' \subseteq X \cup \{v\} \mid |X'| \leq 2^c\}$, implying that $|\mathcal{X}'| \leq 2|\mathcal{X}|$.

Choose $\tilde{R}$ such that we have the partition $V(G') = \tilde{R} \cup (X \cup \{v\})$. After applying exhaustively Rule 1 and Rule 2 to $(G, X \cup \{v\}, k)$, by Lemma 6.6 it follows that $\#\text{CC}(G'[\tilde{R}]) \leq |\mathcal{X}'| \cdot |X \cup \{v\}|$. As $|\mathcal{X}'| \cdot |X \cup \{v\}| \leq 2|\mathcal{X}| \cdot (|X| + 1) \leq 3|\mathcal{X}| \cdot |X|$, where we have assumed that $|X| \geq 2$ (as otherwise the problem can be solved in polynomial time), we deduce that at least one connected component of $G[R - \{v\}]$ has been deleted in order to obtain G', and thus that $|V(G')| < |V(G)|$. Moreover, by Lemma 6.6 it follows that G' is a subgraph of G, and that $\tilde{R}$ is a subset of $R - \{v\}$. This implies that $G'[\tilde{R} \cup \{v\}]$ is a subgraph of $G[R]$, and thus that X is still a c-bd-modulator in G'. ◀

▶ **Observation 6.15.** *If Meta-Rule 3 cannot be applied to* $((G, X, k), T)$*, then* $\Delta(T) \leq 3|\mathcal{X}| \cdot |X|$.

Let us now bound the number of leaves of T. The following simple lemma will be used in the counting argument in the proof of Lemma 6.21.

▶ **Lemma 6.16.** *Let* (G, X, k) *be an input of* IS/c-bd-mod*, let* T *be a tree of bridges of a connected component* R' *of* R*, and let* T_1 *be the tree obtained from* T *by removing any leaf* v *of type* A *whose parent has type* B*. Then* T_1 *has no* A*-leaf with* B*-parent,* $\Delta(T_1) \leq \Delta(T)$*, and* $\mathsf{diam}(T_1) \leq \mathsf{diam}(T)$.

Proof. Let v be a leaf of T of type A whose parent u has type B in T; we use the shortcuts A-leaf and B-parent. Observe that u still has type B in T_1. Indeed, as $H_u^{T_1} = H_u^T \cup H_v^T$, we get that any maximum independent set of $G[H_u^{T_1}]$ necessarily contains u. Thus, even if removing v creates a new leaf u in T_1, u cannot be a A-leaf with B-parent in T_1, implying that T_1 has no A-leaf with B-parent. This also implies the claimed bounds on $\Delta(T_1)$ and $\mathsf{diam}(T_1)$. ◀

In order to bound the number of leaves of T, we distinguish several types. Since our goal is to bound the size of T by a polynomial in $|X|$, we may assume that T contains at least three vertices, and therefore there exists a non-leaf vertex. In the following definition, we assume that the tree T is rooted at an arbitrary non-leaf vertex, so that we can speak about the parent of a leaf; the definition is easily seen to be oblivious to the choice of the root, since we only need to consider parents of leaves.

▶ **Definition 6.17.** *Given an input* (G, X, k) *of* IS/c-bd-mod *and a tree of bridges* T *of a connected component* R' *of* R*, let*
- $n_{*\mathsf{A}}(T) = |\{v \mid v$ *is a leaf of* T *whose parent has type* $A\}|$,
- $n_{\mathsf{AB}}(T) = |\{v \mid v$ *is a leaf of* T *of type* A *whose parent has type* $B\}|$,

- $n_{\mathsf{BB2}}(T) = |\{v \mid v$ *is a leaf of* T *of type* B *whose parent* u *has type* B *and is such that* $d_T(u) = 2\}|$,
- $n_{\mathsf{BBB}}(T) = |\{v \mid v$ *is a leaf of* T *of type* B *whose parent* u *has type* B *and is such that* $d_T(u) > 2$ *and there exists a leaf* $v' \neq v$ *of type* B *with the same parent* $u\}|$, *and*
- $n_{\mathsf{BBbad}}(T) = |\{v \mid v$ *is a leaf of* T *of type* B *whose parent* u *has type* B *and is such that* $d_T(u) > 2$ *and* u *is adjacent to a unique* B*-leaf*$\}|$.

Note that, since we may assume that T is rooted at a non-leaf vertex, the cases in Definition 6.17 are exhaustive, and each leaf of T is counted in exactly one set.

▶ **Lemma 6.18.** *Let* (G, X, k) *be an input of* IS/c-bd-mod, R' *be a connected component of* R, *and* T *be a lowering tree of* R'. *Suppose that Meta-Rule 1 cannot be applied to* $((G, X, k), T)$. *Then,* $n_{*\mathsf{A}}(T) \leq \Delta(T) \cdot |\mathcal{X}| \cdot (|X| + 2)$. *This implies that after Line 18 of Algorithm 1,* $n_{*\mathsf{A}}(T_1) = \mathcal{O}((|\mathcal{X}| \cdot |X|)^2)$.

Proof. Let us define an α-additive partition of R'. Let N_A be the set of vertices $v \in V(T)$ such that v has type A in T and has at least one leaf adjacent to it. For each $v \in N_A$, let Y_v be the set of leaves of T adjacent to v. Define $C_v = \{H_u \mid u \in Y_v\} \cup H_v$. Let $\mathcal{P}_1 = \{C_v \mid v \in N_A\}$ and $\mathcal{P}_2 = \{V_2\}$, where $V_2 = V(R') \setminus \bigcup_{V' \in \mathcal{P}_1} V'$. We claim that $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ is an α-additive partition, as in each C_v we can take a maximum independent set S_v of $G[C_v]$ such that S_v does not contain v. Thus, defining S' as a maximum independent set of $G[V_2]$, we get that $(\bigcup_{v \in N_A} S_v) \cup S'$ is a maximum independent set of R'.

Let us now prove that for any $v \in N_A$, C_v is not $(|X| + 2)$-almost-free. Let $v \in N_A$ and let $u \in Y_v$. As Meta-Rule 1 cannot be applied to $C = H_u \cup H_v$, we know that C is not $(|X| + 2)$-almost-free, implying that there exists $X' \in \mathcal{X}$ such that $\mathsf{conf}_C(X') \neq 0$ and $\mathsf{conf}_R(X') < |X| + 2$. However, as $\{C, C_v \setminus C\}$ is an α-additive partition of C_v, $\mathsf{conf}_C(X') \neq 0$ implies $\mathsf{conf}_{C_v}(X') \neq 0$, and thus that C_v is not $(|X| + 2)$-almost-free. By Lemma 6.11, this implies that $|\mathcal{P}_1| \leq |\mathcal{X}| \cdot (|X| + 2)$. As $n_{*\mathsf{A}}(T) \leq \Delta(T) \cdot |\mathcal{P}_1|$, we get the desired result. ◀

We now define a meta-rule to bound $n_{\mathsf{BBB}}(T)$.

▶ **Meta-Rule 4.**
Input: *An input* (G, X, k) *of* IS/c-bd-mod *and a tree of bridges* T *of a connected component* R' *of* R.
Action: *If there exists a* T*-conflict structure* C *of type* 3, *namely* $C = H_u$ *where* u *has type* B *and is a leaf of* T, *such that* C *is* $(|X| + 2)$*-almost-free, then remove vertex* u *and its parent* v *in* T *from the graph* G, *and define* $k' = k - 1$.

Proof of safeness. Let G' be the graph obtained after applying the rule. Let us first prove that $\alpha(G) \geq k$ implies that $\alpha(G') \geq k'$. Let S be an independent set of G with $|S| \geq k$. Then $S' = S \setminus \{u, v\}$ is an independent set of G' of size at least k'.

Let us now prove that $\alpha(G') \geq k'$ implies that $\alpha(G) \geq k$. Let S' be an independent set of G' with $|S'| \geq k'$. If $\mathsf{conf}_C^G(S'_X) = 0$ then there exists a maximum independent set Z of C such that $S'_X \cup Z$ is an independent set in G. Observe that replacing $(S' \cap (C \setminus u))$ by Z in S' gives an independent set S in G. Moreover, as u has type B we know $|(S' \setminus (C \setminus u))| \leq \alpha(G[C]) - 1$, so that $|S| \geq |S'| + 1$. Otherwise, if $\mathsf{conf}_C^G(S'_X) \neq 0$, according to Lemma 6.4, there exists $X' \in \mathcal{X}$ with $X' \subseteq S'_X$ such that $\mathsf{conf}_C^G(X') \neq 0$. As C is $(|X| + 2)$-almost-free, this implies that $\mathsf{conf}_R^G(X') \geq |X| + 2$. Observe that $\alpha(G' \setminus X) \geq \alpha(G \setminus X) - 1$, and that for any $X' \in \mathcal{X}$ we have $\alpha((G \setminus X) \setminus N_G(X')) \geq \alpha((G' \setminus X) \setminus N_G(X'))$. Thus, we can apply Lemma 6.12 with $\Delta_1 = 1$ and $\Delta_2 = 0$, and we get that $\mathsf{conf}_{V(G') \setminus X}^{G'}(X') \geq |X| + 1$. This implies that $|S'_{V(G') \setminus X}| \leq \alpha(G' \setminus X) - (|X| + 1) \leq \alpha(G \setminus X) - (|X| + 1)$, implying in turn that we can take for S any maximum independent set of $G[R] = G \setminus X$ as $\alpha(G[R]) \geq |S'| + 1$. ◀

▶ **Lemma 6.19.** *Let (G, X, k) be an input of* IS/c-bd-mod*, R' be a connected component of R, and T be a lowering tree of R'. Suppose that Meta-Rule 4 cannot be applied to $((G, X, k), T)$. Then, $n_{\mathsf{BBB}}(T) \leq \Delta(T) \cdot |\mathcal{X}| \cdot (|X| + 2)$. This implies that after Line 20 of Algorithm 1, $n_{\mathsf{BBB}}(T_1) = \mathcal{O}((|\mathcal{X}| \cdot |X|)^2)$.*

Proof. Let us define an α-additive partition of R'. Let N_B be the set of vertices $v \in V(T)$ such that v has type B in T and has at least two leaves of type B adjacent to it. For each $v \in N_B$, let Y_v be the set of B leaves of T adjacent to v. Define $C_v = \{H_u \mid u \in Y_v\} \cup H_v$. Let $\mathcal{P}_1 = \{C_v \mid v \in N_B\}$ and $\mathcal{P}_2 = \{V_2\}$, where $V_2 = V(R') \setminus \bigcup_{V' \in \mathcal{P}_1} V'$. We claim that $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ is an α-additive partition as in each C_v, any maximum independent set S_v of $G[C_v]$ does not contain v as there are at least two leaves of type B adjacent to v. Thus, defining S' as maximum independent set of $G[V_2]$, we get that $(\bigcup_{v \in N_B} S_v) \cup S'$ is a maximum independent set of R'.

Let us now prove that for any $v \in N_B$, C_v is not $(|X| + 2)$-almost-free. Let $v \in N_B$ and let $u \in Y_v$. As Meta-Rule 4 cannot be applied on $C = \{H_u\}$, we know that C is not $(|X| + 2)$-almost-free, implying that there exists $X' \in \mathcal{X}$ such that $\mathsf{conf}_C(X') \neq 0$ and $\mathsf{conf}_R(X') < |X| + 2$. However, as $\{C, C_v \setminus C\}$ is an α-additive partition of C_v, $\mathsf{conf}_C(X') \neq 0$ implies $\mathsf{conf}_{C_v}(X') \neq 0$, and thus that C_v is not $(|X| + 2)$-almost-free. By Lemma 6.11, this implies that $|\mathcal{P}_1| \leq |\mathcal{X}| \cdot (|X| + 2)$. As $n_{\mathsf{BBB}}(T) \leq \Delta(T) \cdot |\mathcal{P}_1|$, we get the desired result. ◀

Finally, our last meta-rule will be used to bound $n_{\mathsf{BB2}}(T)$.

▶ **Meta-Rule 5.**
Input: *An input (G, X, k) of* IS/c-bd-mod *and a tree of bridges T of a connected component R' of R.*
Action: *If there exists a T-conflict structure C of type 4, namely $C = H_{v_1} \cup H_{v_2}$ with a path (v_1, v_2, u) in T using notations of Definition 6.8, such that C is $(|X| + 1)$-almost-free, then identify v_1 and u, remove v_2, and define $k' = k - 1$.*

Proof of safeness. Let G' be the graph obtained after applying the meta-rule. We will prove the safeness of this meta-rule using the safeness of Meta-Rule 2. Let G_1 be the graph obtained from G by adding two vertices x, y, and two edges $\{x, y\}$, $\{y, v_1\}$. Let $R_1 = R' \cup \{x, y\}$. It is immediate that $(G_1, X, k + 1)$ is equivalent to (G, X, k). Now, as the two new edges are bridges in G_1, we get that $T_1 = T \cup \{\{x, y\}, \{y, v_1\}\}$ is a tree of bridges of R_1. As v_1 and v_2 still have type B in T_1, we get that C is a conflict structure of type 2 in T_1, and is still $(|X| + 1)$-almost-free. Thus, we can apply Meta-Rule 2 to $(G_1, X, k + 1)$ with T_1, and get an equivalent instance (G_2, X, k). Now, observe that x still has degree one in G_2, and thus by defining $G_3 = G_2 \setminus (\{x\} \cup N_{G_2}(x))$, we get that $(G_3, X, k - 1)$ is equivalent to (G_2, X, k). As $G_3 = G'$, we get the desired result. ◀

▶ **Lemma 6.20.** *Let (G, X, k) be an input of* IS/c-bd-mod*, R' be a connected component of R, and T be a lowering tree of R'. Suppose that Meta-Rule 5 cannot be applied to $((G, X, k), T)$. Then, $n_{\mathsf{BB2}}(T) \leq \Delta(T) \cdot |\mathcal{X}| \cdot (|X| + 1)$. This implies that after Line 22 of Algorithm 1, $n_{\mathsf{BB2}}(T_1) = \mathcal{O}((|\mathcal{X}| \cdot |X|)^2)$.*

Proof. The proof follows the same approach as those of Lemmas 6.18 and 6.19. The corresponding α-additive partition of R' used here is $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$, where $\mathcal{P}_1$ is the set of T-conflict structures of type 4 and $\mathcal{P}_2 = \{V_2\}$, where $V_2 = V(R') \setminus \bigcup_{V' \in \mathcal{P}_1} V'$. (We use the observation here that if T has at least four vertices, then any two distinct T-conflict structures of type 4 are vertex-disjoint, so that this indeed gives a valid partition. If T has at most three vertices, the lemma trivially holds.) ◀

Let us now prove that if no meta-rule can be applied to the subtree T_1 defined in Algorithm 1, then $|V(T)|$ is bounded.

▶ **Lemma 6.21.** *Let (G, X, k) be an input of* IS/c-bd-mod*, R' be a connected component of R, and T be a lowering tree of R'. Then, after Line 22 of Algorithm 1, it holds that $|V(T)| = \mathcal{O}(|\mathcal{X}|^5 \cdot |X|^5)$.*

Proof. Let T_1 be the tree obtained from T by removing any leaf v of type A whose parent has type B (thus, $n_{\mathsf{AB}}(T_1) = 0$ by Lemma 6.16). Let us root T_1 arbitrarily at a non-leaf vertex (since our intermediate goal is to bound the size of T_1 by a polynomial on $|\mathcal{X}|$ and $|X|$, we may assume that $|V(T_1)| \geq 3$, hence T_1 contains a non-leaf vertex). Let f be the number of leaves of T_1. It is sufficient to bound f, as $|V(T_1)| \leq f \cdot \mathsf{diam}(T_1)$. Let P be the set of type B vertices v of T_1 such that v is adjacent to exactly one B-leaf, and $d_{T_1}(v) > 2$. By the definition of $n_{\mathsf{BBbad}}(T_1)$, it holds that $|P| = n_{\mathsf{BBbad}}(T_1)$. Observe that we get $f = n_{*\mathsf{A}}(T_1) + n_{\mathsf{AB}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1) + n_{\mathsf{BBbad}}(T_1) = n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1) + |P|$, where we have used that $n_{\mathsf{AB}}(T_1) = 0$ and $n_{\mathsf{BBbad}}(T_1) = |P|$. Let us now bound $|P|$.

Let P' be the set of vertices of P which have no vertex of P as a proper descendant. Observe that $|P| \leq |P'| \cdot \mathsf{diam}(T_1)$, and thus it now remains to bound $|P'|$.

Let $p \in P'$ and let $Y_p = \{u_1, \ldots, u_{x_p}\}$ be the children of p in T_1. Observe that $x_p \geq 2$ as every vertex $v \in P$ satisfies $d_{T_1(v)} > 2$. As there is exactly one type B-leaf in Y_p, w.l.o.g. let us assume it is u_1. Then, for any $i \geq 2$, let T_p^i be the subtree of T_1 rooted at u_i. (Note that $|V(T_p^i)| \geq 2$ since T_1 has no leaf of type A with parent of type B.) By definition of P', T_p^i has no vertex of P. Therefore, for any two distinct vertices $p_1, p_2 \in P'$, and any two indices $i \in [2, x_{p_1}]$ and $j \in [2, x_{p_2}]$, it follows that $V(T_{p_1}^i) \cap V(T_{p_2}^j) = \emptyset$. Thus, for every $p \in P'$ and every $i \in [2, x_p]$, every leaf of T_p^i contributes at least one to $n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1)$. Since $x_p \geq 2$ for every $p \in P'$, every $p \in P'$ contributes at least one to $n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1)$. This implies that $|P'| \leq n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1)$. Thus, we have that

$$\begin{aligned}
|V(T_1)| &\leq (n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1) + |P|) \cdot \mathsf{diam}(T_1) \\
&\leq (n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1) + |P'| \cdot \mathsf{diam}(T_1)) \cdot \mathsf{diam}(T_1) \\
&\leq ((n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1)) \cdot (1 + \mathsf{diam}(T_1))) \cdot \mathsf{diam}(T_1) \\
&\leq 2((n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1)) \cdot \mathsf{diam}(T_1)^2 \\
&\leq 2((n_{*\mathsf{A}}(T_1) + n_{\mathsf{BBB}}(T_1) + n_{\mathsf{BB2}}(T_1)) \cdot \mathsf{diam}(T)^2,
\end{aligned}$$

where the last inequality follows from Observation 6.15. Applying Lemma 6.13, Lemma 6.18, Lemma 6.19, and Lemma 6.20 it follows that $|V(T_1)| = \mathcal{O}(|\mathcal{X}|^4 \cdot |X|^4)$. Finally, as $|V(T)| \leq \Delta(T) \cdot |V(T_1)|$ and $\Delta(T) = \mathcal{O}(|\mathcal{X}| \cdot |X|)$ by Observation 6.15, we obtain that $|V(T)| = \mathcal{O}(|\mathcal{X}|^5 \cdot |X|^5)$ and the lemma follows. ◀

6.4 Applying recursion

As now the size of the lowering tree of each connected component of R, and the number of connected components of R, are bounded by a polynomial of $|X|$, we can move all these lowering trees in the modulator and recurse to get the polynomial kernel for IS/c-bd-mod.

▶ **Theorem 6.22.** *Algorithm 1 is a polynomial kernel for* IS/c-bd-mod *of size $\mathcal{O}_c(|X|^{2^{f(c)}})$, where the constant hidden in $\mathcal{O}_c$ depends on c, and $f(c) = \sum_{i=5}^{c+4} i = \mathcal{O}(c^2)$.*

Proof. Observe first that Algorithm 1 is indeed polynomial for any fixed c as Lemma 6.3 and Lemma 6.9 imply that all rules and meta-rules can be applied in polynomial time. The

fact that the output is an equivalent instance immediately follows from the safeness of all rules and meta-rules. Let us now bound the size of the kernel by induction on c. For $c = 0$, $f(c) = 0$ and Algorithm 1 outputs X, which has the required size. Let $c \geq 1$. Let X_2 be defined as in Line 26 of Algorithm 1. Let $x = |X|$. Observe first that $|\mathcal{X}| = \mathcal{O}_c(x^{2^c})$. At Line 26 we know that $\#\textsc{cc}(R) \leq |\mathcal{X}| \cdot x$, and that for each connected component R' of R, its lowering tree T has $|V(T)| = \mathcal{O}(|\mathcal{X}|^5 \cdot x^5)$. This implies that at Line 26 we have $|X_1| = \mathcal{O}(|\mathcal{X}|^6 \cdot x^6) = \mathcal{O}_c(x^{2^{c+3}+6}) = \mathcal{O}_c(x^{2^{c+4}})$. Thus, the new $(c-1)$-bd-modulator X_2 has size $|X_2| = x + |X_1| = \mathcal{O}_c(x^{2^{c+4}})$. By induction hypothesis, Algorithm 1 with input $(G, X_2, k, c-1)$ will return an equivalent instance of size $\mathcal{O}_c(|X_2|^{2^{f(c-1)}})$, implying the claimed bound. ◀

The degree-bound of the kernel size bound of Theorem 6.22 is exponential in c. This is known to be unavoidable. Indeed, this can be proven directly from the fact that graphs of bridge-depth c have minimal blocking sets of size 2^c and are closed under disjoint union. Alternatively, an explicit lower bound is given in the literature [23, Thm. 2] that shows that Vertex Cover parameterized by a tree-depth-c modulator X does not admit a kernel with $(|X|^{2^{c-4}-\varepsilon})$ bits for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Since $\mathsf{bd}(G) \leq \mathsf{td}(G)$ for every graph G, the same lower bound holds for bridge-depth.

Theorem 6.22, together with the equivalence between Independent Set and Vertex Cover for structural parameterizations discussed in Section 2, proves the first implication of Theorem 1.1. Indeed, given a graph $G \in \mathcal{F}$ with bounded bridge-depth, Algorithm 1 will provide an equivalent instance (G', X', k') of IS/c-bd-mod of size $\mathcal{O}_c(|X|^{2^{f(c)}})$. As $G'[V(G') \setminus X']$ may not belong to $\mathcal{F}$, we move the entire graph to the modulator and say that $(G', V(G'), k')$ is an equivalent instance with $G'[V(G') \setminus X'] = \emptyset \in \mathcal{F}$.

▶ **Theorem 6.23.** *Let $\mathcal{F}$ be a minor-closed family of graphs. If $\mathcal{F}$ has bounded bridge-depth, then Vertex Cover parameterized by vertex-deletion distance to $\mathcal{F}$ has a polynomial kernelization.*

6.5 Negative result and proof of Theorem 1.1

The following negative result following easily from the results discussed so far, by using known gadgets.

▶ **Theorem 6.24.** *Let $\mathcal{F}$ be a minor-closed family of graphs with unbounded bridge-depth (or equivalently, by Theorem 1.2, with unbounded inclusion-minimal blocking sets). Then,* IS/dist-to-$\mathcal{F}$ *and VC/*dist-to-$\mathcal{F}$ *do not admit a polynomial kernel unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

Proof. Let $t \geq 1$. An *extended triangle path of length t* is a triangle-path of length t (see Definition 4.9) with two extra vertices u and v, and two extra edges $\{u, a_1\}$ and $\{b_t, v\}$. Let $\mathcal{F}^{\mathsf{etp}}$ be the family containing all graphs composed of disjoint unions of extended triangle paths. It is known from [17, Thm. 2] that IS/dist-to-$\mathcal{F}^{\mathsf{etp}}$ does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, not even when a modulator is given together with the input. (In fact, [17, Thm. 2] proves that IS parameterized by distance to mock forests does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, but their construction even produces graphs that are disjoint unions of extended triangle paths.) The result for IS is immediate as $\mathcal{F}^{\mathsf{tp}} \subseteq \mathcal{F}$ by Corollary 4.13 (recall that $\mathcal{F}^{\mathsf{tp}}$ is the family of all triangle-paths), and as any disjoint union of extended triangle paths is a minor of a sufficiently large triangle-path. The same lower bound for VC follows since there are parameter-preserving reductions in both ways. ◀

Theorem 6.23 and Theorem 6.24 together imply Theorem 1.1.

7 Conclusion

In this paper we introduced the graph parameter bridge-depth and used it to characterize the minor-closed graph classes $\mathcal{F}$ for which Vertex Cover parameterized by $\mathcal{F}$-modulator has a polynomial kernel. It would be interesting to see whether the characterization can be extended to subgraph-closed or even hereditary graph classes. If a characterization exists of the hereditary graph classes whose modulators lead to a polynomial kernel, it will likely not be as clean as Theorem 1.1: it will have to deal with the fact that bipartite graphs can be arbitrarily complex in terms of width parameters, while bipartite modulators allow for a polynomial kernel. Hence such a characterization has to capture parity conditions of $\mathcal{F}$.

A natural attempt to generalize our approach to deal with bipartite graphs is to consider the following parameter, which we call *bipartite-contraction-depth*: we mimic the definition of bridge-depth (cf. Definition 3.1), except that we redefine the graph G_{CB} to be the graph obtained from G by *simultaneously* contracting all edges that do *not* lie on an odd cycle. Note that bipartite-contraction-depth generalizes bridge-depth, in the sense that bridges do not lie on an odd cycle, and that the bipartite-contraction-depth of a graph with an odd cycle transversal of size k is at most $k+1$. Having defined this parameter, we would need, in order to obtain a statement similar to Theorem 4.14, that large bipartite-contraction-depth implies the existence of structures that allow to obtain kernel lower bounds, similarly to the fact that large bridge-depth implies the existence of large triangle-paths (cf. Corollary 4.13). The appropriate structure here seems to be an *odd-cycle-path of length* t, defined as a set of t vertex-disjoint odd cycles $C_1, \ldots, C_t$, and a set of $t-1$ vertex-disjoint paths (of any length) connecting C_i to C_{i+1} for $i \in [t-1]$, in such a way that for every $i \in \{2, \ldots, t-1\}$, the two attachment vertices in C_i are distinct. Now the expected property would be that large bipartite-contraction-depth forces long odd-cycle-paths. Unfortunately, this is not true. Indeed, consider the *Escher wall of size* h depicted in [33, Fig. 3]. It is proved in [33] that this graph does not contain two vertex-disjoint odd cycles, but a smallest hitting set for odd cycles has size h. Since there are no two vertex-disjoint cycles, a longest odd-cycle-path has length one. On the other hand, it can be easily verified that an Escher wall of size h has bipartite-contraction-depth $\Omega(h)$. Informally, this can be seen by noting that, initially, all edges lie on an odd cycle, hence a vertex removal is required, and that each such removal cascades in a constant number of contractions until all edges lie again on an odd cycle. Since a smallest hitting set for odd cycles of an Escher wall of size h has size h, the claimed bound follows. Therefore, summarizing this discussion, if one aims at a result similar to Theorem 1.1 that also applies to families $\mathcal{F}$ containing bipartite graphs, it seems that significant new ideas are required.

Another open research direction consists of a further algorithmic exploration of the merits of bridge-depth. We expect that several polynomial-space fixed-parameter tractable algorithms that work for graphs of bounded tree-depth [6, 32] can be extended to work with bridge-depth instead. Which other ways to enrich the recursive definition of tree-depth lead to novel algorithmic insights?

As for kernelization purposes, it is plausible that bridge-depth also characterizes the existence of polynomial kernels for other problems other than Vertex Cover, parameterized by the vertex-deletion distance of the input graph to a minor-closed graph class. For instance, the Feedback Vertex Set problem and the generalizations considered by Jansen and Pieterse [23] seem to be good candidates.

Acknowledgement. We would like to thank the anonymous reviewers for helpful comments

that improved the presentation of the manuscript.

References

1. Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.
2. Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.
3. Hans L. Bodlaender. Kernelization, exponential lower bounds. In *Encyclopedia of Algorithms*, pages 1013–1017. Springer, 2016. doi:10.1007/978-1-4939-2864-4_521.
4. Marin Bougeret and Ignasi Sau. How much does a treedepth modulator help to obtain polynomial kernels beyond sparse graphs? *Algorithmica*, 81(10):4043–4068, 2019. doi:10.1007/s00453-018-0468-8.
5. Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016. doi:10.1007/s00453-015-0045-3.
6. Li-Hsuan Chen, Felix Reidl, Peter Rossmanith, and Fernando Sánchez Villaamil. Width, depth, and space: Tradeoffs between branching and dynamic programming. *Algorithms*, 11(7):98, 2018. doi:10.3390/a11070098.
7. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
8. Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On the hardness of losing width. *Theory Comput. Syst.*, 54(1):73–82, 2014. doi:10.1007/s00224-013-9480-1.
9. Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014. doi:10.1145/2629620.
10. Reinhard Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, 5th edition, 2016.
11. Rod G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
12. Michael R. Fellows, Lars Jaffke, Aliz Izabella Király, Frances A. Rosamond, and Mathias Weller. What is known about vertex cover kernelization? In Hans-Joachim Böckenhauer, Dennis Komm, and Walter Unger, editors, *Adventures Between Lower Bounds and Higher Altitudes - Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday*, volume 11011 of *Lecture Notes in Computer Science*, pages 330–356. Springer, 2018. doi:10.1007/978-3-319-98355-4_19.
13. Michael R. Fellows, Bart M. P. Jansen, and Frances Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013. doi:10.1016/j.ejc.2012.04.008.
14. Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006. doi:10.1007/3-540-29953-X.
15. Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting forbidden minors: Approximation and kernelization. *SIAM J. Discrete Math.*, 30(1):383–410, 2016. doi:10.1137/140997889.
16. Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019. doi:10.1017/9781107415157.
17. Fedor V. Fomin and Torstein J. F. Strømme. Vertex cover structural parameterization revisited. In *Proc. 42nd WG*, volume 9941 of *LNCS*, pages 171–182, 2016. doi:10.1007/978-3-662-53536-3_15.
18. Eva-Maria C. Hols and Stefan Kratsch. Smaller parameters for vertex cover kernelization. In *Proc. 12th IPEC*, volume 89 of *LIPIcs*, pages 20:1–20:12, 2017. doi:10.4230/LIPIcs.IPEC.2017.20.

19 Eva-Maria C. Hols, Stefan Kratsch, and Astrid Pieterse. Elimination distances, blocking sets, and kernels for vertex cover. In *Proc. 37nd STACS*, volume 154 of *LIPIcs*, pages 36:1–36:14, 2020. doi:10.4230/LIPIcs.STACS.2020.36.

20 John E. Hopcroft and Robert Endre Tarjan. Efficient algorithms for graph manipulation (algorithm 447). *Commun. ACM*, 16(6):372–378, 1973. doi:10.1145/362248.362272.

21 Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited: Upper and lower bounds for a refined parameter. In *Proc. 28th STACS*, volume 9 of *LIPIcs*, pages 177–188, 2011. doi:10.4230/LIPIcs.STACS.2011.177.

22 Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013. doi:10.1007/s00224-012-9393-4.

23 Bart M. P. Jansen and Astrid Pieterse. Polynomial kernels for hitting forbidden minors under structural parameterizations. In *Proc. 26th ESA*, volume 112 of *LIPIcs*, pages 48:1–48:15, 2018. doi:10.4230/LIPIcs.ESA.2018.48.

24 Stefan Kratsch. A randomized polynomial kernelization for vertex cover with a smaller parameter. *SIAM J. Discrete Math.*, 32(3):1806–1839, 2018. doi:10.1137/16M1104585.

25 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *Proc. 53rd FOCS*, pages 450–459, 2012. doi:10.1109/FOCS.2012.46.

26 Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Kernelization - Preprocessing with a guarantee. In *The Multivariate Algorithmic Revolution and Beyond*, pages 129–161, 2012. doi:10.1007/978-3-642-30891-8_10.

27 Diptapriyo Majumdar, Venkatesh Raman, and Saket Saurabh. Polynomial kernels for vertex cover parameterized by small degree modulators. *Theory Comput. Syst.*, 62(8):1910–1951, 2018. doi:10.1007/s00224-018-9858-1.

28 George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: structural properties and algorithms. *Math. Program.*, 8:232–248, 1975. doi:10.1007/BF01580444.

29 Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

30 Jaroslav Nesetril and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *Eur. J. Comb.*, 27(6):1022–1041, 2006. doi:10.1016/j.ejc.2005.01.010.

31 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. doi:10.1093/acprof:oso/9780198566076.001.0001.

32 Michal Pilipczuk and Marcin Wrochna. On space efficiency of algorithms working on structural decompositions of graphs. *ACM TOCT*, 9(4):18:1–18:36, 2018. doi:10.1145/3154856.

33 Dieter Rautenbach and Bruce A. Reed. The Erdös-Pósa Property for Odd Cycles in Highly Connected Graphs. *Combinatorica*, 21(2):267–278, 2001. doi:10.1007/s004930100024.

34 Neil Robertson and Paul D. Seymour. Graph minors. V. Excluding a planar graph. *J. Comb. Theory, Ser. B*, 41(1):92 – 114, 1986. doi:10.1016/0095-8956(86)90030-4.

35 Neil Robertson and Paul D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004. doi:10.1016/j.jctb.2004.08.001.

36 Wouter Cames van Batenburg, Tony Huynh, Gwenaël Joret, and Jean-Florent Raymond. A tight Erdös-Pósa function for planar minors. In *Proc. 30th SODA*, pages 1485–1500. SIAM, 2019. doi:10.1137/1.9781611975482.