(Meta) Kernelization

HANS L. BODLAENDER, Utrecht University and Eindhoven University of Technology
FEDOR V. FOMIN and DANIEL LOKSHTANOV, University of Bergen
EELKO PENNINKX, Utrecht University
SAKET SAURABH, Institute of Mathematical Sciences and University of Bergen
DIMITRIOS M. THILIKOS, National & Kapodistrian University of Athens and AlGCo Project Team, CNRS, LIRMM

In a parameterized problem, every instance I comes with a positive integer k. The problem is said to admit a polynomial kernel if, in polynomial time, one can reduce the size of the instance I to a polynomial in k while preserving the answer. In this work, we give two meta-theorems on kernelization. The first theorem says that all problems expressible in counting monadic second-order logic and satisfying a coverability property admit a polynomial kernel on graphs of bounded genus. Our second result is that all problems that have finite integer index and satisfy a weaker coverability property admit a linear kernel on graphs of bounded genus. These theorems unify and extend all previously known kernelization results for planar graph problems.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.2 [**Graph Theory**]: Graph Algorithms

General Terms: Algorithms, Design, Theory

Additional Key Words and Phrases: Monadic second-order logic, parameterized complexity, embedded graphs, preprocessing, kernelization, treewidth, protrusions, finite integer index

ACM Reference Format:
Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. 2016. (Meta) kernelization. J. ACM 63, 5, Article 44 (November 2016), 69 pages.
DOI: http://dx.doi.org/10.1145/2973749

44

A preliminary version of this article appeared in Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS'09), IEEE, Los Alamitos, CA, 629–638. The first author was partially supported by the NETWORKS project, funded by the Netherlands Organization for Scientific Research NWO. The work of the second author was supported by the European Research Council (ERC) via grant Rigorous Theory of Preprocessing, reference 267959. The work of the fifth author was supported by funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant 306992. The work of the last author was co-financed by the E.U. European Social Fund and Greek national funds through the operational program Education and Lifelong Learning of the National Strategic Reference Framework Research Funding Program: "Thales. Investing in Knowledge Society Through the European Social Fund."
Authors' addresses: H. L. Bodlaender and E. Penninkx, Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, Netherlands; email: h.l.bodlaender@uu.nl, eelko@koningshuis.org; F. V. Fomin, D. Lokshtanov, Department of Informatics, University of Bergen, N-5020 Bergen, Norway; email: {fomin | daniello}@ii.uib.no; S. Saurabh, Institute of Mathematical Sciences, Chennai 600113, India; email: saket@imsc.res.in; D. M. Thilikos, Department of Mathematics, National and Kapodistrian University of Athens, Panepistimioupolis, GR-15784, Athens, Greece; email: sedthilk@thilikos.info.

DOI: http://dx.doi.org/10.1145/2973749

1. INTRODUCTION

Preprocessing (data reduction or kernelization) as a strategy of coping with hard problems is universally used in almost every implementation. The history of preprocessing, like applying reduction rules to simplify truth functions, can be traced back to the 1950s [Quine 1952]. A natural question in this regard is how to measure the quality of the preprocessing rules proposed for a specific problem. For a long time, the mathematical analysis of polynomial time preprocessing algorithms was neglected. The basic reason for this anomaly was that if we start with an instance I of an NP-hard problem and can show that, in polynomial time, we can replace this with an equivalent instance I' with $|I'| < |I|$ that then would imply $\mathsf{P} = \mathsf{NP}$ in classical complexity. The situation changed drastically with advent of parameterized complexity. Combining tools from parameterized and classical complexities, it has become possible to derive upper and lower bounds on the sizes of reduced instances, or so-called kernels.

Kernelization. In parameterized complexity, each problem instance comes with a parameter k, and the parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm (the degree of polynomial is independent of k), called a *kernelization* algorithm, that reduces the input instance down to an instance with size bounded by a polynomial $p(k)$ in k while preserving the answer. This reduced instance is called a $p(k)$ *kernel* for the problem. If $p(k) = O(k)$, then we call it a *linear kernel* (for a more formal definition, see Section 2.1.1). Kernelization has been extensively studied in the realm of parameterized complexity, resulting in polynomial kernels for a variety of problems. Notable examples of kernelization include a $2k$-sized vertex kernel for Vertex Cover [Chen et al. 2001]; a $355k$ vertex kernel for Dominating Set on planar graphs [Alber et al. 2004], which later was improved to a $67k$ vertex kernel [Chen et al. 2007]; and an $O(k^2)$ kernel for Feedback Vertex Set [Thomassé 2010] parameterized by the solution size.

One of the most important results in the area of kernelization was given by Alber et al. [2004]. They gave the first linear-size kernel for the Dominating Set problem on planar graphs. Their work triggered an explosion of papers on kernelization, particularly on kernelization of problems on planar graphs. Combining the ideas of Alber et al. [2004] with problem-specific data reduction rules, kernels of linear sizes were obtained for a variety of parameterized problems on planar graphs, including Connected Vertex Cover, Minimum Edge Dominating Set, Maximum Triangle Packing, Efficient Edge Dominating Set, Induced Matching, Full-Degree Spanning Tree, Feedback Vertex Set, Cycle Packing, and Connected Dominating Set [Alber et al. 2004, 2006a; Bodlaender and Penninkx 2008; Bodlaender et al. 2008; Chen et al. 2007; Guo and Niedermeier 2007b; Guo et al. 2010; Kanj et al. 2011; Lokshtanov et al. 2011; Moser and Sikdar 2009]. Dominating Set has received special attention from kernelization view point, leading to a linear kernel on graphs of bounded genus [Fomin and Thilikos 2004] and a polynomial kernel on graphs excluding a fixed graph H as a minor and on d-degenerated graphs [Alon and Gutner 2008; Philip et al. 2012]. We refer to several surveys [Guo and Niedermeier 2007a; Fomin and Saurabh 2014; Lokshtanov et al. 2012] and books [Cygan et al. 2015; Downey and Fellows 2013; Flum and Grohe 2006; Niedermeier 2006] for a detailed treatment of the area of kernelization.

Most of the works on linear kernels on planar graphs have the following idea in common: find an appropriate region decomposition (essentially a partitioning of the vertex set into graphs of small diameter) of the input planar graph based on the problem in question, then perform problem-specific rules to reduce the part of the graph inside each region. The first step toward the general abstraction of all of these algorithms was initiated by Guo and Niedermeier [2007b], who proved a general decomposition theorem for all problems with a specific distance property. Combining this decomposition theorem with problem-specific reduction rules yields linear kernels for various problems on planar graphs. Thus, all previous work on kernelization was strongly based on the design of reduction rules particular to the problem in question. In this article, we step aside and find properties of problems, such as expressibility in counting monadic second-order (CMSO) logic, which allows these reduction rules to be automated.

Algebraic reduction techniques. The idea of graph replacement for algorithms dates back to Fellows and Langston [1989]. Arnborg et al. [1993] proved that every set of graphs of bounded treewidth that is definable by a monadic second-order (MSO) logic formula is also definable by reduction. By making use of algebraic reductions, Arnborg et al. [1993] obtained a linear time algorithm for MSO-expressible problems on graphs of bounded treewidth. Bodlaender and de Fluiter [1996], Bodlaender and van Antwerpen-de Fluiter [2001], and de Fluiter [1997] generalized these ideas in several ways—in particular, they applied it to several optimization problems. It is also important to mention the work of Bodlaender and Hagerup [1998], who used the concept of graph reduction to obtain parallel algorithms for MSO-expressible problems on graphs of bounded treewidth.

Algorithmic meta-theorems. Our results can be seen as what Grohe and Kreutzer call *algorithmic meta-theorems* [Grohe 2007; Kreutzer 2011]. Meta-theorems bring out the deep relations between logic and combinatorial structures, which is a fundamental issue of computational complexity. Such theorems also yield a better understanding of the scope of general algorithmic techniques and the limits of tractability. A typical example of meta-theorem is the celebrated Courcelle's theorem [Courcelle 1992], which states that all graph properties definable in MSO can be decided in linear time on graphs of bounded treewidth. More recent examples of such meta-theorems state that all first-order definable properties on planar graphs can be decided in linear time [Frick and Grohe 2001] and that all first-order definable optimization problems on classes of graphs with excluded minors can be approximated in polynomial time to any given approximation ratio [Dawar et al. 2007]. Our meta-theorems not only give a uniform and natural explanation for a large family of known kernelization results but also provide a variety of new results. In what follows, we build up toward our theorems. We first give necessary definitions needed to formulate our results.

Parameterized graph problems. A parameterized graph problem Π in general can be seen as a subset of $\Sigma^* \times \mathbb{Z}^+$ where, in each instance (x, k) of Π, x encodes a graph and k is the parameter (we denote by $\mathbb{Z}^+$ the set of all nonnegative integers). In this article, we extend this definition by permitting the parameter k to be negative with the additional constraint that either all pairs with nonpositive value of the parameter are in Π or that no such pair is in Π. Formally, a parametrized problem Π is a subset of $\Sigma^* \times \mathbb{Z}$ where for all $(x_1, k_1), (x_2, k_2) \in \Sigma^* \times \mathbb{Z}$ with $k_1, k_2 < 0$ it holds that $(x_1, k_1) \in \Pi$ if and only if $(x_2, k_2) \in \Pi$. This extended definition encompasses the traditional one and is being adopted for technical reasons (see Section 2.3). In many cases, in the pair (x, k), x will encode an *annotated graph*—that is, a pair (G, S), where S is a subset of the vertices of G (i.e., S contains the *annotated* vertices of G). In this article, we mostly work on problems restricted to certain graph classes. For this reason, given a graph class $\mathcal{G}$, we use notation $\Pi \Cap \mathcal{G}$ for the set of instances of Π minus the instances (x, k), where x does not encode a graph in $\mathcal{G}$. That way, the new problem $\Pi' = \Pi \Cap \mathcal{G}$ is a subset of $\Sigma^* \times \mathbb{Z}$ that corresponds to the restriction of Π to graphs in $\mathcal{G}$. In this work, we mostly apply such restrictions to bounded genus graphs. We denote by $\mathcal{G}_r$ the class of graphs that are 2-cell embeddable in some surface of Euler genus at most r.

r-coverable problems. Let $G = (V, E)$ be a graph embedded without crossings in a surface. (For more details on graph embeddings, see Section 6.) The *radial distance* between two vertices x, y of G in this embedding is one less than the minimum length of an alternating sequence of vertices and faces starting from x and ending in y such that every two consecutive elements of this sequence are incident with each other. Given a set $S \subseteq V$, we define $\mathbf{R}^r_G(S)$ to be the set of all vertices of G whose radial distance from some vertex of S is at most r.

Let r be a nonnegative integer. We say that a parameterized graph problem Π has the *radial r-coverability property* if all YES-instances (G, k), $G = (V, E)$, of Π encode graphs embeddable in some surface of Euler genus at most r and there exist such an embedding and a set $S \subseteq V$ such that $|S| \le r \cdot k$ and $\mathbf{R}^r_G(S) = V$. We say that a problem Π is *radially r-coverable* if either Π or its "complement in $\mathcal{G}_r$," namely $\overline{\Pi} \cap \mathcal{G}_r$, has the radial r-coverability property (here, $\overline{\Pi} = \Sigma^* \backslash \Pi$). Every problem Π that has the radial r-coverability property is radially r-coverable. However, the converse is not necessarily true. In particular, the p-INDEPENDENT SET problem can easily be seen to be radially r-coverable, but it does not have the radial r-coverability property.

r-quasi-coverable problems. A parameterized graph problem Π has the *radial r-quasi-coverability property* if all YES-instances of Π encode graphs embeddable in some surface of Euler genus at most r and there exist such an embedding and a set $S \subseteq V$ such that $|S| \le r \cdot k$ and $\mathbf{tw}(G \backslash \mathbf{R}^r_G(S)) \le r$ (by $\mathbf{tw}(G)$, we denote the treewidth of G; for the formal definition, see Section 2.1.2). We say that a problem Π is *radially r-quasi-coverable* if either Π or $\overline{\Pi} \cap \mathcal{G}_r$ has the radial r-quasi-coverability property. Every problem Π that has the radial r-quasi-coverability property is radially r-quasi-coverable. Again, the converse is not necessarily true. For example, the p-CYCLE PACKING problem is radially r-quasi-coverable, but it does not have the radial r-quasi-coverability property.

Thus, for a coverable problem, we are able to cover the whole graph with $O(k)$ balls of constant radius, whereas in a quasi-coverable one, we can cover with $O(k)$ balls of constant radius an "essential" part of the graph. Of course, if a problem is r-coverable, then it is also r-quasi-coverable. From now on, for simplicity, we drop the terms *radial* and *radially* and simply use the terms *r-quasi-coverability property* or *r-quasi-coverable*.

CMSO Logic. We use CMSO logic [Arnborg et al. 1991; Courcelle 1990, 1997], an extension of MSO logic, as a basic tool to express properties of vertex/edge sets in graphs. Considering that in this section our aim is to define a series of CMSO-based problem properties, we avoid the formal definitions of CMSO and postpone them for Section 2.6.

Our first result concerns a parameterized analogue of graph optimization problems where the objective is to find a maximum- or minimum-size vertex or edge set satisfying a CMSO-expressible property. We now define a class of parameterized problems, called p-MIN-CMSO *problems*,[1] with one problem for each CMSO sentence ψ on graphs, where ψ has a free vertex set variable S. The p-MIN-CMSO problem defined by ψ is denoted by p-MIN-CMSO$[\psi]$ and defined as follows.

> p-MIN-CMSO$[\psi]$
> *Input*: A graph $G = (V, E)$ and an integer k
> *Parameter*: k
> *Question*: Is there a subset $S \subseteq V$ such that $|S| \le k$ and $(G, S) \models \psi$?

[1] We follow the notation given in the book by Flum and Grohe [2006] and add "p" in front of names of problems to emphasize that these are parameterized problems.

In other words, p-MIN-CMSO[ψ] is a subset Π of $\Sigma^* \times \mathbb{Z}$ where for every $(x, k) \in \Sigma^* \times \mathbb{Z}^+$, $(x, k) \in \Pi$ if and only if there exists a set $S \subseteq V$ where $|S| \leq k$ such that the graph G encoded by x together with S satisfy ψ (i.e., $(G, S) \models \psi$). For $(x, k) \in \Sigma^* \times \mathbb{Z}^-$, we know that $(x, k) \notin \Pi$. In this case, we say that Π *is definable by the sentence* ψ and that Π is a p-MIN-CMSO[ψ].

The definition of a p-EQ-CMSO[ψ] (respectively, p-MAX-CMSO[ψ]) problem is the same as the one for a p-MIN-CMSO[ψ] problem with the difference that now we ask that $|S| = k$ (respectively, $|S| \geq k$) and that for any $(x, k) \in \Sigma^* \times \mathbb{Z}^-$ we have that $(x, k) \in \Pi$. We can also extend the notion of a p-MIN/EQ/MAX-CMSO[ψ] problems to edge versions. In these problems, S is a subset of edges instead of a subset of vertices. All of our results can be straightforwardly extended to this alternate setting. In particular, an edge set problem on graph $G = (V, E)$ can be transformed to a vertex subset problem on the edge-vertex incidence graph $I(G)$ of G, which is is a bipartite graph with vertex bipartitions V and E with edges between vertices $v \in V$ and $e \in E$ if and only if v is incident with e in G. Observe that if G can be embedded in surface Σ, then so does $I(G)$, and even the treewidth of these graphs only differ by a factor of 2. To make the translation work throughout the article, it is sufficient to use the fact that the property of being an incidence graph of a graph G is expressible in MSO. To avoid complications in our proof, we omit the details for this.

The *annotated version* Π^α of a p-MIN/EQ/MAX-CMSO[ψ] problem Π is the parameterized graph problem whose instances are pairs of the form $((G, Y), k)$, where (G, Y) is an annotated graph and k is a nonnegative integer. In the *annotated version* of a p-MIN/EQ-CMSO[ψ] problem, S is additionally required to be a subset of Y. For the annotated version of a p-MAX-CMSO[ψ] problem, S is not required to be a subset of Y, but instead of $|S| \geq k$, we demand that $|S \cap Y| \geq k$. A problem is an *annotated* p-MIN/EQ/MAX-CMSO[ψ] *problem* if it is the annotated version of some p-MIN/EQ/MAX-CMSO[ψ] problem.

Our results. Our first result is the following theorem (the proofs of Theorems 1.1, 1.2, and 1.3 are given in Section 4).

THEOREM 1.1. *If* Π *is an r-coverable* p-MIN/MAX-CMSO[ψ] (*respectively,* p-EQ-CMSO[ψ]) *problem, then the annotated version* Π^α *admits a quadratic (respectively, cubic) kernel.*

Let us remark that although a parameterized graph problem is a special case of its annotated version where all vertices are annotated, the existence of a polynomial kernel for the annotated version does not imply directly that the corresponding (nonannotated) parameterized graph problem admits a polynomial kernel. Indeed, a polynomial kernelization for an annotated parameterized graph problem Π^α is a polynomial time algorithm that, given an input $(G = (V, E), Y, k)$ of Π^α, computes an equivalent instance $(G' = (V', E'), Y', k')$ of Π^α such that $\max\{|V'|, k'\} = k^{O(1)}$. The point here is that even when $Y = V$, we cannot guarantee that $Y' = V'$. However, there is a simple trick resolving this issue, given some additional complexity conditions. In particular, Theorem 1.1 can be used to prove the following.

THEOREM 1.2. *If* Π *is an* NP*-hard r-coverable* p-MIN/EQ/MAX-CMSO[ψ] *problem and* Π^α *is in* NP, *then* Π *admits a polynomial kernel.*

Theorems 1.1 and 1.2 provide polynomial kernels for a variety of parameterized graph problems. However, many parameterized graph problems in the literature are known to admit linear kernels on planar graphs. Our next theorem unifies and generalizes all known linear kernels for parametrized graph problems on surfaces. To this end, we make use of the notion of having *finite integer Index* (FII). This term first appeared in the works of Bodlaender and van Antwerpen-de Fluiter [2001] and de Fluiter [1997] and is similar to the notion of *finite state* [Abrahamson and Fellows 1993; Borie et al. 1992; Courcelle 1990]. As the definition of the property of having FII is long, we defer it to Section 2.3. Out next result is the following.

THEOREM 1.3. *If* Π *is an r-quasi-coverable parameterized graph problem that has FII, then* Π *admits a linear kernel.*

Our theorems are similar in spirit, yet they have a few differences. In particular, not every p-MIN/EQ/MAX-CMSO[ψ] problem has FII. For example, the INDEPENDENT DOMINATING SET problem is a p-MIN-CMSO[ψ] problem, but it does not have FII. In addition, the class of parameterized graph problems that have FII does not have a syntactic characterization, and hence it may take some more work to apply Theorem 1.3 than Theorem 1.1. On the other hand, Theorem 1.3 applies to r-quasi-coverable problems and yields linear kernels. That way, it unifies and implies results presented in Alber et al. [2004, 2006b], Bodlaender and Penninkx [2008], Bodlaender et al. [2008], Chen et al. [2007], Fomin and Thilikos [2004], Guo and Niedermeier [2007b], Guo et al. [2010], Kanj et al. [2011], Lokshtanov et al. [2011], and Moser and Sikdar [2009] as a corollary.

At high level, the proofs of our theorems consist of combinatorial decomposition and algebraic reductions. The combinatorial part shows how a graph can be decomposed into pieces with specific properties, and the algebraic reductions part explains how these pieces can be reduced. The important tool in both parts is the notion of *protrusion*—that is, a subset of vertices of a graph, inducing a graph of constant treewidth and separated from the remaining part of the graph by a constant number of vertices. In the algebraic reductions part of the proof, we show that sufficiently large protrusions can be replaced by equivalent protrusions of smaller size. For CMSO problems, the algebraic reduction step is much more technical and involved than for FII. Here we work with annotated problems and perform replacements in several stages.

In the combinatorial part, the result concerning quasi-coverable problems is roughly as follows. Suppose that after deleting k constant radius balls from a bounded genus graph G the remaining part of G has constant treewidth. Then either G has a protrusion of sufficiently large size (in which case we can apply protrusion reduction to reduce the instance) or G has $O(k)$ vertices. The proof of this result is based on a new treewidth obstruction lemma for graphs embedded on a surface of bounded genus, which is interesting in its own right. More precisely, the lemma states that if a graph of bounded genus has two vertices that are far apart (in the radial distance) and cannot be separated by a small separator, then the treewidth of the graph is large. Concerning coverable problems, we show that every bounded genus graph G whose vertices can be covered by k balls of constant radius admits a *protrusion decomposition*. A protrusion decomposition is a partition of the vertex set into $O(k)$ sets: one of these sets is a set S of size $O(k)$, and the other sets are protrusions separated from each other by S. Combined with protrusion replacement rules for CMSO problems, such a decomposition implies the existence of a polynomial kernel for every coverable CMSO problem.

The remainder of this article is organized as follows. In Section 2, we give a series of definitions on basic notions that are necessary to describe our results. In Section 3, we give a proof of a variant of the classical Courcelle's theorem, which we use in the proofs of our results. In Section 4, we present our meta-algorithmic framework for kernelization and explain how our main results are derived from a series of algorithmic and combinatorial properties. The algorithmic properties are proved in Section 5, whereas our combinatorial results are proven in Section 6. Some criterion for proving that a problem in graphs has FII are given in Section 7, and in Section 8 we give an extended exposition of how our results can be applied to concrete problems. In Section 9, we conclude with some open problems and further research directions. At the end of the article, we append a short compendium of problems for which linear or polynomial kernels are consequences of our results.

2. DEFINITIONS AND NOTATIONS

In this section, we give necessary definitions, set up notations, and derive some preliminary results that we make use of in proving the main results of the article.

2.1. Preliminaries

We now define some concepts that we use in the rest of this work. Given a graph $G = (V, E)$, we use the notation $V(G)$ and $E(G)$ for V and E, respectively. Given a set $S \subseteq V(G)$, we define $\partial_G(S)$ as the set of vertices in S that have a neighbor in $V \setminus S$. For a set $S \subseteq V(G)$, the *neighborhood* of S in G is $N_G(S) = \partial_G(V(G)\setminus S)$. We also define the *closed neighborhood of S in G* as $N_G[S] = S \cup \partial_G(V(G)\setminus S)$. When it is clear from the context, we omit the subscripts.

Let $G = (V, E)$ be a graph. A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. The subgraph G' is called an *induced subgraph* of G if $E' = \{\{u, v\} \in E \mid u, v \in V'\}$. In this case, G' is also called the subgraph *induced by* V' and is denoted by $G[V']$. Given a graph G and a set $S \subseteq V$, we denote by $G\setminus S$ the graph $G[V\setminus S]$. If $S \subseteq E$, we denote $G\setminus S = (V, E\setminus S)$. We also use the term (x, y)-path for a path in G that has x and y as endpoints.

Throughout this article, we use $\mathbb{Z}$, $\mathbb{Z}^+$, and $\mathbb{Z}^-$ for the sets of integers, nonnegative integers, and nonpositive integers, respectively. Finally, we use $\mathbb{N}$ for the set of positive integers.

2.1.1. Parameterized Algorithms and Kernels. An instance of a parameterized problem consists of (x, k), where k is called the *parameter*. Thus, a parameterized problem Π is a subset of $\Sigma^* \times \mathbb{Z}$ for some finite alphabet Σ such that for all $(x_1, k_1), (x_2, k_2) \in \Sigma^* \times \mathbb{Z}$ with $k_1, k_2 < 0$, it holds that $(x_1, k_1) \in \Pi \iff (x_2, k_2) \in \Pi$. A central notion in parameterized complexity is *fixed parameter tractability*, which means, for a given instance (x, k), solvability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size. The notion of *kernelization* is formally defined as follows.

Definition 2.1 *[Kernelization].* Let $\Pi \subseteq \Sigma^* \times \mathbb{Z}$ be a parameterized problem and g be a computable function. We say that Π *admits a kernel of size g* if there exists an algorithm $\mathcal{K}$, called the *kernelization algorithm* (or, in short, a *kernelization)* that, given $(x, k) \in \Sigma^* \times \mathbb{Z}^+$, outputs, in time polynomial in $|x| + k$, a pair $(x', k') \in \Sigma^* \times \mathbb{Z}^+$ such that

(a) $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$ and
(b) $\max\{|x'|, k'\} \leq g(k)$.

For every $(x, k) \in \Sigma^* \times \mathbb{Z}^-$, the algorithm outputs a trivial equivalent instance. When $g(k) = k^{O(1)}$ or $g(k) = O(k)$, we say that Π *admits a polynomial* or *linear kernel,* respectively.

In this article, we study parameterized problems on graphs. However, in many cases, we have to deal with annotated graph problems whose input is a pair (G, S), where S is a set of annotated vertices of G. For such problems, the task is to find a solution that is contained in S. For this reason, we use the term *parameterized graph problem* for every subset Π of $\Sigma^* \times \mathbb{Z}$, where in each instance $I = (x, k) \in \Sigma^* \times \mathbb{Z}$ the string x is encoding either a graph $G = (V, E)$ or a pair (G, S) with $S \subseteq V$ and the integer k encodes the parameter.

2.1.2. Treewidth. Let $G = (V, E)$ be a graph. A *tree decomposition* of G is a pair $(T, \mathcal{X} = \{X_t\}_{t\in V(T)})$, where T is a tree and $\mathcal{X}$ is a collection of subsets of V such that

—$\forall e = \{u, v\} \in E, \exists t \in V(T) : \{u, v\} \subseteq X_t$ and
—$\forall v \in V,\ T[\{t \mid v \in X_t\}]$ is nonempty and connected.

We call the vertices of T *nodes* and the sets in $\mathcal{X}$ *bags* of the tree decomposition $(T, \mathcal{X})$. The *width* of $(T, \mathcal{X})$ is equal to $\max\{|X_t| - 1 \mid t \in V(T)\}$, and the *treewidth* of $G = (V, E)$ is the minimum width over all tree decompositions of G. We denote the treewidth of a graph G by $\mathbf{tw}(G)$.

A *nice tree decomposition* is a triple $(T, \mathcal{X}, r)$, where $(T, \mathcal{X})$ is a tree decomposition in which the tree T is rooted on some vertex $r \in V(T)$ and the following conditions are satisfied:

—every node of the tree T has at most two children;
—if a node t has two children t_1 and t_2, then $X_t = X_{t_1} = X_{t_2}$ (we call t a *join node*); and
—if a node t has one child t_1, then either $|X_t| = |X_{t_1}| + 1$ and $X_{t_1} \subset X_t$ (in this case, we call t_1 the *introduce node*) or $|X_t| = |X_{t_1}| - 1$ and $X_t \subset X_{t_1}$ (in this case, we call t_1 the *forget node*).

It is possible to transform a given tree decomposition $(T, \mathcal{X})$ into a nice tree decomposition $(T', \mathcal{X}', r)$, where the root r is any vertex of T in time $O(|V| + |E|)$ [Bodlaender 1996].

2.2. Boundaried Graphs

Here we define the notion of *boundaried graphs* and various operations on them.

Definition 2.2 *[Boundaried Graphs].* A boundaried graph is a graph G with a set $B \subseteq V(G)$ of distinguished vertices and an injective labeling λ from B to the set $\mathbb{Z}^+$. The set B is called the *boundary* of G, and the vertices in B are called *boundary vertices* or *terminals*. Given a boundaried graph G, we denote its boundary by $\delta(G)$, denote its labeling by λ_G, and define its *label set* by $\Lambda(G) = \{\lambda_G(v) \mid v \in \delta(G)\}$. Given a finite set $I \subseteq \mathbb{Z}^+$, we define $\mathcal{F}_I$ to denote the class of all boundaried graphs whose label set is I. Similarly, we define $\mathcal{F}_{\subseteq I} = \bigcup_{I' \subseteq I} \mathcal{F}_{I'}$. We also denote by $\mathcal{F}$ the class of all boundaried graphs. Finally, we say that a boundaried graph is a *t-boundaried* graph if $\Lambda(G) \subseteq \{1, \ldots, t\}$.

Definition 2.3 *[Gluing by $\oplus$].* Let G_1 and G_2 be two boundaried graphs. We denote by $G_1 \oplus G_2$ the graph (not boundaried) obtained by taking the disjoint union of G_1 and G_2 and identifying equally labeled vertices of the boundaries of G_1 and G_2. In $G_1 \oplus G_2$, there is an edge between two labeled vertices if there is either an edge between them in G_1 or in G_2.

Definition 2.4. Let $G = G_1 \oplus G_2$, where G_1 and G_2 are boundaried graphs. We define the *glued* set of G_i as the set $B_i = \lambda_{G_i}^{-1}(\Lambda(G_1) \cap \Lambda(G_2)), i = 1, 2$. For a vertex $v \in V(G_1)$, we define its *heir* $h(v)$ in G as follows: if $v \notin B_1$, then $h(v) = v$; otherwise, $h(v)$ is the result of the identification of v with an equally labeled vertex in G_2. The *heir* of a vertex in G_2 is defined symmetrically. The *common boundary* of G_1 and G_2 in G is equal to $h(B_1) = h(B_2)$, where the evaluation of h on vertex sets is defined in the obvious way. The *heir* of an edge $\{u, v\} \in E(G_i)$ is the edge $\{h(u), h(v)\}$ in G.

Let $\mathcal{G}$ be a class of (not boundaried) graphs. By slightly abusing notation, we say that a boundaried graph *belongs to a graph class* $\mathcal{G}$ if the underlying graph belongs to $\mathcal{G}$.

2.3. Finite Integer Index

Definition 2.5 *[Canonical Equivalence on Boundaried Graphs].* Let Π be a parameterized graph problem whose instances are pairs of the form (G, k). Given two boundaried graphs $G_1, G_2 \in \mathcal{F}$, we say that $G_1 \equiv_\Pi G_2$ if $\Lambda(G_1) = \Lambda(G_2)$ and there exist a *transposition constantc* $\in \mathbb{Z}$ such that

$$\forall(F, k) \in \mathcal{F} \times \mathbb{Z} \quad (G_1 \oplus F, k) \in \Pi \Leftrightarrow (G_2 \oplus F, k + c) \in \Pi.$$

Note that the relation $\equiv_\Pi$ is an equivalence relation. Observe that c could be negative in the preceding definition. This is the reason we extended the definition of parameterized problems to include negative parameters.

Next we define a notion of "transposition-minimality" for the members of each equivalence class of $\equiv_\Pi$.

Definition 2.6 *[Progressive Representatives].* Let Π be a parameterized graph problem whose instances are pairs of the form (G, k), and let $\mathcal{C}$ be some equivalence class of $\equiv_\Pi$. We say that $J \in \mathcal{C}$ is a *progressive representative* of $\mathcal{C}$ if for every $H \in \mathcal{C}$ there exists $c \in \mathbb{Z}^-$ such that

$$\forall(F, k) \in \mathcal{F} \times \mathbb{Z} \quad (H \oplus F, k) \in \Pi \Leftrightarrow (J \oplus F, k + c) \in \Pi. \tag{1}$$

The following lemma guaranties the existence of a progressive representative for each equivalence class of $\equiv_\Pi$.

Lemma 2.7. *Let Π be a parameterized graph problem whose instances are pairs of the form (G, k). Then each equivalence class of $\equiv_\Pi$ has a progressive representative.*

Proof. We first examine the case in which every instance of Π with a negative valued parameter is a NO-instance.

Let $\mathcal{C}$ be an equivalence class of $\equiv_\Pi$. We distinguish the following two cases.

Case 1. Suppose first that for every $H \in \mathcal{C}$, every $F \in \mathcal{F}$, and every integer $k \in \mathbb{Z}$, it holds that $(H \oplus F, k) \notin \Pi$. Then we set J to be an arbitrary chosen graph in $\mathcal{C}$ and $c = 0$. In this case, it is obvious that (1) holds for every $(F, k) \in \mathcal{F} \times \mathbb{Z}$.

Case 2. Suppose now that for some $H_0 \in \mathcal{C}$, $F_0 \in \mathcal{F}$, and $k_0 \in \mathbb{Z}$, it holds that that $(H_0 \oplus F_0, k_0) \in \Pi$. Among all such triples, choose the one where the value of k_0 is minimized. Since every instance of Π with a negative-valued parameter is a NO-instance, it follows that k_0 is well defined and nonnegative. We claim that H_0 is a progressive representative.

Let $H \in \mathcal{C}$. As $H_0 \equiv_\Pi H$, there is a constant c such that

$$\forall(F, k) \in \mathcal{F} \times \mathbb{Z} \quad (H \oplus F, k) \in \Pi \Leftrightarrow (H_0 \oplus F, k + c) \in \Pi.$$

It suffices to prove that $c \leq 0$. Assume for a contradiction that $c > 0$. Then by taking $k = k_0 - c$ and $F = F_0$, we have that

$$(H \oplus F_0, k_0 - c) \in \Pi \Leftrightarrow (H_0 \oplus F_0, k_0 - c + c) \in \Pi.$$

Since $(H_0 \oplus F_0, k_0) \in \Pi$, it follows that $(H \oplus F_0, k_0 - c) \in \Pi$, contradicting the choice of H_0, F_0, k_0.

Suppose now that every instance of Π with a negative-valued parameter is a YES-instance. The proof of this case is symmetric to the previous one: just replace every occurrence of "$\in \Pi$" with a "$\notin \Pi$" and every occurrence of "$\notin \Pi$" with "$\in \Pi$" and the "NO-instance" with "YES-instance." □

Notice that two boundaried graphs with different label sets belong to different equivalence classes of $\equiv_\Pi$. Hence, for every equivalence class $\mathcal{C}$ of $\equiv_\Pi$, there exists some finite set $I \subseteq \mathbb{Z}^+$ such that $\mathcal{C} \subseteq \mathcal{F}_I$. We are now in position to give the following definition.

Definition 2.8 *[Finite Integer Index].* A parameterized graph problem Π whose instances are pairs of the form (G, k) has *finite integer index* (or simply has *FII*) if and only if for every finite $I \subseteq \mathbb{Z}^+$, the number of equivalence classes of $\equiv_\Pi$ that are subsets of $\mathcal{F}_I$ is finite. For each $I \subseteq \mathbb{Z}^+$, we define $\mathcal{S}_I$ to be a set containing exactly one progressive representative of each equivalence class of $\equiv_\Pi$ that is a subset of $\mathcal{F}_I$. We also define $\mathcal{S}_{\subseteq I} = \bigcup_{I' \subseteq I} \mathcal{S}_{I'}$.

2.4. An Alternate Way to Define Extended Formulation of a Parameterized Problem

Parameterized problems usually have been defined as subsets of $\Sigma^* \times \mathbb{Z}^+$. In other words, a parameterized problem Π is a subset of $\Sigma^* \times \mathbb{Z}^+$. However, in this article, we define a parameterized problem Π to be a subset of $\Sigma^* \times \mathbb{Z}$, thus allowing negative parameters. An alternate route to obtain all results in this work without changing the classical notion of a parameterized problem would be to define an extension of a parameterized problem as follows. For $\Pi \subseteq \Sigma^* \times \mathbb{Z}^+$, we say that $\Pi_{\mathsf{ext}} \subseteq \Sigma^* \times \mathbb{Z}$ is an *extension* of Π if

—for all $k \geq 0$, $(x, k) \in \Pi_{\mathsf{ext}}$ if and only if $(x, k) \in \Pi$, and
—for all $(x_1, k_1), (x_2, k_2) \in \Sigma^* \times \mathbb{Z}$ with $k_1, k_2 < 0$, it holds that $(x_1, k_1) \in \Pi$ if and only if $(x_2, k_2) \in \Pi$.

Observe that a parameterized problem $\Pi \subseteq \Sigma^* \times \mathbb{Z}^+$ has two extensions based on whether all (x, k), $k < 0$, is a NO-instance or a YES-instance. For an extension Π_{ext}, we could now use the definition of FII used in this article. For a parameterized problem Π (i.e., a subset of $\Sigma^* \times \mathbb{Z}^+$), we say that Π has FII if at least one of the two extensions of Π has FII. These simple modifications will allow us to work with the traditional definition of a parameterized problem. However, for the clarity of presentation and to avoid going between Π and Π_{ext} throughout the article, we decided to modify the definition of a parameterized problem to allow negative parameters.

2.5. Structures and Its Properties

We first define the notions of *structure* and *arity of a structure*.

Definition 2.9 *[Structure and Arity].* A *structure* is a tuple where the first element of the tuple is a graph G and the remaining elements of the tuple are either subsets of V, subsets of E, vertices in G, or edges in G. The arity of the structure is the number of elements in the tuple.

Given a structure α of arity p and an integer $i \in \{1, \ldots, p\}$, we let $\alpha[i]$ denote the i-th element of α. The graph of a structure α is denoted by G_α, and it appears as the first element of the structure—that is, $G_\alpha = \alpha[1]$. *Appending* a subset S of $V(G_\alpha)$ to a structure α of arity p produces a new structure, denoted by $\alpha' = \alpha \diamond S$, of arity $p+1$ with the first p elements of α' being the elements of α and $\alpha'[p+1] = S$. Appending an edge set, a vertex, or an edge to a structure is defined similarly. For example, consider the structure $\alpha = (G_\alpha, S, e)$ of arity 3, where $S \subseteq V(G_\alpha)$ and $e \in E(G_\alpha)$. In addition, let S' be some subset of $V(G_\alpha)$, and let $u \in V(G_\alpha)$. Appending S' to α results in the structure $\alpha' = \alpha \diamond S' = (G_\alpha, S, e, S')$, whereas appending u to α' results in the structure $\alpha'' = \alpha' \diamond u = (G_\alpha, S, e, S', u)$.

Next we define the notions of *type* of a structure and *property* of structures.

Definition 2.10 *[Type of Structure].* The type of a structure of arity p is another tuple of arity p, denoted by $\mathbf{type}(\alpha)$, where the first element $\mathbf{type}(\alpha)[1]$ is graph, whereas for every $i \in \{2, \ldots, p\}$, $\mathbf{type}(\alpha)[i]$ is vertex, edge, vertex set, or edge set according to what the i-th element of α is. Note that we distinguish between a set containing a single vertex or edge from just a single vertex or edge.

Definition 2.11 *[Properties of Structures].* A *property* of structures is a function σ that assigns to each structure a value in {true, false}.

2.6. CMSO Logic and Its Properties

The syntax of MSO logic of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$; variables for vertices, edges, sets of vertices, and sets of edges; the quantifiers $\forall$, $\exists$ that can be applied to these variables; and the following five binary relations:

(1) $u \in U$, where u is a vertex variable and U is a vertex set variable;
(2) $d \in D$, where d is an edge variable and D is an edge set variable;
(3) $\mathbf{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident with the vertex u;
(4) $\mathbf{adj}(u, v)$, where u and v are vertex variables and the interpretation is that u and v are adjacent; and
(5) equality of variables representing vertices, edges, sets of vertices, and sets of edges.

In addition to the usual features of MSO logic, if we have atomic sentences testing whether the cardinality of a set is equal to q modulo r, where q and r are integers such that $0 \leq q < r$ and $r \geq 2$, then this extension of MSO logic is called *CMSO logic*. Thus, CMSO is MSO with the following atomic sentence for a set S:

$$\mathbf{card}_{q,r}(S) = \mathbf{true} \text{ if and only if } |S| \equiv q \pmod{r}.$$

We refer to Arnborg et al. [1991] and Courcelle [1990, 1997] for a detailed introduction on CMSO.

A CMSO sentence ψ where some of the variables are free can be evaluated on a structure α by instantiating the free variables of ψ by the elements of α. To determine which variables of ψ are instantiated by which elements of α, we need to introduce some conventions.

In a CMSO sentence ψ, each free variable x has a *rank* $r_x \in \mathbb{N}\setminus\{1\}$ associated to it. Thus, a CMSO sentence ψ can be seen as a string accompanied by a tuple of integers containing one integer r_x for each free variable x of ψ.

We say that $\mathbf{type}(\alpha)$ *matches* ψ if the arity of α is at least $\max r_x$, where the maximum is taken over each free variable x of ψ and for each free variable x of ψ, $\mathbf{type}(\alpha)[r_x]$ corresponds to the kind of the variable x. For example, if x is a vertex set variable, then $\mathbf{type}(\alpha)[r_x] =$ vertex set. Finally, we say that α *matches* ψ if $\mathbf{type}(\alpha)$ matches ψ. For each free variable x of ψ and a structure α that matches ψ, the *corresponding element* of x in α is $\alpha[r_x]$.

Definition 2.12 *[Property σ_ψ].* Each CMSO sentence ψ defines a property σ_ψ on structures as follows. For every structure α that does not match ψ, the value of $\sigma_\psi(\alpha)$ is equal to false; otherwise, the value of $\sigma_\psi(\alpha)$ is the result of the evaluation of ψ with each free variable x of ψ instantiated by $\alpha[r_x]$.

Note that it is not necessary that every element of α corresponds to some variable of ψ. However, it is still possible that the sentence ψ can be evaluated on the structure α; in this case, the evaluation of the sentence does not depend on all elements of the structure.

A property σ is *CMSO definable* if there exists a sentence ψ such that $\sigma = \sigma_\psi$. In this case, we say that the CMSO sentence ψ *defines* σ.

OBSERVATION 1. *For every CMSO-definable property σ, there exists a CMSO sentence ψ that defines σ and has the following additional features:*

(*1*) *each variable of* ψ *has a unique name,*
(*2*) ψ *does not use the* **adj** *operator,*
(*3*) ψ *does not have conjunctions, and*
(*4*) ψ *does not have universal quantifiers.*

PROOF. Let ψ' be a CMSO sentence defining σ. We construct another CMSO sentence ψ defining σ so that ψ satisfies Properties (1) through (4). For Property (1), we rename each variable so that it has a unique name. When we rename a free variable x of ψ of rank r_x to x', we let x' have rank $r_{x'} = r_x$ in ψ'.

For Property (2), we replace each occurrence of $\mathbf{adj}(x, x')$ by $\exists x'' \in E : \mathbf{inc}(x'', x) \wedge \mathbf{inc}(x'', x')$. For Properties (3) and (4), just use the fact that $\wedge$ and $\forall$ can be expressed using $\vee$, $\exists$, and $\neg$ by De Morgan's laws. □

We call CMSO sentences satisfying Properties (1) through (4) of Observation 1 *normalized* CMSO sentences.

2.7. Boundaried Structures

In this subsection we extend the notion of boundaried graphs to boundaried structures.

Definition 2.13 *[Boundaried Structure].* A *boundaried structure* is a tuple where the first element is a boundaried graph G and the remaining elements are either subsets of $V(G)$, subsets of $E(G)$, vertices in $V(G)$, edges in $E(G)$, or the symbol $\star$. For a boundaried structure α, $\alpha[i]$ is the i-th element of α and $G_\alpha = \alpha[1]$ is always a boundaried graph.

Definition 2.14 *[Type of a Boundaried Structure].* The *type* of a boundaried structure is defined similarly to the type of a structure; for a boundaried structure α of arity p, $\mathbf{type}(\alpha)$ is a tuple of arity p, where the first element of $\mathbf{type}(\alpha)$ is boundaried graph, whereas for every $i \in \{2, \ldots, p\}$, $\mathbf{type}(\alpha)[i]$ is vertex, edge, $\star$, vertex set, or edge set according to what $\alpha[i]$ is.

Definition 2.15 *[Type Matching].* Given a CMSO formula ψ, we say that $\mathbf{type}(\alpha)$ *matches* ψ if the arity of α is at least $\max r_x$, where the maximum is taken over each free variable x of ψ and for every free variable x of ψ

—if x is a vertex variable, then $\mathbf{type}(\alpha)[r_x] \in \{\star, \text{vertex}\}$,
—if x is a edge variable, then $\mathbf{type}(\alpha)[r_x] \in \{\star, \text{edge}\}$,
—if x is a vertex set variable, then $\mathbf{type}(\alpha)[r_x] = \text{vertex set}$, and
—if x is a edge set variable, then $\mathbf{type}(\alpha)[r_x] = \text{edge set}$.

We say that α matches ψ if $\mathbf{type}(\alpha)$ *matches* ψ.

We denote by $\mathcal{A}$ the set of all boundaried structures. Given some $p \in \mathbb{N}$, we denote by $\mathcal{A}^p$ the set of all boundaried structures of arity p, and given a finite set $I \subseteq \mathbb{Z}^+$, we denote by $\mathcal{A}_I^p$ the set of all boundaried structures of arity p whose boundaried graph has label set I. Notice that according to this definition, $\mathcal{A}_I^1$ is essentially the same as $\mathcal{F}_I$. Finally, we say that a boundaried structure α is a *t-boundaried structure* if $\Lambda(G_\alpha) \subseteq \{1, \ldots, t\}$.

Definition 2.16 *[Compatibility].* For two boundaried structures α and β, we say that α and β are *compatible*. We denote this by $\alpha \sim_c \beta$ if the following conditions are satisfied:

—α and β have the same arity p.
—For every $i \leq p$, $\mathbf{type}(\alpha)[i] = \mathbf{type}(\beta)[i] \neq \star$ *or* exactly one out of $\mathbf{type}(\alpha)[i]$, $\mathbf{type}(\beta)[i]$ is a vertex or edge and exactly one of them is a $\star$.

—For every $i \in \{2, \ldots, p\}$ such that both $\alpha[i]$ and $\beta[i]$ are vertices, $\alpha[i] \in \delta(G_\alpha)$, $\beta[i] \in \delta(G_\beta)$ and $\lambda_{G_\alpha}(\alpha[i]) = \lambda_{G_\beta}(\beta[i])$.
—For every i such that both $\alpha[i]$ and $\beta[i]$ are edges, $\alpha[i] \in E(G_\alpha[\delta(G_\alpha)])$, $\beta[i] \in E(G_\beta[\delta(G_\beta)])$ and $\lambda_{G_\alpha}(\alpha[i]) = \lambda_{G_\beta}(\beta[i])$ (here we extend the function λ to sets in the obvious way).

Definition 2.17 *[Gluing of Boundaried Compatible Structures].* When two boundaried structures α and β are *compatible*, the operation of *gluing* α and β is defined as follows:

—$\alpha \oplus \beta$ is a structure γ with the same arity, say p, as α and β.
—$G_\gamma = G_\alpha \oplus G_\beta$.
—For every $i \in \{2, \ldots, p\}$ such that both $\alpha[i]$ and $\beta[i]$ are both vertex sets or both edge sets, we define $\gamma[i] = h(\alpha[i]) \cup h(\beta[i])$.
—For every $i \in \{2, \ldots, p\}$ such that both $\alpha[i]$ and $\beta[i]$ are vertices or both are edges, we have $h(\alpha[i]) = h(\beta[i])$ (by compatibility), and we set $\gamma[i] = h(\alpha[i]) = h(\beta[i])$. If $\alpha[i] = \star$, we set $\gamma[i] = h(\beta[i])$, whereas if $\beta[i] = \star$, we set $\gamma[i] = h(\alpha[i])$. By compatibility, exactly one of these cases applies for every i.

3. A VARIANT OF COURCELLE'S THEOREM

In this section, we give a proof of a variant of the classical Courcelle's theorem [Courcelle 1990, 1992, 1997] (see also Courcelle and Engelfriet [2012], which we use in the proofs of our results).

We define the *compatibility equivalence* relation $\equiv_c$ on boundaried structures as follows. We say that $\alpha \equiv_c \beta$ if for every boundaried structure γ,

$$\alpha \sim_c \gamma \iff \beta \sim_c \gamma.$$

Clearly, $\equiv_c$ is an equivalence relation. We now make the following observation.

OBSERVATION 2. *For every arity p and finite set $I \subseteq \mathbb{Z}^+$, the relation $\equiv_c$ has a finite number of equivalence classes when restricted to $\mathcal{A}_I^p$.*

PROOF. Define the *compatibility signature* of a boundaried structure α to be a string $\mathbf{s}(\alpha)$ that encodes the following information about α:

—$\Lambda(G_\alpha)$.
—**type**(α).
—For every i such that $\alpha[i]$ is a vertex, $\mathbf{s}(\alpha)$ encodes whether $\alpha[i] \in \delta(G_\alpha)$, and if so, it encodes $\lambda_{G_\alpha}(\alpha[i])$.
—For every i such that $\alpha[i]$ is an edge, $\mathbf{s}(\alpha)$ encodes whether $\alpha[i] \in E(G_\alpha[\delta(G_\alpha)])$, and if so, it also encodes $\lambda_{G_\alpha}(\alpha[i])$.

Clearly, for every fixed I and p, the compatibility signature $\mathbf{s}(\alpha)$ can be encoded by a number of bits that depends only on I and p, and hence there are only finitely many different compatibility signatures for boundaried structures in $\mathcal{A}_I^p$. It is easy to verify whether a boundaried structure $\alpha \in \mathcal{A}_I^p$ is compatible with a boundaried structure $\gamma \in \mathcal{A}^p$ can be deduced solely from γ and the compatibility signature of α. Thus, if two boundaried structures α and β have the same compatibility signatures, then $\alpha \equiv_c \beta$. This completes the proof. □

Definition 3.1 *[Canonical Equivalence on Structures].* For a property σ of structures, we define the corresponding *canonical equivalence relation* $\equiv_\sigma$ on boundaried structures. For two boundaried structures α and β we say that $\alpha \equiv_\sigma \beta$ if $\alpha \equiv_c \beta$ and for all boundaried structures γ compatible to α (and thus also to β) we have

$$\sigma(\alpha \oplus \gamma) = \mathsf{true} \Leftrightarrow \sigma(\beta \oplus \gamma) = \mathsf{true}.$$

It is easy to verify that $\equiv_\sigma$ is an equivalence relation. We say that a property σ of structures is *finite state* if, for every $p \in \mathbb{N}$ and $I \subseteq \mathbb{Z}^+$, the equivalence relation $\equiv_\sigma$ has a finite number of equivalence classes when restricted to $\mathcal{A}_I^p$. Given a CMSO sentence ψ, we say that $\equiv_{\sigma_\psi}$ is the *canonical equivalence relation* corresponding to ψ, and we simply denote this relation by $\equiv_\psi$.

In our arguments, the following lemma will be crucial. Although it is an implicit consequence of the results [Arnborg et al. 1991; Courcelle 1990, 1997, 1992; Abrahamson and Fellows 1993; Borie et al. 1992; Downey and Fellows 1998], in the rest of this section we give a complete and self-contained proof.

LEMMA 3.2. *Every CMSO-definable property on structures has finite state.*

PROOF. Our aim is to prove that for every $p \in \mathbb{N}$ and finite $I \subseteq \mathbb{Z}^+$, and CMSO-definable property σ, the equivalence relation $\equiv_\sigma$ has a finite number of equivalence classes when restricted to $\mathcal{A}_I^p$. For this, we will define, for every normalized CMSO sentence ψ, a function sgn_ψ that takes as input a boundaried structure and outputs a string in $\{0, 1\}^*$. To prove the result, it suffices to show the following two properties of the function sgn_ψ:

- (**i**) For all $p \in \mathbb{N}$, $J \subseteq \mathbb{Z}^+$, the set $\mathsf{sgn}_\psi(\mathcal{A}_I^p)$ is finite.
- (**ii**) For every two boundaried structures α and β, if $\mathsf{sgn}_\psi(\alpha) = \mathsf{sgn}_\psi(\beta)$, then $\alpha \equiv_\sigma \beta$.

We need the following claim:

Decoder Claim: To prove Property (**ii**), it is enough to prove that for every CMSO sentence ψ defining a property σ, there exist two functions,

$$\begin{aligned} \mathsf{dec}_c &: \{0, 1\}^* \times \mathcal{A}^p \rightarrow \{\mathsf{true}, \mathsf{false}\} \\ \mathsf{dec}_\psi &: \{0, 1\}^* \times \mathcal{A}^p \rightarrow \{\mathsf{true}, \mathsf{false}\}, \end{aligned}$$

such that for every pair $\alpha \in \mathcal{A}_I^p$ and $\gamma \in \mathcal{A}^p$, we have that

$$\mathsf{dec}_c(\mathsf{sgn}_\psi(\alpha), \gamma) = \mathsf{true} \iff \alpha \sim_c \gamma, \tag{2}$$

and for every pair $\alpha \in \mathcal{A}_I^p$ and $\gamma \in \mathcal{A}^p$ with $\alpha \sim_c \gamma$, it holds that

$$\mathsf{dec}_\psi(\mathsf{sgn}_\psi(\alpha), \gamma) = \mathsf{true} \iff \sigma(\alpha \oplus \gamma) = \mathsf{true}. \tag{3}$$

PROOF OF DECODER CLAIM: For the proof of the preceding claim, assume that for some $\alpha, \beta \in \mathcal{A}_I^p$, it holds that

$$\mathsf{sgn}_\psi(\alpha) = \mathsf{sgn}_\psi(\beta). \tag{4}$$

Then for all $\gamma \in \mathcal{A}^p$, it holds that

$$\alpha \sim_c \gamma \Leftrightarrow^{(2)} \mathsf{dec}_c(\mathsf{sgn}_\psi(\alpha), \gamma) = \mathsf{true} \Leftrightarrow^{(4)} \mathsf{dec}_c(\mathsf{sgn}_\psi(\beta), \gamma) = \mathsf{true} \Leftrightarrow^{(2)} \beta \sim_c \gamma,$$

and hence $\alpha \equiv_c \beta$. Further, for all $\gamma \in \mathcal{A}^p$ such that $\alpha \sim_c \gamma$, it holds that

$$\begin{aligned} \sigma(\alpha \oplus \gamma) = \mathsf{true} &\Leftrightarrow^{(3)} \mathsf{dec}_\psi(\mathsf{sgn}_\psi(\alpha), \gamma) = \mathsf{true} \\ &\Leftrightarrow^{(4)} \mathsf{dec}_\psi(\mathsf{sgn}_\psi(\beta), \gamma) = \mathsf{true} \\ &\Leftrightarrow^{(3)} \sigma(\beta \oplus \gamma) = \mathsf{true}, \end{aligned}$$

and thus $\alpha \equiv_\sigma \beta$, as required. This completes the proof of the decoder claim.

We start by partially defining the outputs of sgn_ψ as follows. If α does not match ψ, then $\mathsf{sgn}_\psi(\alpha)$ is the null string, denoted by ϵ; otherwise, sgn_ψ encodes the compatibility signature of α (as defined in the proof of Observation 2) and additional information about α that will be specified later in the proof.

The existence of a function dec_c satisfying (2) follows directly from the proof of Observation 2.

We define the function dec_ψ such that $\mathsf{dec}_\psi(\epsilon, \gamma) = \mathsf{false}$ for every boundaried structure γ. In addition, $\mathsf{dec}_\psi(\mathsf{sgn}_\psi(\alpha), \gamma) = \mathsf{false}$ whenever $\mathbf{type}(\alpha \oplus \gamma)$ does not match ψ. Observe that this can be checked using the compatibility signature of α (that is already encoded in $\mathsf{sgn}_\psi(\alpha)$) and γ. Thus, dec_ψ satisfies (3) for all pairs α, γ such that $\alpha \oplus \gamma$ does not match ψ.

In the remainder of the proof, we will complete the definition of sgn_ψ and will define dec_ψ for all pairs $\mathsf{sgn}_\psi(\alpha)$, γ such that $\alpha \oplus \gamma$ match ψ. This should be done in a way such that (**i**) holds for sgn_ψ and (3) holds for dec_ψ.

We now define sgn_ψ and dec_ψ and prove that they have the claimed properties for the case where α matches ψ and ψ is an atomic CMSO sentence. An atomic CMSO sentence is a sentence of the form "$u \in S$," "$e \in S$," "$u = v$," "$e = d$," "$\mathbf{inc}(d, u)$," or "$\mathbf{card}_{q,r}(S)$," where S is a set variable, u and v are vertex variables, e and d are edge variables, and $r \in \mathbb{N}\backslash\{1\}$ and $q \in \{0, \ldots, r-1\}$. In this case, we append to $\mathsf{sgn}_\psi(\alpha)$ certain information about α that

(i) encodes $G[\delta(G_\alpha)]$,
(ii) encodes λ_{G_α},
(iii) for every vertex variable x encodes whether $\alpha[r_x] = \star$ or not (recall that r_x is the rank of x), and if $\alpha[r_x] \neq \star$, then $\mathsf{sgn}_\psi(\alpha)$ encodes whether $\alpha[r_x] \in \delta(G_\alpha)$, and, if this is the case, also encodes $\lambda_{G_\alpha}(\alpha[r_x])$,
(iv) for every edge variable x encodes whether $\alpha[r_x] = \star$ or not, and if $\alpha[r_x] \neq \star$, $\mathsf{sgn}_\psi(\alpha)$ also encodes whether $\alpha[r_x] \subseteq \delta(G_\alpha)$, and, if this is the case, also encodes $\lambda_{G_\alpha}(\alpha[r_x])$,
(v) for every vertex set variable x encodes $\lambda_{G_\alpha}(\alpha[r_x] \cap \delta(G_\alpha))$,
(vi) for every edge set variable x encodes $\lambda_{G_\alpha}(\alpha[r_x] \cap E(\delta(G_\alpha)))$ (here λ_{G_α} is extended to sets of unordered pairs in the natural way),
(vii) for every vertex variable x, such that $\alpha[r_x] \neq \star$, and every vertex set variable x', encodes whether $\alpha[r_x] \in \alpha[r_{x'}]$,
(viii) for every edge variable x, such that $\alpha[r_x] \neq \star$, and every edge set variable x', encodes whether $\alpha[r_x] \in \alpha[r_{x'}]$,
(ix) for every pair of vertex variables x, x', where $\alpha[r_x] \neq \star \neq \alpha[r_{x'}]$, encodes whether $\{\alpha[r_x], \alpha[r_{x'}]\} \in E(G_\alpha)$,
(x) for every vertex variable x and every edge variable x', where $\alpha[r_x] \neq \star \neq \alpha[r_{x'}]$, encodes whether $\alpha[r_x] \in \alpha[r_{x'}]$ (i.e., whether $\alpha[r_{x'}]$ is incident to $\alpha[r_x]$),
(xi) if ψ is "$\mathbf{card}_{q,r}(x)$," where x is either a vertex set or an edge set variable, encodes $|\alpha[r_x]| \pmod r$,
(xii) for every pair of vertex variables x, x', where $\alpha[r_x] \neq \star \neq \alpha[r_{x'}]$, encodes whether $\alpha[r_x] = \alpha[r_{x'}]$, and
(xiii) for every pair of edge variables x, x', where $\alpha[r_x] \neq \star \neq \alpha[r_{x'}]$, encodes whether $\alpha[r_x] = \alpha[r_{x'}]$.

To see that $\mathsf{sgn}_\psi(\alpha)$ satisfies Property (**i**), it is enough to verify that for every $\alpha \in \mathcal{A}_I^p$, the length of $\mathsf{sgn}_\psi(\alpha)$ is upper bounded by a function depending only the atomic formula ψ, the integer p, and the set I.

Table I. Procedure of Case 1 in the Proof of Lemma 3.2

if $\alpha[r_x] \neq \star$ *(using the compatibility signature of* α*)*
  then if $\alpha[r_x] \in \alpha[r_{x'}]$ *(using* (vii)*)*
    then return true
    else if $\alpha[r_x] \in \delta(G_\alpha)$ *(using* (iii)*)*
      then if $\lambda_{G_\gamma}^{-1}(\lambda_{G_\alpha}(\alpha[r_x])) \in \gamma[r_{x'}]$ *(using* (iii)*)*
        then return true
        else return false
      else return false
  else if $\gamma[r_x] \in \gamma[r_{x'}]$ *(notice that* $\gamma[r_x] \neq \star$*, since* $\alpha \sim_c \gamma$*)*
    then return true
    else if $\gamma[r_x] \in \delta(G_\gamma)$
      then if $\lambda_{G_\alpha}^{-1}(\lambda_{G_\gamma}(\gamma[r_x])) \in \alpha[r_{x'}]$ *(using* (iii) *and* (v)*)*
        then return true
        else return false
      else return false

We now define $\mathsf{dec}_\psi(\mathsf{sgn}_\psi(\alpha), \gamma)$ for the case where ψ is an atomic CMSO formula and $\alpha \oplus \gamma$ matches ψ and prove that dec_ψ satisfies (3) for this case. For this, we distinguish cases depending on the kind of ψ. During our case analysis, we use quotation marks to delimit the string that corresponds to a formula and use the symbol $\circ$ to denote the concatenation operation between strings. For example, if $\psi =$ "$\exists x \forall y\ \neg\phi(x, y)$," then $\psi =$ "$\exists x \forall y$" $\circ$ "$\neg\phi(x, y)$."

We give a detailed proof in the case where $\psi =$ "$x \in x'$." We also provide a brief description of the proofs for the remaining cases that can all be formalized in a similar fashion.

Case 1. $\psi =$ "$x \in x'$," where x is a vertex variable and x' is a vertex set variable. Then $\mathsf{dec}_\psi(\mathsf{sgn}_\psi(\alpha), \gamma)$ is computed by the procedure in Table I.

It can be easily verified that the procedure in Table I outputs true if and only if $(\alpha \oplus \gamma)[r_x] \in (\alpha \oplus \gamma)[r_{x'}]$—that is, if and only if $\sigma(\alpha \oplus \gamma) =$ true. Furthermore, every query of the procedure can be answered by inspecting $\mathsf{sgn}_\psi(\alpha)$ and γ. The numbers in the parentheses in the procedure correspond to the items of the encoding of $\mathsf{sgn}_\psi(\alpha)$ that are used to answer each query about α. This completes the proof of Case 1.

Case 2. $\psi =$ "$x \in x'$," where x is an edge variable and x' is a edge set variable. Here the function dec_ψ should decide whether $\sigma(\alpha \oplus \gamma)$ is true, which in this case is the same as asking whether $(\alpha \oplus \gamma)[r_x] \in (\alpha \oplus \gamma)[r_{x'}]$ is true. This last question is equivalent to asking whether one of the following holds:

$$\alpha[r_x] \in \alpha[r_{x'}] \tag{5}$$

$$\gamma[r_x] \in \gamma[r_{x'}] \tag{6}$$

$$\alpha[r_x] \in E(G_\alpha[\delta(G_\alpha)]) \text{ and } \lambda_{G_\alpha}(\alpha[r_x]) \in \lambda_{G_\gamma}(\gamma[r_{x'}] \cap E(G_\gamma[\delta(G_\gamma)])) \tag{7}$$

$$\gamma[r_x] \in E(G_\gamma[\delta(G_\gamma)]) \text{ and } \lambda_{G_\gamma}(\gamma[r_x]) \in \lambda_{G_\alpha}(\gamma[r_{x'}] \cap E(G_\alpha[\delta(G_\alpha)])). \tag{8}$$

Each query in (5) through (8) can be answered given γ and $\mathsf{sgn}_\psi(\alpha)$ (but no access to α itself).

Case 3. $\psi =$ "$x = x'$," where both x and x' are vertex variables. Here the function dec_ψ should decide whether $\sigma(\alpha \oplus \gamma)$ is true, which in this case is the same as asking whether $(\alpha \oplus \gamma)[r_x] = (\alpha \oplus \gamma)[r_{x'}]$ is true. This last question is equivalent to asking whether one of the following holds:

$$\alpha[r_x] = \alpha[r_{x'}] \neq \star \tag{9}$$

$$\gamma[r_x] = \gamma[r_{x'}] \neq \star \tag{10}$$

$$\alpha[r_x] \in \delta_{G_\alpha} \text{ and } \gamma[r_{x'}] \in \delta_{G_\gamma} \text{ and } \lambda_{G_\alpha}(\alpha[r_x]) = \lambda_{G_\gamma}(\gamma[r_{x'}]) \tag{11}$$

$$\alpha[r_{x'}] \in \delta_{G_\alpha} \text{ and } \gamma[r_x] \in \delta_{G_\gamma} \text{ and } \lambda_{G_\alpha}(\alpha[r_{x'}]) = \lambda_{G_\gamma}(\gamma[r_x]). \tag{12}$$

The preceding is correct because $\alpha \sim_c \gamma$ implies that at most one of $\alpha[r_x]$ and $\gamma[r_x]$ is a $\star$, and whenever neither of them are $\star$'s, it holds that $\alpha[r_x] \in \delta_{G_\alpha}$, $\gamma[r_x] \in \delta_{G_\gamma}$, and $\lambda_{G_\alpha}(\alpha[r_x]) = \lambda_{G_\gamma}(\gamma[r_x])$ and the same holds for $\alpha[r_{x'}]$ and $\gamma[r_{x'}]$. Again, each query in (9) through (12) can be answered given γ and $\mathsf{sgn}_\psi(\alpha)$.

Case 4. $\psi =$ "$x = x'$," where both x and x' are edge variables. This case is very similar to Case 3 and is omitted.

Case 5. $\psi =$ "$\mathbf{inc}(x, x')$," where x is an edge variable and x' is a vertex variable. Again, here the function dec_ψ should decide whether $\sigma(\alpha \oplus \gamma)$ is true and this is equivalent to $(\alpha \oplus \gamma)[r_{x'}] \subseteq (\alpha \oplus \gamma)[r_x]$. This last question is equivalent to asking whether one of the following holds:

$$\star \neq \alpha[r_{x'}] \subseteq \alpha[r_x] \tag{13}$$

$$\star \neq \gamma[r_{x'}] \subseteq \gamma[r_x] \tag{14}$$

$$\alpha[r_{x'}] \in \delta(G_\alpha) \text{ and } \lambda_{G_\alpha}(\alpha[r_{x'}]) \in \lambda_{G_\gamma}(\gamma[r_x]) \tag{15}$$

$$\gamma[r_{x'}] \in \delta(G_\gamma) \text{ and } \lambda_{G_\gamma}(\gamma[r_{x'}]) \in \lambda_{G_\alpha}(\alpha[r_x]). \tag{16}$$

As in Case 3, the preceding is correct because $\alpha \sim_c \gamma$, and it is enough to verify that each query in (13) through (16) can be answered given γ and $\mathsf{sgn}_\psi(\alpha)$.

Case 6. $\psi =$ "$\mathbf{card}_{q,r}(x)$," where x is a vertex set variable. The function dec_ψ should decide whether $\sigma(\alpha \oplus \gamma)$ is true, which in this case means that

$$|(\alpha \oplus \gamma)[r_x]| \equiv q \pmod r.$$

This, in turn, is equivalent to

$$|\alpha[r_x]| + |\gamma[r_x]| - |\lambda_{G_\alpha}(\alpha[r_x] \cap \delta(G_\alpha)) \cap \lambda_{G_\gamma}(\gamma[r_x] \cap \delta(G_\gamma))| \equiv q \pmod r. \tag{17}$$

It is easy to see that (17) can be evaluated given γ and $\mathsf{sgn}_\psi(\alpha)$. This proves Property (**ii**), and therefore the statement of the lemma holds when ψ is an atomic sentence.

To complete the proof, we now complete the definition of sgn_ψ for every nonatomic normalized CMSO sentence ψ, and we will define dec_ψ for all pairs $\mathsf{sgn}_\psi(\alpha)$, γ such that $\alpha \oplus \gamma$ match ψ. As in the case of atomic formulas, this should be done in a way such that (**i**) holds for sgn_ψ and (3) holds for dec_ψ.

By using induction, we assume that $\mathsf{sgn}_{\psi'}$ and $\mathsf{dec}_{\psi'}$ have been defined such that $\mathsf{sgn}_{\psi'}$ satisfies Property (**i**) and $\mathsf{dec}_{\psi'}$ satisfies (3) for every normalized CMSO sentence ψ' and has length smaller than ψ. This together with the decoder claim implies Property (**ii**) for ψ', namely that

$$\forall \alpha', \beta' \in \mathcal{A} \;\; \mathsf{sgn}_{\psi'}(\alpha') = \mathsf{sgn}_{\psi'}(\beta') \Rightarrow \alpha' \equiv_{\psi'} \beta'. \tag{18}$$

One of the following cases applies.

Case 1. $\psi =$ "$\neg$" $\circ\, \psi'$, where both ψ and ψ' have the same free variables whose rank is the same in ψ and ψ'. From the induction hypothesis, we know that there exist $\mathsf{sgn}_{\psi'}$

and $\mathsf{dec}_{\psi'}$ such that $\mathsf{sgn}_{\psi'}$ satisfies Property (**i**) and $\mathsf{dec}_{\psi'}$ satisfies (3). We define

$$\mathsf{sgn}_{\psi}(\alpha) \;=\; \mathsf{sgn}_{\psi'}(\alpha). \tag{19}$$

We also define

$$\mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) \;=\; \neg\mathsf{dec}_{\psi'}(\mathsf{sgn}_{\psi'}(\alpha), \gamma). \tag{20}$$

Notice that in (20), dec_{ψ} is indeed a function of $\mathsf{sgn}_{\psi}(\alpha)$ and γ because of the definition of $\mathsf{sgn}_{\psi}(\alpha)$ in (19). By the induction hypothesis, for every $p \in \mathbb{N}$ and $I \subseteq \mathbb{Z}^+$, $\mathsf{sgn}_{\psi}(\mathcal{A}_I^p) = \mathsf{sgn}_{\psi'}(\mathcal{A}_I^p)$ is finite, yielding that sgn_{ψ} satisfies Property (**i**).

To prove that dec_{ψ} satisfies (3), let $\alpha \in \mathcal{A}_I^p$ and $\gamma \in \mathcal{A}^p$ with $\alpha \sim_c \gamma$. Then

$$\sigma_{\psi}(\alpha \oplus \gamma) = \neg\sigma_{\psi'}(\alpha \oplus \gamma) = \neg\mathsf{dec}_{\psi'}(\mathsf{sgn}_{\psi'}(\alpha)) =^{(20)} \mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma),$$

where the second equation holds because of the induction hypothesis.

Case 2. $\psi = \psi_1 \circ$ "$\vee$" $\circ\, \psi_2$, where ψ_1 and ψ_2 have the same free variables and the free variables have the same rank in ψ, ψ_1, and ψ_2. From the induction hypothesis, we know that there exist sgn_{ψ_1}, sgn_{ψ_2}, dec_{ψ_1}, and dec_{ψ_2} such that sgn_{ψ_1} and sgn_{ψ_2} both satisfy Property (**i**), whereas dec_{ψ_1} and dec_{ψ_2} both satisfy (3).

We define

$$\mathsf{sgn}_{\psi}(\alpha) \;=\; \mathsf{encode}(\mathsf{sgn}_{\psi_1}(\alpha), \mathsf{sgn}_{\psi_2}(\alpha)), \tag{21}$$

where encode is a function that receives two strings and encodes them as a single string. We also define two functions, decode_1 and decode_2, such that

$$\mathsf{decode}_i(\mathsf{encode}(\mathbf{s}_1, \mathbf{s}_2)) = \mathbf{s}_i, \text{ for } i \in \{1, 2\}.$$

We now define

$$\begin{aligned}\mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) \;=\; & \mathsf{dec}_{\psi_1}(\mathsf{decode}_1(\mathsf{sgn}_{\psi}(\alpha)), \gamma)\\ & \vee\; \mathsf{dec}_{\psi_2}(\mathsf{decode}_2(\mathsf{sgn}_{\psi}(\alpha)), \gamma).\end{aligned}$$

From (21), we have that for every $p \in \mathbb{N}$ and $I \subseteq \mathbb{Z}^+$,

$$\mathsf{sgn}_{\psi}\big(\mathcal{A}_I^p\big) \subseteq \mathsf{encode}(\mathsf{sgn}_{\psi_1}\big(\mathcal{A}_I^p\big), \mathsf{sgn}_{\psi_2}\big(\mathcal{A}_I^p\big)) \cup \{\epsilon\}. \tag{22}$$

By the induction hypothesis, $\mathsf{sgn}_{\psi_i}(\mathcal{A}_I^p)$ is finite, for $i \in \{1, 2\}$. This, together with (22), implies that sgn_{ψ} satisfies Property (**i**).

To prove that dec_{ψ} satisfies (3), observe that for all $\alpha \in \mathcal{A}_I^p$, $\gamma \in \mathcal{A}^p$ such that $\alpha \sim_c \gamma$,

$$\begin{aligned}\sigma_{\psi}(\alpha \oplus \gamma) = \mathsf{true} \iff & (\sigma_{\psi_1}(\alpha \oplus \gamma) = \mathsf{true}) \bigvee (\sigma_{\psi_2}(\alpha \oplus \gamma) = \mathsf{true})\\ \iff & (\mathsf{dec}_{\psi_1}(\mathsf{sgn}_{\psi_1}(\alpha), \gamma) = \mathsf{true}) \bigvee (\mathsf{dec}_{\psi_2}(\mathsf{sgn}_{\psi_2}(\alpha), \gamma) = \mathsf{true})\\ \iff & (\mathsf{dec}_{\psi_1}(\mathsf{decode}_1(\mathsf{sgn}_{\psi}(\alpha)), \gamma) = \mathsf{true})\\ & \bigvee (\mathsf{dec}_{\psi_2}(\mathsf{decode}_2(\mathsf{sgn}_{\psi}(\alpha)), \gamma) = \mathsf{true})\\ \iff & \mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) = \mathsf{true}.\end{aligned}$$

The first equivalence holds because of the definition of ψ, the second by the induction hypothesis, the third by the definition of decode_i, and the last one by the definition of dec_{ψ}.

Case 3. ψ = "$\exists x \subseteq V(G)$" $\circ\, \psi'$, where ψ has p free variables and ψ' has $p+1$ free variables, the ranks of the free variables of ψ and ψ' are the same, except for the variable x, which is a free variable in ψ' but is not free in ψ and the rank of x in ψ' is $p+1$. From the induction hypothesis, we know that there exist $\mathsf{sgn}_{\psi'}$ and $\mathsf{dec}_{\psi'}$ such that $\mathsf{sgn}_{\psi'}$ satisfies Property (**i**) and $\mathsf{dec}_{\psi'}$ satisfies (3). We define

$$\mathsf{sgn}_{\psi}(\alpha) \;=\; \mathsf{encode}(\{\mathsf{sgn}_{\psi'}(\alpha \diamond x) \mid x \subseteq V(G_\alpha)\}), \tag{23}$$

where given a set $\mathcal{W}$ of signatures, the string $\mathsf{encode}(\mathcal{W})$ encodes all members of $\mathcal{W}$. We also define the function decode that receives as an entry a string $\mathbf{s}$ and outputs the set of strings that are encoded to it, particularly $\mathsf{decode}(\mathsf{encode}(\mathcal{W})) = \mathcal{W}$. We now define

$$\mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) \;=\; \bigvee_{\substack{\mathbf{s}\in \mathsf{decode}\,(\mathsf{sgn}_{\psi}(\alpha))\\ y \subseteq V(G_\gamma)\\ \text{such that } \mathsf{invsgn}_{\psi'}(\mathbf{s}) \sim_c (\gamma \diamond y),}} \sigma_{\psi'}(\mathsf{invsgn}_{\psi'}(\mathbf{s}) \oplus (\gamma \diamond y)) \tag{24}$$

where given a string $\mathbf{s}$ encoding a signature, $\mathsf{invsgn}_{\psi'}(\mathbf{s})$ returns the lexicographically smallest boundaried structure $\alpha^\star$ such that $\mathsf{sgn}_{\psi'}(\alpha^\star) = \mathbf{s}$. First observe that the function dec_{ψ} is indeed a function of $\mathsf{sgn}_{\psi}(\alpha)$ and γ. By the construction of sgn_{ψ}, for all $p \in \mathbb{N}$ and every finite $I \subseteq \mathbb{N}$, it holds that

$$\mathsf{sgn}_{\psi}\big(\mathcal{A}_I^p\big) \in \mathsf{encode}\big(2^{\mathsf{sgn}_{\psi'}(\mathcal{A}_I^{p+1})}\big) \cup \{\epsilon\},$$

which proves that sgn_{ψ} satisfies Property (**i**) (given a set X, we denote by 2^X the set of all of its subsets). It remains to prove that dec_{ψ} satisfies (3), namely that for all $\alpha \in \mathcal{A}_I^p$ and $\gamma \in \mathcal{A}_I$ such that $\alpha \sim_c \gamma$, the following hold:

$$\mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) = \mathsf{true} \Rightarrow \sigma_{\psi}(\alpha \oplus \gamma) = \mathsf{true} \tag{25}$$

$$\mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) = \mathsf{true} \Leftarrow \sigma_{\psi}(\alpha \oplus \gamma) = \mathsf{true}. \tag{26}$$

To prove (25), assume that $\mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) = \mathsf{true}$. Thus, there exist some $y \subseteq V(G_\gamma)$ and $\mathbf{s} \in \mathsf{decode}(\mathsf{sgn}_{\psi}(\alpha))$ such that $\mathsf{invsgn}_{\psi'}(\mathbf{s}) \sim_c (\gamma \diamond y)$ and

$$\sigma_{\psi'}(\mathsf{invsgn}_{\psi'}(\mathbf{s}) \oplus (\gamma \diamond y)) = \mathsf{true}. \tag{27}$$

As $\mathsf{decode}(\mathsf{sgn}_{\psi}(\alpha)) = \{\mathsf{sgn}_{\psi'}(\alpha \diamond x) \mid x \subseteq V(G_\alpha)\}$, we may select an $x \subseteq V(G_\alpha)$ such that $\mathbf{s} = \mathsf{sgn}_{\psi'}(\alpha \diamond x)$. Therefore, the construction of $\mathsf{invsgn}_{\psi'}$ ensures that $\mathsf{sgn}_{\psi'}(\mathsf{invsgn}_{\psi'}(\mathbf{s})) = \mathbf{s} = \mathsf{sgn}_{\psi'}(\alpha \diamond x)$. From (18), $\mathsf{invsgn}_{\psi'}(\mathbf{s}) \equiv_{\psi'} \alpha \diamond x$. This means that $(\alpha \diamond x) \sim_c (\gamma \diamond y)$; $\sigma_{\psi'}(\mathsf{invsgn}_{\psi'}(\mathbf{s}) \oplus (\gamma \diamond y)) = \sigma_{\psi'}((\alpha \diamond x) \oplus (\gamma \diamond y))$; and, from (27), it follows that

$$\sigma_{\psi'}((\alpha \diamond x) \oplus (\gamma \diamond y)) = \mathsf{true}.$$

Recall that $(\alpha \diamond x) \oplus (\gamma \diamond y) = (\alpha \oplus \gamma) \diamond (x \cup y)$. Therefore,

$$\sigma_{\psi'}((\alpha \oplus \gamma) \diamond (x \cup y)) = \mathsf{true},$$

which, by the definition of ψ, implies that $\sigma_{\psi}(\alpha \oplus \gamma) = \mathsf{true}$ and (25) follows.

It now remains to prove (26). Assume that the value of $\sigma_{\psi}(\alpha \oplus \gamma) = \mathsf{true}$. Thus, by the definition of ψ, there exist some $x \subseteq V(G_\alpha)$ and some $y \subseteq V(G_\gamma)$ such that $(\alpha \diamond x) \sim_c (\gamma \diamond y)$ and

$$\sigma_{\psi'}((\alpha \diamond x) \oplus (\gamma \diamond y)) \;=\; \mathsf{true}. \tag{28}$$

Let $\mathbf{s} = \mathsf{sgn}_{\psi'}(\alpha \diamond x)$, and observe, by (23), that $\mathbf{s} \in \mathsf{decode}(\mathsf{sgn}_{\psi}(\alpha))$. By the definition of $\mathsf{invsgn}_{\psi'}$, we have that $\mathsf{sgn}_{\psi'}(\mathsf{invsgn}_{\psi'}(\mathbf{s})) = \mathsf{sgn}_{\psi'}(\alpha \diamond x) = \mathbf{s}$. By (18), $\mathsf{invsgn}_{\psi'}(\mathbf{s}) \equiv_{\psi'} \alpha \diamond x$.

Hence, from (28), we obtain that $\mathsf{invsgn}_{\psi'}(\mathbf{s}) \sim_c (\gamma \diamond y)$ and

$$\sigma_{\psi'}(\mathsf{invsgn}_{\psi'}(\mathbf{s}) \oplus (\gamma \diamond y)) = \mathsf{true}.$$

Notice that $\mathbf{s}$ and y certify, in (24), that $\mathsf{dec}_{\psi}(\mathsf{sgn}_{\psi}(\alpha), \gamma) = \mathsf{true}$, yielding (26).

(Multi) case 4. $\psi =$ "$\exists x \subseteq E(G)$" $\circ\, \psi'$ or $\psi =$ "$\exists x \in V(G)$" $\circ\, \psi'$ or $\psi =$ "$\exists x \in E(G)$" $\circ\, \psi'$. The proof of the first case is the same as the proof of Case 3. The proof for the remaining two cases differs from the proof of Case 3 only in that when the variables of x and y in the proof are quantified as vertices or edges of the vertex or edge set, respectively, of a boundaried structure, they may also take the value $\star$.

Considering that the preceding case analysis is complete, the proof follows. □

4. DERIVATION OF OUR RESULTS

In this section, we give two master theorems from which all of our results will be derived. We start with fundamental notions of our article. These are the notions of *protrusion*, *protrusion replacement*, and *protrusion decomposition*.

Definition 4.1 *[t-Protrusion].* Given a graph G, we say that a set $X \subseteq V$ is a *t-protrusion* of G if $|\partial(X)| \leq t$ and $\mathbf{tw}(G[X]) \leq t$.

Definition 4.2 *[(f, a)-Protrusion Replacement Family].* Let Π be a parameterized graph problem, let $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ be a nondecreasing function, and let $a \in \mathbb{Z}^+$. An (f, a)-*protrusion replacement family* for Π is a collection $\mathcal{A} = \{\mathsf{A}_i \mid i \geq 0\}$ of algorithms such that algorithm A_i receives as input a pair (I, X), where

—I is an instance of Π whose graph and parameter are G and $k \in \mathbb{Z}$, *and*
—X is an i-protrusion of G with at least $f(i) \cdot k^a$ vertices,

and outputs an equivalent instance I^* such that if G^* and k^* are the graph and the parameter of I^*, then $|V(G^*)| < |V(G)|$ and $k^* \leq k$. The running time of a (f, a)-protrusion replacement family is the running time of A_i. □

Definition 4.3 *[(α, β)-Protrusion Decomposition].* An (α, β)-*protrusion decomposition* of a graph G is a partition $\mathcal{P} = \{R_0, R_1, \ldots, R_\rho\}$ of $V(G)$ such that

—$\max\{\rho, |R_0|\} \leq \alpha$,
—each $R_i^+ = N_G[R_i]$, $i \in \{1, \ldots, \rho\}$, is a β-protrusion of G, and
—for every $i \in \{1, \ldots, \rho\}$, $N_G(R_i) \subseteq R_0$.

We call the sets R_i^+, $i \in \{1, \ldots, \rho\}$, the *protrusions* of $\mathcal{P}$.

4.1. Meta-Algorithmic Properties

We define the following two properties for a parameterized graph problem Π.

A [**Protrusion Replacement**]**:** There exists an (f, a)-protrusion replacement family $\mathcal{A}$ for Π, for some function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ and some $a \in \mathbb{Z}^+$.
B [**Protrusion Decomposition**]**:** There exists a constant c such that if G and $k \in \mathbb{Z}^+$ are the graph and the parameter of a YES-instance of Π, then G admits a $(c \cdot k, c)$-protrusion decomposition.

We also consider the following weaker version of the combinatorial property:

B* [**Weak Protrusion Decomposition**]**:** There exist a constant c' and a nondecreasing function $g : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that for every $x \in \mathbb{Z}^+$, if G and $k \in \mathbb{Z}^+$ are the graph and the parameter of a YES-instance of Π such that all c'-protrusions of G are of size at most x, then G has a $(g(x) \cdot k, g(x))$-protrusion decomposition.

To see that $\mathbf{B}$ implies $\mathbf{B}^*$, set $c' = 1$ and consider the function g, with $g(x) = c$, where c is the constant in the definition of $\mathbf{B}$.

4.2. The Meta-Algorithm

All of our kernelization algorithms are based on the following procedure, which makes use of some (f, a)-protrusion replacement family $\mathcal{A} = \{\mathsf{A}_i \mid i \geq 0\}$. In the procedure, given a set $R \subseteq V(G)$, we define $\mathcal{C}_R$ as the set of connected components of $G \backslash R$ that have treewidth at most $|R|$. Let X_R be the set of vertices that are either in R or in some of the connected components of $\mathcal{C}_R$.

Meta-kernelization(t)
Input: An instance I of a parameterized graph problem.
Output: An equivalent instance I'.

If $k \geq 0$ and $|I| \leq k$, we return I. Although there exists some $R \subseteq V(G)$ of size at most $2t$ such that$|X_R| \geq f(2 \cdot |R|) \cdot k^a$, applyalgorithm $\mathsf{A}_{2\cdot|R|}$ with the pair (I, X_R) as input and replace I by the output I' of this algorithm. In case the parameter k' of I' is negative, then output a trivial YES- or NO-instance of Π depending on whether $(I', -1) \in \Pi$ or not.

LEMMA 4.4. *Procedure* Meta-kernelization(t) *runs in* $|I|^{O(t)}$ *steps. Moreover, it outputs an instance with a graph G such that for all $i \in \{0, \ldots, t\}$, all i-protrusions of G have size at most $f(2i) \cdot k^a$.*

PROOF. Notice that the while-loop of the procedure will be applied less than $n = |I|$ times, as each iteration decreases the size of the graph by at least one. In each iteration of the outer loop, we have to consider $O(|I|^{2t})$ different choices for R. For each choice of R, the set X_R can be computed in linear time using the algorithm of Bodlaender [1996]. That way, the procedure requires $O(|I|^{2t+2})$ steps in total. To show that the input specifications of the algorithm $\mathsf{A}_{2\cdot|R|}$ are satisfied when it is called, we argue that every time the algorithm $A_{2\cdot|R|}$ is applied to (I, X_R), X_R is a $2 \cdot |R|$-protrusion of the graph G in the instance of I. For this, notice that $\partial_G(X_R) \subseteq R$ and $\mathbf{tw}(G[X_R]) \leq \mathbf{tw}(G[X_R \backslash R]) + |R| \leq 2|R|$.

Let I' be the output of Meta-kernelization(t) and G be the graph of I'. Assume toward a contradiction that for some $j \in \{0, \ldots, t\}$,G contains a j-protrusion X of size $> f(2j) \cdot k^a$. Let $R = \partial_G(X)$. Observe that $|R| \leq j$ and that every connected component C of $G \backslash R$ that contains at least one vertex of X is contained in X. Thus, $\mathbf{tw}(C) \leq j$, and therefore $X \subseteq X_R$. But then X_R is a $2j$-protrusion of G of size $\geq f(2j) \cdot k^a$, contradicting the fact that I' is the output of Meta-kernelization(t). □

4.3. Two Master Theorems

Our results can be deduced from the following two master theorems. Although their proofs are similar in spirit, we present them separately to illustrate the way properties $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{B}^*$ are combined.

THEOREM 4.5. *If a parameterized graph problem Π has property* $\boldsymbol{A}$ *for some nonnegative constant a and property* $\boldsymbol{B}$ *for some constant c, then Π admits a kernel of size* $O(k^{a+1})$.

PROOF. Let $\mathcal{A} = \{\mathsf{A}_i \mid i \geq 0\}$ be an (f, a)-protrusion replacement family for Π. We claim that the required kernelization algorithm is Meta-kernelization(c).

Suppose that I is a YES-instance of Π. The Meta-kernelization(c) procedure transforms I to a YES-instance I^* of Π. Assume that G^* and k^* respectively are the graph and the parameter of I^*. First of all we, assume that $k^* \geq 0$, else Meta-kernelization(c) returns a trivial YES- or NO-instance. Let $\mathcal{P} = \{R_0, R_1, \ldots, R_\rho\}$ be a $(c \cdot k^*, c)$-protrusion decomposition of G^* for some $\rho \leq c \cdot k^*$, whose existence follows from property **B**. Notice that $k^* \leq k$. Therefore, from Lemma 4.4, we have that

$$|V(G^*)| \leq |R_0| + \sum_{i=1}^{\rho} |R_i| \leq c \cdot k + c \cdot k \cdot f(2c) \cdot k^a = c \cdot k \cdot (f(2c) \cdot k^a + 1).$$

Hence, if the preceding procedure outputs an instance whose graph has more than $c \cdot k \cdot (f(2c) \cdot k^a + 1)$ vertices, then the (I, k) is a NO-instance; in this case, the algorithm outputs a trivial NO-instance of Π. Otherwise, by Lemma 4.4, the algorithm outputs, in $O(|I|^{2c+2})$ steps, an equivalent instance with a graph on $O(k^{a+1})$ vertices, as required. □

When $a = 0$, we can use the weaker condition $\mathbf{B}^*$ and have a linear kernel.

THEOREM 4.6. *If a parameterized graph problem* Π *has property* **A** *for* $a = 0$ *and property* $\mathbf{B}^*$ *for some constant* c, *then* Π *admits a linear kernel.*

PROOF. Let $\mathcal{A} = \{\mathsf{A}_i \mid i \geq 0\}$ be an $(f, 0)$-protrusion replacement family for Π. (Notice that in this proof, it is important that $a = 0$.)

Let also $g : \mathbb{Z}^+ \to \mathbb{Z}^+$ be a function such that for every $x \in \mathbb{Z}^+$, if G and k are the graph and the parameter of a YES-instance of Π such that all c-protrusions of G have size at most x, then G has a $(g(x) \cdot k, g(x))$-protrusion decomposition. We claim that the required kernelization algorithm is Meta-kernelization(c). Let $t = g(f(2c))$.

Suppose now that I is a YES-instance of Π. The Meta-kernelization(c) procedure transforms I to a YES-instance I^* of Π. Assume that G^* and k^* respectively are the graph and the parameter of I^*. First of all, we assume that $k^* \geq 0$, else Meta-kernelization(c) returns a trivial YES- or NO-instance. By Lemma 4.4, I^* has no c-protrusion of size at least $f(2c)$. By applying Condition $\mathbf{B}^*$ for $x = f(2c)$, we have that G^* has a $(t \cdot k^*, t)$-protrusion decomposition $\mathcal{P} = \{R_0, R_1, \ldots, R_\rho\}$ for some $\rho \leq t \cdot k^*$. Notice that $k^* \leq k$. By Lemma 4.4, we have that

$$|V(G^*)| \leq |R_0| + \sum_{i=1}^{\rho} |R_i| \leq t \cdot k + t \cdot k \cdot f(2c) = t \cdot k \cdot (f(2c) + 1).$$

Hence, if the preceding procedure outputs an instance whose graph has more than $t \cdot k \cdot (f(2c) + 1)$ vertices, then the algorithm outputs a trivial NO-instance of Π. Otherwise, by Lemma 4.4, the algorithm outputs, in $O(|I|^{2t+2})$ steps, an equivalent instance on $O(k)$ vertices, as required. □

We now have all necessary notions to present how the meta-algorithmic theorems mentioned in Section 1 are derived from Master Theorems 4.5 and 4.6.

4.4. Problems Having the Algorithmic and Combinatorial Properties

Our meta-algorithmic results follow by combining the following six results. The first four imply the protrusion replacement property **A**:

—Every annotated p-MIN-CMSO[ψ] problem has the protrusion replacement property **A** for $a = 1$ (Lemma 5.8 in Section 5.2).
—Every annotated p-EQ-CMSO[ψ] problem has the protrusion replacement property **A** for $a = 2$ (Lemma 5.12 in Section 5.3).
—Every annotated p-MAX-CMSO[ψ] has the protrusion replacement property **A** for $a = 1$ (Lemma 5.17 in Section 5.4).
—Every parameterized graph problem Π that has FII has the protrusion replacement property **A** for $a = 0$ (Lemma 5.19 in Section 5.5).

The two last results imply the protrusion decomposition properties **B** and **B***:

—Every r-coverable problem has the protrusion decomposition property **B** (Lemma 6.1 in Section 6.2).
—Every r-quasi-coverable problem has the weak protrusion decomposition property **B*** (Lemma 6.4 in Section 6.3).

4.5. Derivation of Theorems 1.1, 1.2, and 1.3

All of our main results are consequences of Master Theorems 4.5 and 4.6. Theorem 1.1 follows from Master Theorem 4.5 and Lemmata 5.8, 5.12, 5.17, and 6.1. Moreover, Theorem 1.3 follows from Master Theorem 4.6 and Lemmata 5.19 and 6.4. We conclude this section with the proof of Theorem 1.2.

Proof of Theorem 1.2. Suppose that Π is NP-hard and its annotated version Π^α is in NP. Consider an algorithm that, given an instance $I = (G, k)$ of Π, applies first the kernelization algorithm of Theorem 1.1 as a subroutine on the annotated instance $((G, V(G)), k)$—that is, all vertices of G are set to be annotated. This subroutine outputs an equivalent annotated instance $I' = ((G', Y'), k)$ of Π^α where the number of vertices in G' is a polynomial function of k. The next step of the algorithm is to apply a polynomial time many-to-one reduction from Π^α to Π on I' and obtain an equivalent instance $I'' = (G'', k'')$, where $|I''|$ is a polynomial function of $|I'|$. This reduction exists from the Cook–Levin theorem, as $\Pi^\alpha \in$ NP and Π is NP-hard. Then $|I''|$ is a polynomial function of k, and this two-step polynomial time algorithm is the desired kernelization algorithm for Π. The reduction from Π^α to Π might output an instance I'' with parameter k'', where k'' is exponential in $|I''|$ because k'' could be encoded in binary. However, since Π is a p-min/eq/max-CMSO$[\psi]$ problem, $(I'', k'') \in \Pi$ if and only if $(I'', k''') \in \Pi$, where $k''' = \min\{k'', |I''| + 1\}$. The kernelization algorithm outputs (I'', k'''). □

5. REDUCTION RULES

In this section, we prove the existence of protrusion replacement families for p-min/eq/max-CMSO$[\psi]$ graph problems and for parameterized problems that have FII.

5.1. Model Checking on Structures

To prove our reduction rules, we consider an extension of p-min/eq/max-CMSO problems to a setting where the input is a structure rather than a graph. Specifically, we consider the following problems.

Min/Max-CMSO on Structures
Input: A structure α and a CMSO sentence ψ.
Output: A minimum/maximum size subset S of $V(G)$ (or $E(G)$) such that $(\alpha \diamond S) \models \psi$.

Eq-CMSO on Structures
Input: A structure α, a CMSO sentence ψ, and an integer k.
Output: A subset S of $V(G)$, (or $E(G)$) $|S| = k$ such that $(\alpha \diamond S) \models \psi$.

Observe that in the preceding problems, the CMSO sentence is part of the input and not fixed as in the case of p-min/eq/max-CMSO$[\psi]$ problems. We will repeatedly apply the following result from Theorem 5 of Borie et al. [1992] (see also Arnborg et al. [1991]).

Proposition 5.1. *There exists a computable function $f : \mathbb{Z}^+ \times \mathbb{Z}^+ \to \mathbb{Z}^+$ and an algorithm that solves* Min/Max/Eq-CMSO on Structures *in $f(\mathbf{tw}(G_\alpha), |\psi|) \cdot |V(G_\alpha)|$ steps.*

Proposition 5.1 is a slight strengthening of Theorem 5 of Borie et al. [1992]; what is shown there explicitly is the corresponding version where the input is a graph rather than a structure. Arnborg et al. [1991] show the variant of Proposition 5.1 for MSO logic rather than CMSO logic. Either of these proofs can be made to work both on structures and with CMSO logic.

The construction of each protrusion replacement family depends on whether we are dealing with an annotated p-MIN-CMSO[ψ], p-EQ-CMSO[ψ], or p-MAX-CMSO[ψ] problem, or whether the problem in question has FII. For the case of annotated problems, the constructions consist of three parts. In the first two parts, we focus on reducing the set of annotated vertices, and in the last part, we reduce the set of vertices. In all cases, we assume that we are given a sufficiently large t-protrusion. In the following discussion, we deal with annotated p-MIN/EQ/MAX-CMSO[ψ] problems where the set S in question is a set of vertices. The case where S is a set of edges can be dealt with in an identical manner.

5.2. Protrusion Replacement Families for Annotated *p*-MIN-CMSO[ψ] Problems

We start from the existence of a protrusion replacement family for annotated p-MIN-CMSO[ψ] problems. The technique employed in this section will act as a template for other types of annotated problems. Recall that in an annotated p-MIN-CMSO[ψ] problem Π^α, we are given a structure (G, Y) and an integer k. The objective is to find a set $S \subseteq Y$ of size at most k such that (G, S) models some CMSO sentence ψ. For our reduction rule, we are also given a sufficiently large t-protrusion X. In the first step of the reduction, we show that the set $Y \cap X$ can be substituted in $O(|X|)$ steps by a new set Z of $O(k)$ vertices such that $((G, Y), k)$ is a YES-instance if and only if $((G, Z \cup (Y \setminus X)), k)$ is a YES-instance. In the second step, we show that the t-protrusion X can be partitioned into $O(k)$ t'-protrusions, where $t' = O(t)$, such that each t'-protrusion contains vertices from Z only in its (bounded size) boundary. In the third and final step of the reduction rule, we replace the largest t'-protrusion with an equivalent, but smaller, t'-boundaried graph. For the case of p-MIN-CMSO[ψ] problems, these three reduction steps correspond to Lemmata 5.3, 5.4, and 5.6, respectively.

We start by proving a lemma that lets us analyze the interior of a protrusion without bothering about the rest of the graph.

LEMMA 5.2. *There is an algorithm that, given two boundaried structures* (G_X, Y_X) *and* (G_R, S_R) *of type* (graph, vertex set) *and a CMSO sentence* ψ, *finds a minimum size set* $S_X \subseteq Y_X$, *if such a set exists, such that* $(G_X, S_X) \oplus (G_R, S_R) \models \psi$ *in time* $|V(G_X \oplus G_R)| \cdot f(|\psi|, \mathbf{tw}(G_X \oplus G_R))$.

PROOF. Let $(G', Y', S'_R) = (G_X, Y_X, \emptyset) \oplus (G_R, \emptyset, S_R)$. Finding the desired set $S_X \subseteq Y$ now amounts to finding a minimum size set $S'_X \subseteq Y'$ such that $(G', S'_X \cup S'_R) \models \psi$. This is easily formulated as MIN-CMSO ON STRUCTURES and hence may be solved in the desired running time by Proposition 5.1. □

Reducing the set of annotated vertices. The first step of our reduction rule is based on the following lemma.

LEMMA 5.3. *Let* Π^α *be an annotated* p-MIN-CMSO[ψ] *problem, and let* t *be an integer. Then there exists an algorithm that, given an instance* $((G, Y), k)$ *of* Π^α *and a* t-*protrusion* X *of* G, *outputs, in time* $O(|X|)$, *an equivalent instance* $((G, Y'), k)$ *of* Π^α, *where* $|Y' \cap X| = O(k)$ *and* $Y' \subseteq Y$.

We remark that the constants hidden in the "O"-notation of the complexity of the algorithm and the size of its output depend only on the length of the CMSO sentence ψ defining Π^α and the constant t. From now onward, we will not explicitly mention this.

Proof. Let ψ be the CMSO sentence mentioned in the definition of Π^α. Lemma 3.2 implies that the canonical equivalence relation $\equiv_{\sigma_\psi}$ has finitely many equivalence classes on the set of boundaried structures of arity 2 with label set $\{1,\ldots,t\}$. Let $\mathbf{MinRep}(\psi,t)$ be a set containing a representative (a boundaried structure of arity 2) for each equivalence class of $\equiv_{\sigma_\psi}$ with the minimum number of vertices in the graph of a structure. Given G, Y, and X, we define the sets $B=\partial_G(X)$, $R=(V(G)\backslash X)\cup B$, and the boundaried structures (G_X,Y_X) and (G_R,Y_R) as follows. The boundaried graphs G_X and G_R are just $G[X]$ and $G[R]$, respectively. Both have boundary B, with labels from $\{1,\ldots,t\}$ such that $G_X\oplus G_R=G$. Similarly, $Y_X=Y\cap X$, whereas $Y_R=Y\backslash X$, such that $(G,Y)=(G_X,Y_X)\oplus(G_R,Y_R)$.

For every structure $\alpha=(G_R^\alpha,S_R^\alpha)\in\mathbf{MinRep}(\psi,t)$, we find, using Lemma 5.2, a minimum size set $S_X^\alpha\subseteq Y_X$ such that $(G_X,S_X^\alpha)\oplus\alpha\models\psi$. Since $|\mathbf{MinRep}(\psi,t)|$ and the size of each structure in $\mathbf{MinRep}(\psi,t)$ depends only on $|\psi|$ and t, and the treewidth of $G[X]$ is at most t, this takes time $O(|X|)$. Now, define

$$Y'_X=\bigcup_{\alpha\in\mathbf{MinRep}(\psi,t)}\begin{cases}S_X^\alpha & \text{if } |S_X^\alpha|\le k,\\ \emptyset & \text{otherwise.}\end{cases}$$

We set $Y'=Y'_X\cup Y_R$ (formally, Y'_X and Y_R are vertex sets of different graphs, so actually Y' is the second element of the 2-tuple of $(G_X,Y'_X)\oplus(G_R,Y_R)$, i.e., $Y'=((G_X,Y'_X)\oplus(G_R,Y_R))[2]$, but this is just semantics). Since $|\mathbf{MinRep}(\psi,t)|$ depends only on $|\psi|$ and t, the construction of Y' implies that $|Y'\cap X|=O(k)$.

To complete the proof, it remains to show that $((G,Y'),k)\in\Pi^\alpha$ if and only if $((G,Y),k)\in\Pi^\alpha$. For the forward direction, we have that $Y'\subseteq Y$, and hence feasible solutions to $((G,Y'),k)$ are also feasible for $((G,Y),k)$. We now turn to proving the reverse direction. Let $S\subseteq Y$, $|S|\le k$ be such that $(G,S)\models\psi$. Let $S_X=X\cap S$ and $S_R=S\backslash X$. Observe that $(G_X,S_X)\oplus(G_R,S_R)=(G,S)$ and that $|S_X|+|S_R|=|S|\le k$. Choose $\alpha=(G_R^\alpha,S_R^\alpha)\in\mathbf{MinRep}(\psi,t)$ such that $\alpha\equiv_{\sigma_\psi}(G_R,S_R)$. Let $S_X^\alpha\subseteq Y_X$ be the set computed for α in the previous paragraph. Since

$$(G_X,S_X)\oplus\alpha\models\psi \iff (G_X,S_X)\oplus(G_R,S_R)\models\psi \iff \text{true},$$

it follows that $|S_X^\alpha|\le|S_X|\le k$. Thus, $S_X^\alpha\subseteq Y'_X$. Let $S'=S_X^\alpha\cup S_R$ (again, formally, S_X^α and S_R are vertex sets of different graphs, so actually $S'=((G_X,S_X^\alpha)\oplus(G_R,S_R))[2]$). We have that $S'\subseteq Y'$, $|S'|\le|S_X^\alpha|+|S_R|\le|S_X|+|S_R|=|S|\le k$. Finally, we observe that

$$\begin{aligned}&(G,S')\models\psi\\ \iff&(G_X,S_X^\alpha)\oplus(G_R,S_R)\models\psi\\ \iff&(G_X,S_X^\alpha)\oplus\alpha\models\psi\\ \iff&\text{true}.\end{aligned}$$

This concludes the proof. □

Partitioning protrusions. In the second step of the reduction rule, the t-protrusion X is partitioned into $O(k)$ smaller t'-protrusions for some $t'=O(t)$.

Lemma 5.4. *Let G be a graph, Y be a subset of its vertices, and k be an integer. In addition, let X be a t-protrusion and $Z=X\cap Y$ such that $|Z|\le k$. There is time $O(|X|)$ algorithm that outputs a collection $\mathcal{Q}$ of $(4t+2)$-protrusions such that $X=\bigcup_{Q\in\mathcal{Q}}Q$, $|\mathcal{Q}|=O(k)$, and for every $Q\in\mathcal{Q}$, $Z\cap Q\subseteq\partial_G(Q)$.*

Proof. We assume that $G[X]$ is connected; otherwise, we work independently on its connected components. We find a nice tree decomposition of $G[X]$ and then add $\partial_G(X)$

to all its bags. We denote the resulting tree decomposition by $(T, \mathcal{X})$, and clearly it has width at most $2t$.

The decomposition $(T, \mathcal{X})$ can be constructed in time $O(|X|)$ (e.g., see Bodlaender [1996]). Now we mark a subset of the nodes of T. For each vertex $z \in Z$, we mark, if it exists, the forget node t_z with the property that $\{z\} = X_{t_z} \backslash X_{t'_z}$, where t_z is the child of t'_z in T. As each vertex is forgotten at most once in a nice tree decomposition, so far we have marked at most $|Z| + 1$ nodes of T. Now, as long as this is possible, we keep marking each bag that is the lowest common ancestor of two already marked nodes. Using a standard counting argument for trees, it follows that, in the worst case, this operation doubles the number of marked nodes. Hence, there are at most $O(|Z|)$ marked nodes; we denote this set by M. We say that two nodes $t_1, t_2 \in M$ are *linked* if these nodes are the only marked nodes of the (t_1, t_2)-path in T. We define the set

$$P = \{(t_1, t_2) \mid t_1 \text{ and } t_2 \text{ are linked nodes of } M \text{ and } t_1 \text{ is a predecessor of } t_2\}.$$

We observe that $|P| = O(|Z|)$, and each marked node belongs to some pair in P. Let $\mathcal{C}$ be the set of the connected components of $G[X] \backslash \bigcup_{t \in M} X_t$. By the construction of M, the neighborhood of a connected component C in $\mathcal{C}$ may intersect either a single bag X_t of T or two bags X_{t_1}, X_{t_2} of T such that $(t_1, t_2) \in P$. In the first case, we define $R(C)$ to be some pair in P that contains t as an endpoint (if there are many such pairs, we make an arbitrary choice). In the second case, we define $R(C) = \{t_1, t_2\}$. Given a pair p of P, we use the notation L^{-1} to denote the union of the vertex sets of all connected components of $\mathcal{C}$ that map to p. It is now easy to see that that $\mathcal{R} = \{L^{-1}(p) \mid p \in P\}$ is a partition of $G[X] \backslash \bigcup_{t \in M} X_t$. As each vertex from Z is in some bag corresponding to a marked node, none of the sets in $\mathcal{R}$ intersects Z. Moreover, the neighborhood in G of each set in $\mathcal{R}$ is a subset of at most two bags of $(T, \mathcal{X})$, and thus its neighborhood has at most $2(2t+1)$ vertices. We now define the set $\mathcal{Q} = \{V(R) \cup \partial_G(V(R)) \mid R \in \mathcal{R}\}$. Then each member Q of $\mathcal{Q}$ is an $(4t+2)$-protrusion of G where $Z \cap Q \subseteq \partial_G(Q)$. Moreover, $\bigcup_{Q \in \mathcal{Q}} = X$, and the lemma follows as $|\mathcal{Q}| = |P| = O(k)$. □

We will also need the following simple decomposition lemma for t-protrusions.

LEMMA 5.5. *If a graph G contains a t-protrusion X where $|X| > c > 0$, then it also contains a $(2t+1)$-protrusion Y where $c < |Y| \leq 2c$. Moreover, given a tree-decomposition of X of width at most r, a tree decomposition of Y of width at most $2t$ can be found in time $O(|X|)$.*

PROOF. If $|X| \leq 2c$, we are done. Assume that $|X| > 2c$, and let

$$(T, \mathcal{X} = \{X_x\}_{x \in V(T)}, s)$$

be a nice tree-decomposition of $G[X]$, rooted at some, arbitrarily chosen, node s of T. Given a node x of the rooted tree T, we denote by $D(x)$ the subset of $V(T)$ containing x and all of its descendants in T and by T_x the subtree of T rooted at x. Let $B \subseteq V(T)$ be the set containing each node x of T with the property that the number of vertices appearing in $\bigcup_{y \in D(x)} X_y$ (i.e., the vertices in the bags corresponding to x and its descendants) is more than c. As $|X| \geq 2c$, B is a nonempty set. We choose b to be a member of B whose descendants in T do not belong to B. The choice of b and the fact that T is a binary tree ensure that $c < |\bigcup_{y \in D(b)} X_y| \leq 2c$. We define $Y = \partial_G(X) \cup \bigcup_{y \in D(b)} X_y$ and observe that

$$(T_b, \mathcal{X}') = \{\partial_G(X) \cup X_t\}_{t \in D(b)}, b) \tag{29}$$

is a tree decomposition of $G[Y]$. As $|\partial_G(X)| \leq t$, the width of the tree decomposition in (29) is at most $2t$. Moreover, it holds that $\partial_G(Y) \subseteq \partial_G(X) \cup X_b$, and therefore Y is a $(2t+1)$-protrusion of G. □

Reducing protrusions. In the third phase of our reduction rule, we find a protrusion to replace and perform the replacement.

LEMMA 5.6. *Let Π^α be an annotated p-MIN/EQ-CMSO$[\psi]$ problem. Then for every integer t, there is a $c_1 \in \mathbb{Z}^+$ (depending only on $|\psi|$ and t) and an algorithm that, given an instance $((G, Y), k)$ of Π^α and a t-protrusion X of G, where $c_1 < |X| \leq 2c_1$ and $X \cap Y \subseteq \partial_G(X)$, outputs, in time $O(|X|)$, an equivalent instance $((G^*, Y^*), k)$ of Π^α such that $|V(G')| < |V(G)|$.*

PROOF. We define an equivalence relation between boundaried structures of type (graph, vertex set) as follows. Let $\alpha_1 = (G_1, Y_1)$ and $\alpha_2 = (G_2, Y_2)$ be two boundaried structures with labeling functions $\lambda_1 : \delta(G_1) \to \{1, \ldots, t\}$ and $\lambda_2 : \delta(G_2) \to \{1, \ldots, t\}$, respectively, such that $Y_1 \subseteq \delta(G_1)$ and $Y_2 \subseteq \delta(G_2)$.

We say that $\alpha_1 \approx \alpha_2$ if the following conditions are satisfied:

(1) $\Lambda(G_1) = \Lambda(G_2)$;
(2) $\lambda_1(Y_1) = \lambda_2(Y_2)$; and
(3) for every $S_1 \subseteq Y_1$ and $S_2 \subseteq Y_2$ such that $\lambda_1(S_1) = \lambda_2(S_2)$, it follows that $(G_1, S_1) \equiv_{\sigma_\psi} (G_2, S_2)$.

Notice that $\approx$ is an equivalence relation. Because, in the preceding definition, the sets S_1 and S_2 cannot have more than t vertices, the number of equivalence classes of $\approx$ depends only on t and the number of equivalence classes of $\equiv_{\sigma_\psi}$ on boundaried structures of arity 2 whose label set is a subset of $\{1, \ldots, t\}$. By Lemma 3.2, the number of such equivalence classes is finite and upper bounded by a function of $|\psi|$ and t. Thus, the number of equivalence classes of $\approx$ is also upper bounded by a function of $|\psi|$ and t. Let $\mathcal{S}$ be a set of minimum size representatives of the equivalence classes of $\approx$, and let $c_1 = \max_{\alpha \in \mathcal{S}} |V(G_\alpha)|$.

Let G, Y, and X be a graph and vertex sets as in the statement of the lemma. We now define the sets $B = \partial_G(X)$, $R = (V(G) \backslash X) \cup B$, and the boundaried structures (G_X, Y_X) and (G_R, Y_R) as follows. The boundaried graphs G_X and G_R are just $G[X]$ and $G[R]$, respectively. Both have boundary B, with labels from $\{1, \ldots, t\}$ such that $G_X \oplus G_R = G$. Similarly, $Y_X = Y \cap X$, whereas $Y_R = Y \backslash X$, such that $(G, Y) = (G_X, Y_X) \oplus (G_R, Y_R)$. Observe that $|V(G_X)| = |X| > c_1$.

Hardwired in its source code, our algorithm has a table that for every boundaried structure α of type (graph, vertex set) with label sets from $\{1, \ldots, t\}$ and $|V(G_\alpha)| \leq 2c_1$ contains the $\beta \in \mathcal{S}$ such that $\beta \approx \alpha$. The size of this table is a constant that depends only on $|\psi|$ and t. The algorithm looks up in the table and finds the representative $(G'_X, Y'_X) \in \mathcal{S}$ such that $(G'_X, Y'_X) \approx (G_X, Y_X)$. By construction, we have $|V(G'_X)| \leq c_1 < |V(G_X)|$. The algorithm outputs the instance $((G', Y'), k)$, where $(G', Y') = (G'_X, Y'_X) \oplus (G_R, Y_R)$. Since $|V(G'_X)| < |V(G_X)|$, it follows that $|V(G')| < |V(G')|$, and it remains to argue that the instances $((G, Y), k)$ and $((G', Y'), k)$ are equivalent.

Suppose that $((G, Y), k)$ is a YES-instance, and let $S \subseteq Y$, $|S| \leq k$ ($|S| = k$ for p-EQ-CMSO$[\psi]$) be such that $(G, S) \models \psi$. Let $S_X = X \cap S$ and $S_R = S \backslash X$. Observe that $(G_X, S_X) \oplus (G_R, S_R) = (G, S)$, $S_X = S_X \cap X \subseteq Y \cap X \subseteq \partial(X)$ and that $|S_X| + |S_R| = |S|$. Let S'_X be the subset of $\delta(G'_X)$ such that $\lambda_{G'_X}(S'_X) = \lambda_{G_X}(S_X)$. Since $S_X \subseteq Y_X \subseteq \delta(G_X)$, it follows that $|S_X| = |S'_X|$. Furthermore, property 3 of $\approx$ yields that $(G_X, S_X) \equiv_{\sigma_\psi} (G'_X, S'_X)$. Let $S' = S'_X \cup S_R$ (formally, S'_X and S_R are vertex sets of different graphs, so we set $S' = ((G'_X, S'_X) \oplus (G_R, S_R))[2]$). Since $S_R \cap \delta(G_R) = \emptyset$, we have that $|S'| = |S'_X| + |S_R| = |S_X| + |S_R| = |S|$. Thus, if $|S| \leq k$, then $|S'| \leq k$, whereas if $|S| = k$, then $|S'| = k$. Finally, we observe that

$$\begin{aligned} &(G', S') \models \psi \\ \iff &(G'_X, S'_X) \oplus (G_R, S_R) \models \psi \\ \iff &(G_X, S_X) \oplus (G_R, S_R) \models \psi \\ \iff &(G, S) \models \psi \iff \text{true}. \end{aligned}$$

This concludes the forward direction of the proof. The reverse direction is symmetric. □

Lemmata 5.3, 5.4, and 5.6 together yield a reduction rule for all annotated p-MIN-CMSO[ψ] problems.

LEMMA 5.7. *Let Π^α be an annotated p-MIN-CMSO[ψ] problem. Then for every t, there is a constant $c_2 > 0$ (depending only on $|\psi|$ and t) and an algorithm that, given an instance $((G, Y), k)$ of Π^α and a t-protrusion X with $|X| > c_2 k$, outputs, in time $O(|X|)$, an equivalent instance $((G^*, Y^*), k)$ of Π^α such that $|V^*| < |V|$.*

PROOF. Let $|\partial_G(X)| = t$. The algorithm starts by applying Lemma 5.3 to X and producing an equivalent instance $((G, Y'), k)$, where $|Y' \cap X| \leq ak$ for some constant a depending only on $|\psi|$ and t. Let $Z = Y' \cap X$. The next step is to apply Lemma 5.4 and construct a collection $\mathcal{Q}$ of $(4t+2)$-protrusions such that $X = \bigcup_{Q \in \mathcal{Q}} Q$, $Z \cap Q \subseteq \partial_G(Q)$ for each $Q \in \mathcal{Q}$, and $|\mathcal{Q}| \leq bk$ for some constant b depending only on $|\psi|$ and t. Let c_1 be the constant as guaranteed by Lemma 5.6 when applied on $(8t+4)$-protrusions, and set $c_2 = c_1 \cdot b$. By the pigeon-hole principle, some $(4t+2)$-protrusion Q in $\mathcal{Q}$ has size at least $|X|/bk > c_1$. We apply Lemma 5.5 and obtain a $(8t+4)$-protrusion $Q' \subseteq Q$ such that $Z \cap Q' \subseteq \partial(Q')$ and $c_1 < |Q'| \leq 2c_1$. Finally, we apply the algorithm of Lemma 5.6 on Q' and construct an equivalent instance of Π^α, as required. □

We are now ready to prove the following result.

LEMMA 5.8. *Every annotated p-MIN-CMSO[ψ] problem has the protrusion replacement property $\mathbf{A}$ for $a = 1$.*

PROOF. According to the terminology that we introduced in Section 4, we have to prove that there exists an $(f, 1)$-protrusion replacement family $\mathcal{A}$ for Π^α. Indeed, this directly follows from Lemma 5.7 if we define $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that for every r, $f(r)$ is the constant c_2 of Lemma 5.7. □

5.3. Protrusion Replacement for Annotated *p*-EQ-CMSO[ψ] Problems

In this section, we give a reduction rule for annotated p-EQ-CMSO[ψ] problems. The rule is very similar to the one for the p-MIN-CMSO[ψ] problems described in the previous section. The main difference between the two problem variants is that we now need to keep track of solutions of every possible size between 0 and k, instead of just the smallest one. Because of this, we require the protrusion to contain at least ck^2 vertices instead of ck vertices to be able to reduce it. We start by proving adaptations of Lemmata 5.2 and 5.3 to p-EQ-CMSO[ψ] problems.

LEMMA 5.9. *There is an algorithm that, given two boundaried structures (G_X, Y_X) and (G_R, S_R) of type* (graph, vertex set)*, a CMSO sentence ψ, and nonnegative integer k, finds a $S_X \subseteq Y_X$ of size k such that $(G_X, S_X) \oplus (G_R, S_R) \models \psi$ or concludes that no such set exists in time $|V(G_X \oplus G_R)| \cdot f(|\psi|, \mathbf{tw}(G_X \oplus G_R))$.*

PROOF. Let $(G', Y', S'_R) = (G_X, Y_X, \emptyset) \oplus (G_R, \emptyset, S_R)$. Finding the desired set $S_X \subseteq Y$ now amounts to finding a set $S'_X \subseteq Y'$ of size k such that $(G', S'_X \cup S'_R) \models \psi$. This is easily formulated as Eq-CMSO on Structures and hence may be solved in the desired running time by Proposition 5.1. □

Lemma 5.10. *Let Π^α be an annotated p-*eq*-CMSO[ψ] problem, and let t be an integer. Then there exists an algorithm that, given an instance $((G, Y), k)$ of Π^α and a t-protrusion X of G, outputs, in time $O(k|X|)$, an equivalent instance $((G, Y'), k)$ of Π^α, where $|Y' \cap X| = O(k^2)$ and $Y' \subseteq Y$.*

Proof. The proof of the lemma starts exactly as in the proof of Lemma 5.3. For a CMSO sentence ψ defining Π^α, Lemma 3.2 implies that the canonical equivalence relation $\equiv_{\sigma_\psi}$ has finitely many equivalence classes on the set of boundaried structures of arity 2 with label set $\{1, \ldots, t\}$. We denote by $\mathbf{MinRep}(\psi, t)$ a set containing a representative (a boundaried structure of arity 2) for each equivalence class of $\equiv_{\sigma_\psi}$ with the minimum number of vertices in the graph of a structure. For given G, Y, and X, we define the sets $B = \partial_G(X)$, $R = (V(G)\backslash X) \cup B$, and the boundaried structures (G_X, Y_X) and (G_R, Y_R) as follows. The boundaried graphs G_X and G_R are just $G[X]$ and $G[R]$, respectively. Both have boundary B, with labels from $\{1, \ldots, t\}$ such that $G_X \oplus G_R = G$. Similarly, $Y_X = Y \cap X$, whereas $Y_R = Y \backslash X$, such that $(G, Y) = (G_X, Y_X) \oplus (G_R, Y_R)$.

For every structure $\alpha = (G_R^\alpha, S_R^\alpha) \in \mathbf{MinRep}(\psi, t)$ and every integer $i \leq k$, we use Lemma 5.9 to find a set $S_X^{\alpha,i} \subseteq Y_X$ such that $|S_X^{\alpha,i}| = i$ and $(G_X, S_X^\alpha) \oplus \alpha \models \psi$. If no such set exists, we set $S_X^{\alpha,i} = \emptyset$. Since $|\mathbf{MinRep}(\psi, t)|$ and the size of each structure in $\mathbf{MinRep}(\psi, t)$ depends only on ψ and t, and the treewidth of $G[X]$ is at most t, this takes time $O(k|X|)$. Now, define

$$Y'_X = \bigcup_{\substack{\alpha \in \mathbf{MinRep}(\psi,t) \\ i \leq k}} S_X^{\alpha,i}.$$

We set $Y' = Y'_X \cup Y_R$ (formally, Y'_X and Y_R are vertex sets of different graphs, so actually $Y' = ((G_X, Y'_X) \oplus (G_R, Y_R))[2]$). Since $|\mathbf{MinRep}(\psi, t)|$ depends only on $|\psi|$ and t, the construction of Y' implies that $|Y' \cap X| = O(k^2)$.

To complete the proof, it remains to show that $((G, Y'), k) \in \Pi^\alpha$ if and only if $((G, Y), k) \in \Pi^\alpha$. For the forward direction, we have that $Y' \subseteq Y$, and hence feasible solutions to $((G, Y'), k)$ are also feasible for $((G, Y), k)$. We now turn to proving the reverse direction. Let $S \subseteq Y$, $|S| = k$ be such that $(G, S) \models \psi$. Let $S_X = X \cap S$ and $S_R = S \backslash X$. Observe that $(G_X, S_X) \oplus (G_R, S_R) = (G, S)$ and that $|S_X| + |S_R| = |S| = k$. Choose $\alpha = (G_R^\alpha, S_R^\alpha) \in \mathbf{MinRep}(\psi, t)$ such that $\alpha \equiv_{\sigma_\psi} (G_R, S_R)$. Set $i = |S_X|$, and let $S_X^{\alpha,i} \subseteq Y_X$ be the set computed for α and i in the previous paragraph. The existence of $S_X^{\alpha,i}$ of size i is guaranteed by the fact that

$$(G_X, S_X) \oplus \alpha \models \psi \iff (G_X, S_X) \oplus (G_R, S_R) \models \psi \iff \text{true}.$$

By construction, $S_X^{\alpha,i} \subseteq Y'_X$. Let $S' = S_X^{\alpha,i} \cup S_R$ (again, formally, $S_X^{\alpha,i}$ and S_R are vertex sets of different graphs, so actually $S' = ((G_X, S_X^{\alpha,i}) \oplus (G_R, S_R))[2]$). We have that $S' \subseteq Y'$. Further, since $S_R \cap \delta(G_R) = \emptyset$, we have that $|S'| = |S_X^{\alpha,i}| + |S_R| = |S_X| + |S_R| = |S| = k$. Finally, we observe that

$$\begin{aligned} & (G, S') \models \psi \\ \iff & (G_X, S_X^{\alpha,i}) \oplus (G_R, S_R) \models \psi \\ \iff & (G_X, S_X^{\alpha,i}) \oplus \alpha \models \psi \\ \iff & \text{true}. \end{aligned}$$

This concludes the proof. □

LEMMA 5.11. *Let Π^α be an annotated p-EQ-CMSO[ψ] problem. Then for every t, there is a constant $c_2 \in \mathbb{Z}^+$ (depending only on $|\psi|$ and t) and an algorithm that, given an instance $((G, Y), k)$ of Π^α and a t-protrusion X with $|X| > c_2k^2$, outputs, in time $O(k \cdot |X|)$, an equivalent instance $((G^*, Y^*), k)$ of Π^α such that $|V^*| < |V|$.*

PROOF. The algorithm starts by applying Lemma 5.10 to X and producing an equivalent instance $((G, Y'), k)$, where $|Y' \cap X| \leq ak^2$ for some constant a depending only on $|\psi|$ and t. Let $Z = Y' \cap X$. The next step is to apply Lemma 5.4 and construct a collection $\mathcal{Q}$ of $(4t + 2)$-protrusions such that $X = \bigcup_{Q \in \mathcal{Q}} Q$, $Z \cap Q \subseteq \partial_G(Q)$ for each $Q \in \mathcal{Q}$, and $|\mathcal{Q}| \leq bk^2$ for some constant b depending only on $|\psi|$ and t. Let c_1 be the constant as guaranteed by Lemma 5.6 when applied on $(8t + 4)$-protrusions, and set $c_2 = c_1 \cdot b$. By the pigeon-hole principle, some $(4t + 2)$-protrusion Q in $\mathcal{Q}$ has size at least $|X|/bk^2 > c_1$. We apply Lemma 5.5 and obtain a $(8t + 4)$-protrusion $Q' \subseteq Q$ such that $Z \cap Q' \subseteq \partial(Q')$ and $c_1 < |Q'| \leq 2c_1$. Finally, we apply the algorithm of Lemma 5.6 on Q' and construct an equivalent instance of Π^α, as required. □

We are now ready to prove the following result.

LEMMA 5.12. *Every annotated p-EQ-CMSO[ψ] problem has the protrusion replacement property **A** for $a = 2$.*

PROOF. According to the terminology that we introduced in Section 4, we have to prove that there exists an $(f, 2)$-protrusion replacement family $\mathcal{A}$ for Π^α. Indeed, this directly follows from Lemma 5.11 if we define $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that for every r, $f(r)$ is the constant c_2 in the proof of the same lemma. □

5.4. Protrusion Replacement for Annotated *p*-MAX-CMSO[ψ] Problems

We now give a reduction rule for annotated p-MAX-CMSO[ψ] problems. The rule is still similar to the ones described in the two previous sections but differs more from the p-MIN-CMSO[ψ] problems than p-EQ-CMSO[ψ] did. We start by proving a variant of Lemma 5.2 for p-MAX-CMSO[ψ] problems.

LEMMA 5.13. *There is an algorithm that, given two boundaried structures (G_X, Y_X) and (G_R, S_R) of type* (graph, vertex set) *and a CMSO sentence ψ, finds a set $S_X \subseteq V(G_X)$ such that $(G_X, S_X) \oplus (G_R, S_R) \models \psi$ and $|S_X \cap Y_X|$ is maximized. The running time of the algorithm is $|V(G_X \oplus G_R)| \cdot f(|\psi|, \mathbf{tw}(G_X \oplus G_R))$.*

PROOF. Let $(G', Y', S'_R, V') = (G_X, Y_X, \emptyset, V(G_X)) \oplus (G_R, \emptyset, S_R, \emptyset)$. Finding the desired set S_X now amounts to finding a set $S'_X \subseteq V'$ such that $(G', S'_X \cup S'_R) \models \psi$ and $|S'_X \cap Y'|$ is maximized. This is easily formulated as MAX-CMSO ON STRUCTURES and hence may be solved in the desired running time by Proposition 5.1. □

LEMMA 5.14. *Let Π^α be an annotated p-MAX-CMSO[ψ] problem, and let t be an integer. There exists an algorithm that, given an instance $((G, Y), k)$ of Π^α and a t-protrusion X of G, outputs, in time $O(|X|)$, an equivalent instance $((G, Y'), k)$ of Π^α, where $|Y' \cap X| = O(k)$ and $Y' \subseteq Y$.*

PROOF. By Lemma 3.2, for a CMSO sentence ψ defining Π^α, the canonical equivalence relation $\equiv_{\sigma_\psi}$ has finitely many equivalence classes on the set of boundaried structures of arity 2 with label set $\{1, \ldots, t\}$. As in proofs of Lemmata 5.3 and 5.10, we define the following objects. We set $\mathbf{MinRep}(\psi, t)$ to be a set containing a representative (a boundaried structure of arity 2) for each equivalence class of $\equiv_{\sigma_\psi}$ with the minimum number of vertices in the graph of a structure. In addition, for G, Y, and X, we define sets $B = \partial_G(X)$, $R = (V(G)\backslash X) \cup B$, and the boundaried structures (G_X, Y_X) and (G_R, Y_R) as follows. Again, the boundaried graphs $G_X = G[X]$ and $G_R = G[R]$ have boundary

B with labels from $\{1, \ldots, t\}$ such that $G_X \oplus G_R = G$. Similarly, $Y_X = Y \cap X$, whereas $Y_R = Y \backslash X$, such that $(G, Y) = (G_X, Y_X) \oplus (G_R, Y_R)$.

By making use of Lemma 5.13, for every structure $\alpha = (G_R^\alpha, S_R^\alpha) \in \mathbf{MinRep}(\psi, t)$, we find a set $S_X^\alpha \subseteq V(G_X)$ such that $(G_X, S_X^\alpha) \oplus \alpha \models \psi$ and $|S_X \cap Y_X|$ is maximized. Since $|\mathbf{MinRep}(\psi, t)|$ and the size of each structure in $\mathbf{MinRep}(\psi, t)$ depends only on $|\psi|$ and t, and the treewidth of $G[X]$ is at most t, this takes time $O(|X|)$. If $|S_X^\alpha \cap Y_X| \leq k$, let $\hat{S}_X^\alpha = S_X^\alpha \cap Y_X$. On the other hand, if $|S_X^\alpha \cap Y_X| > k$, set $\hat{S}_X^\alpha$ to be a set of arbitrarily chosen k vertices from $S_X^\alpha \cap Y_X$. Now, define

$$Y'_X = \bigcup_{\alpha \in \mathbf{MinRep}(\psi,t)} \hat{S}_X^\alpha.$$

We set $Y' = Y'_X \cup Y_R$ (formally, Y'_X and Y_R are vertex sets of different graphs, so actually $Y' = ((G_X, Y'_X) \oplus (G_R, Y_R))[2]$). Since $|\mathbf{MinRep}(\psi, t)|$ depends only on $|\psi|$ and t, the construction of Y' implies that $|Y' \cap X| = O(k)$.

To complete the proof, it remains to show that $((G, Y'), k) \in \Pi^\alpha$ if and only if $((G, Y), k) \in \Pi^\alpha$. For the forward direction, we have that $Y' \subseteq Y$, and hence for any set $S \subseteq V(G)$ such that $(G, S) \models \psi$ and $|S \cap Y'| \geq k$, we also have that $|S \cap Y| \geq k$. We now turn to proving the reverse direction. Let $S \subseteq V(G)$, $|S \cap Y| \geq k$ be such that $(G, S) \models \psi$. Let $S_X = X \cap S$ and $S_R = S \backslash X$. Observe that $(G_X, S_X) \oplus (G_R, S_R) = (G, S)$ and that $|S_X \cap Y_X| + |S_R \cap Y_R| = |S \cap Y| \geq k$. Choose $\alpha = (G_R^\alpha, S_R^\alpha) \in \mathbf{MinRep}(\psi, t)$ such that $\alpha \equiv_{\sigma_\psi} (G_R, S_R)$. Let $S_X^\alpha \subseteq V(G_X)$ be the set computed for α in the previous paragraph. Since

$$(G_X, S_X) \oplus \alpha \models \psi \iff (G_X, S_X) \oplus (G_R, S_R) \models \psi \iff \textsf{true},$$

it follows that $|S_X^\alpha \cap Y_X| \geq |S_X \cap Y_X|$. Furthermore, we have that $|S_X^\alpha \cap Y'_X| \geq |\hat{S}_X^\alpha| \geq \min(|S_X \cap Y_X|, k)$.

Let $S' = S_X^\alpha \cup S_R$ (again, formally, S_X^α and S_R are vertex sets of different graphs, so actually $S' = ((G_X, S_X^\alpha) \oplus (G_R, S_R))[2]$). We have that

$$|S' \cap Y'| \geq |S_X^\alpha \cap Y'_X| + |S_R \cap Y_R| \geq \min(|S_X \cap Y_X|, k) + |S_R \cap Y_R| \geq \min(|S \cap Y|, k) \geq k.$$

Finally, we observe that

$$\begin{aligned} & (G, S') \models \psi \\ \iff & (G_X, S_X^\alpha) \oplus (G_R, S_R) \models \psi \\ \iff & (G_X, S_X^\alpha) \oplus \alpha \models \psi \\ \iff & \textsf{true}. \end{aligned}$$

This concludes the proof. □

LEMMA 5.15. *Let Π^α be an annotated p-MAX-CMSO$[\psi]$ problem. Then for every integer t, there is a $c_1 \in \mathbb{Z}^+$ (depending only on $|\psi|$ and t) and an algorithm, that given an instance $((G, Y), k)$ of Π^α and a t-protrusion X of G, where $c_1 < |X| \leq 2c_1$ and $X \cap Y \subseteq \partial_G(X)$, outputs, in time $O(|X|)$, an equivalent instance $((G^*, Y^*), k)$ of Π^α such that $|V(G')| < |V(G)|$.*

PROOF. Let ψ be the CMSO sentence mentioned in the definition of Π^α. By Lemma 3.2, the canonical equivalence relation $\equiv_{\sigma_\psi}$ has finitely many equivalence classes on the set of boundaried structures of arity 2 with label set $\{1, \ldots, t\}$. Let $\mathbf{MinRep}(\psi, t)$ be a set containing a representative (a boundaried structure of arity 2) for each equivalence class of $\equiv_{\sigma_\psi}$ with the minimum number of vertices in the graph of a structure. We now define an equivalence relation $\approx$ between boundaried structures $\alpha = (G_\alpha, Y_\alpha)$ of type (graph, vertex set) that satisfy $Y_\alpha \subseteq \delta(G_\alpha)$. Let $\alpha_1 = (G_1, Y_1)$ and $\alpha_2 = (G_2, Y_2)$ be two boundaried structures with labeling functions $\lambda_1 : \delta(G_1) \to \{1, \dots, t\}$ and $\lambda_2 : \delta(G_2) \to \{1, \dots, t\}$, respectively, such that $Y_1 \subseteq \delta(G_1)$ and $Y_2 \subseteq \delta(G_2)$. We say that $\alpha_1 \approx \alpha_2$ if the following conditions are satisfied:

(1) $\Lambda(G_1) = \Lambda(G_2)$;
(2) $\lambda_1(Y_1) = \lambda_2(Y_2)$;
(3) for every $S_1 \subseteq V(G_1)$ there is an $S_2 \subseteq V(G_2)$ such that $\lambda_1(S_1 \cap \delta(G_1)) = \lambda_2(S_2 \cap \delta(G_2))$, and $(G_1, S_1) \equiv_{\sigma_\psi} (G_2, S_2)$; and
(4) for every $S_2 \subseteq V(G_2)$ there is an $S_1 \subseteq V(G_1)$ such that $\lambda_1(S_1 \cap \delta(G_1)) = \lambda_2(S_2 \cap \delta(G_2))$, and $(G_1, S_1) \equiv_{\sigma_\psi} (G_2, S_2)$.

Notice that $\approx$ is an equivalence relation. Further, consider two boundaried structures $\alpha_1 = (G_1, Y_1)$ and $\alpha_2 = (G_2, Y_2)$ such that $\Lambda(G_1) = \Lambda(G_2)$, $\lambda_1(Y_1) = \lambda_2(Y_2)$, and for each subset $L \subseteq \{1, \dots, t\}$, the sets

$$\{\beta \in \mathbf{MinRep}(\psi, t) : \exists S_1 \subseteq V(G_1), \lambda_1(S_1 \cap \delta(G_1)) = L \wedge (G_1, S_1) \equiv_{\sigma_\psi} \beta\}$$

and

$$\{\beta \in \mathbf{MinRep}(\psi, t) : \exists S_2 \subseteq V(G_2), \lambda_2(S_2 \cap \delta(G_2)) = L \wedge (G_2, S_2) \equiv_{\sigma_\psi} \beta\}$$

are the same. It is easy to verify that in this case, $(G_1, Y_1) \approx (G_2, Y_2)$. Thus, the number of equivalence classes of $\approx$ is upper bounded by a function of $|\psi|$ and t. Let $\mathcal{S}$ be a set of minimum size representatives of the equivalence classes of $\approx$, and let $c_1 = \max_{\alpha \in \mathcal{S}} |V(G_\alpha)|$.

Let G, Y and X be a graph and vertex sets as in the statement of the lemma. We now define the sets $B = \partial_G(X)$, $R = (V(G) \setminus X) \cup B$, and the boundaried structures (G_X, Y_X) and (G_R, Y_R) as follows. The boundaried graphs $G_X = G[X]$ and $G_R = G[R]$ have boundary B with labels from $\{1, \dots, t\}$ such that $G_X \oplus G_R = G$. We define $Y_X = Y \cap X$ and $Y_R = Y \setminus X$ such that $(G, Y) = (G_X, Y_X) \oplus (G_R, Y_R)$. Observe that $|V(G_X)| = |X| > c_1$.

Hardwired in its source code, our algorithm has a table that for every boundaried structure α of type (graph, vertex set) with label sets from $\{1, \dots, t\}$ and $|V(G_\alpha)| \leq 2c_1$ contains the $\beta \in \mathcal{S}$ such that $\beta \approx \alpha$. The size of this table is a constant that depends only on $|\psi|$ and t. The algorithm looks up in the table and finds the representative $(G'_X, Y'_X) \in \mathcal{S}$ such that $(G'_X, Y'_X) \approx (G_X, Y_X)$. By construction, we have $|V(G'_X)| \leq c_1 < |V(G_X)|$. The algorithm outputs the instance $((G', Y'), k)$, where $(G', Y') = (G'_X, Y'_X) \oplus (G_R, Y_R)$. Since $|V(G'_X)| < |V(G_X)|$, it follows that $|V(G')| < |V(G')|$, and it remains to argue that the instances $((G, Y), k)$ and $((G', Y'), k)$ are equivalent.

Suppose that $((G, Y), k)$ is a YES-instance, and let $S \subseteq V(G)$, $|S \cap Y| \geq k$ be such that $(G, S) \models \psi$. Let $S_X = X \cap S$ and $S_R = S \setminus X$. Observe that $(G_X, S_X) \oplus (G_R, S_R) = (G, S)$, $S_X \cap Y_X \subseteq \delta(G_X)$, and that $|S_X \cap Y_X| + |S_R \cap Y_R| = |S \cap Y|$. Let S'_X be a subset of $V(G'_X)$ such that $\lambda_{G'_X}(S'_X \cap \delta(G'_X)) = \lambda_{G_X}(S_X \cap \delta(G_X))$ and $(G'_X, S'_X) \equiv_{\sigma_\psi} (G_X, S_X)$. The existence of such a set S'_X is implied by property (3) of $\approx$. Since $Y_X \subseteq \delta(G_X)$, $Y'_X \subseteq \delta(G'_X)$, $\Lambda_{G_X}(Y_X) = \Lambda_{G'_X}(Y'_X)$, and $\Lambda_{G_X}(S_X \cap \delta(G_X)) = \Lambda_{G'_X}(S'_X \cap \delta(G'_X))$, we have that $|S_X \cap Y_X| = |S'_X \cap Y'_X|$.

Let $S' = S'_X \cup S_R$ (formally, S'_X and S_R are vertex sets of different graphs, so we set $S' = ((G'_X, S'_X) \oplus (G_R, S_R))[2]$). Since $S_R \cap \delta(G_R) = \emptyset$, we have that $|S' \cap Y'| = |S'_X \cap Y'_X| + |S_R \cap Y_R| = |S_X \cap Y_X| + |S_R \cap Y_R| = |S \cap Y|$. Thus, if $|S \cap Y| \geq k$, then $|S' \cap Y'| \geq k$. Finally, we observe that

$$\begin{aligned} & (G', S') \models \psi \\ \iff & (G'_X, S'_X) \oplus (G_R, S_R) \models \psi \\ \iff & (G_X, S_X) \oplus (G_R, S_R) \models \psi \\ \iff & (G, S) \models \psi \iff \text{true}. \end{aligned}$$

This concludes the forward direction of the proof. The reverse direction is symmetric but uses property 4 of $\approx$ rather than property 3. □

LEMMA 5.16. *Let Π^α be an annotated p-MAX-CMSO[ψ] problem. Then for every t, there is a constant $c_2 > 0$ (depending only on ψ and t) and an algorithm that, given an instance $((G, Y), k)$ of Π^α and a t-protrusion X with $|X| > c_2 k$, outputs, in time $O(|X|)$, an equivalent instance $((G, Y^*), k)$ of Π^α such that $|V^*| < |V|$.*

PROOF. Let $|\partial_G(X)| = t$. The algorithm starts by applying Lemma 5.14 to X and producing an equivalent instance $((G, Y'), k)$ where $|Y' \cap X| \leq ak$ for some constant a depending only on $|\psi|$ and t. Let $Z = Y' \cap X$. The next step is to apply Lemma 5.4 and construct a collection $\mathcal{Q}$ of $(4t+2)$-protrusions such that $X = \bigcup_{Q \in \mathcal{Q}} Q$, $Z \cap Q \subseteq \partial_G(Q)$ for each $Q \in \mathcal{Q}$, and $|\mathcal{Q}| \leq bk$ for some constant b depending only on $|\psi|$ and t. Let c_1 be the constant as guaranteed by Lemma 5.15 when applied on $(8t+4)$-protrusions, and set $c_2 = c_1 \cdot b$. By the pigeon-hole principle, some $(4t+2)$-protrusion Q in $\mathcal{Q}$ has size at least $|X|/bk > c_1$. We apply Lemma 5.5 and obtain a $(8t+4)$-protrusion $Q' \subseteq Q$ such that $Z \cap Q' \subseteq \partial(Q')$ and $c_1 < |Q'| \leq 2c_1$. Finally, we apply the algorithm of Lemma 5.15 on Q' and construct an equivalent instance of Π^α, as required. □

Now we show the following result.

LEMMA 5.17. *Every annotated p-MAX-CMSO[ψ] has the protrusion replacement property* **A** *for $a = 1$.*

PROOF OF LEMMA 5.17. According to the terminology that we introduced in Section 4, we have to prove that there exists an $(f, 1)$-protrusion replacement family $\mathcal{A}$ for Π. Indeed, this directly follows from Lemma 5.16 if we define $f : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ such that for every r, $f(r)$ is the constant c_2 in the statement of the same lemma. □

5.5. A Protrusion Replacement Family Based for Problems That Have FII

In the previous sections, we gave reduction rules for annotated p-MIN/EQ/MAX-CMSO[ψ] problems. These reduction rules, together with the results proved later in this article, will give quadratic or cubic kernels for the problems in question. However, for many problems, a linear kernel is possible. In this section, we provide reduction rules for graph problems that have FII. These reduction rules will yield linear kernels. The main reduction lemma is presented next.

LEMMA 5.18. *Let Π be a problem that has FII. Then for every $t \in \mathbb{Z}^+$, there exists a $c \in \mathbb{Z}^+$ (depending on Π and t) and an algorithm that, given an instance (G, k) of Π and a t-protrusion X in G with $|X| > c$, outputs, in time $O(|X|)$, an equivalent instance (G^*, k^*) of Π where $|V(G^*)| < |V(G)|$ and $k^* \leq k$.*

PROOF. Recall that we denote by $\mathcal{S}_{\subseteq[2t+1]}$ a set of (progressive) representatives for $\equiv_\Pi$ restricted to boundaried graphs with label sets from $\{1, \ldots, 2t+1\}$. Let

$$c = \max\left\{|V(Y)| \,\middle|\, Y \in \mathcal{S}_{\subseteq[2t+1]}\right\}.$$

Hardwired in its source code, our algorithm has a table that stores for each boundaried graph G_Y in $\mathcal{F}_{\subseteq[2t+1]}$ on at most $2c$ vertices a boundaried graph $G'_Y \in \mathcal{S}_{\subseteq[2t+1]}$ and a constant $\mu \leq 0$ such that $G_Y \equiv_\Pi G'_Y$, and specifically

$$\forall (F, k) \in \mathcal{F} \times \mathbb{Z} : (G_Y \oplus F, k) \in \Pi \iff (G'_Y \oplus F, k + \mu) \in \Pi. \quad (30)$$

The existence of such a constant $\mu \leq 0$ is guaranteed by the fact that $\mathcal{S}_{\subseteq[2t+1]}$ is a set of progressive representatives.

We now apply Lemma 5.5 and find a $(2t+1)$-protrusion Y of G where $c < |Y| \leq 2c$. Split G into two boundaried graphs $G_Y = G[Y]$ and $G_R = G[(V(G)\backslash Y) \cup \partial(Y)]$ as follows. Both G_R and G_Y have boundary $\partial(Y)$, and since $|\partial(Y)| \leq 2t+1$, we may label the boundaries of G_Y and G_R with labels from $[2t+1]$ such that $G = G_Y \oplus G_R$. As $c < |V(G_Y)| \leq 2c$, the algorithm can look up in its table and find a $G'_Y \in \mathcal{S}_{\subseteq[2t+1]}$ and a constant μ such that $G_Y \equiv G'_Y$ and G_Y, G'_Y and μ satisfy Equation (30). The algorithm outputs

$$(G^*, k^*) = (G'_Y \oplus G_R, k + \mu).$$

Since $|V(G'_Y)| \leq c < |V(G_Y)|$ and $k^* \leq k + \mu \leq k$, it remains to argue that the instances (G, k) and (G^*, k^*) are equivalent. However, this is directly implied by Equation (30). □

We are now in position to prove the lemma.

LEMMA 5.19. *Every parameterized graph problem* Π *that has FII has the protrusion replacement property* $\mathbf{A}$ *for* $a = 0$.

PROOF. According to the terminology that we introduced in Section 4, we have to prove that there exists an $(f, 0)$-protrusion replacement family $\mathcal{A}$ for Π. Indeed, this directly follows from Lemma 5.18 if we define $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that for each r, $f(r)$ is the constant c in the statement of the same lemma. □

6. COMBINATORIAL RESULTS

We start this section with some necessary definitions from graph theory.

6.1. Definitions from Graph Theory

Let $e = \{u, v\}$ be an edge of a graph $G = (V, E)$. We obtain the graph G/e by *contracting* e. This means that the edge e is removed and its endpoints u, v are merged into a new vertex v_e such that each edge incident with either u or v is incident with v_e. Note that loops and multiple edges can appear as a result of edge contractions. More formally, let f be a function mapping u, v to v_e and all remaining vertices in $V\backslash\{u, v\}$ to itself. The contraction of e results in a new graph $G/e = (V', E')$, where $V' = (V\backslash\{u, v\}) \cup \{v_e\}$, $E' = E\backslash\{e\}$, and for every $w \in V$, $w' = f(w) \in V'$ is incident with an edge $e' \in E'$ if and only if the corresponding edge $e \in E$ is incident with w in G. When we have to remain in the class of simple graphs, loops and multiple edges resulting by contractions are deleted.

A graph H is a *minor* of a graph G; we write $H \preceq G$ if H can be obtained by contracting some edges of a subgraph of G. A graph class $\mathcal{C}$ is *minor closed* if every minor of every graph in $\mathcal{C}$ also belongs to $\mathcal{C}$. A minor-closed graph class $\mathcal{C}$ is H*-minor-free* if $H \notin \mathcal{C}$.

Given a graph $G = (V, E)$, we define the (normal) *distance* between two of its vertex sets X and Y as the shortest path distance between them (i.e., the minimum length of a path with endpoints in X and Y) and denote it by $\mathbf{dist}_G(X, Y)$. Given a set $S \subseteq V$ of vertices, we denote by $\mathbf{B}^r_G(S)$ the set of all vertices that are within distance at most r from some vertex of S in G.

We also need some notions from topological graph theory. All concepts that we do not define here can be found in Mohar and Thomassen [2001]. The *Euler genus* $\mathbf{eg}(\Phi)$ of a nonorientable surface Φ is equal to the nonorientable genus $\tilde{g}(\Phi)$ (or the crosscap number). The Euler genus $\mathbf{eg}(\Phi)$ of an orientable surface Φ is $2g(\Phi)$, where $g(\Phi)$ is the orientable genus of Φ. We say that a graph G is Φ*-embedded* if it is accompanied with an embedding of the graph into Φ. We also sometimes refer to an embedding as to a drawing of G in Φ. We treat edges and loops (in some proofs, we will also allow loops and multiple edges) as subsets of the surface Φ that are homeomorphic to the open interval $(0, 1)$. We define the endpoints of an edge e as the set of points of Φ that are in the closure of e but not in e. We call a *face* of a Φ-embedded graph $G = (V, E)$ any connected component of $\Phi\backslash(E \cup V)$. All embeddings that we consider are *2-cell embeddings*, which are embeddings with each face being homeomorphic to a disk.

For a Φ-embedded connected graph G, the relation between the number of its vertices n, the number of edges m, the number of faces f, and the Euler genus is given by the Euler's formula (e.g., see Section 4.4 in Mohar and Thomassen [2001]):

$$n - m + f = 2 - \mathbf{eg}(\Phi). \tag{31}$$

Given a Φ-embedded graph G, we define its *radial graph* R_G as an embedded graph whose vertices are the vertices and the faces of G (each face f of G is represented by a point v_f in it). Roughly, each point v_f is adjacent to all vertices v incident to f. However, a face can be incident "several times" with the same vertex, and R_G can have multiple edges. For a point v_f in the face f and vertex v incident with f, we draw a maximum number of multiple edges in f such that for every pair of multiple edges e and e', the open disc bounded by these edges intersects G. Thus, R_G is a bipartite multigraph embedded in the same surface as G. Radial graphs provide an alternative way of viewing radial distance defined in Section 1: the radial distance of a pair of vertices in G corresponds to their normal distance in R_G. The relation between radial and normal metrics is captured by the following observation.

OBSERVATION 3. *If G is a Φ-embedded graph, then for every set $S \subseteq V$ and every $r \in \mathbb{Z}^+$, it holds that $\mathbf{B}^r_G(S) \subseteq \mathbf{R}^{2r}_G(S)$.*

6.2. Decomposition Lemma for Coverable Problems

In this section, we show the following decomposition result.

LEMMA 6.1. *Every r-coverable problem has the protrusion decomposition property* **B**.

To prove Lemma 6.1, we have to show that every r-coverable problem satisfies combinatorial property **B** (i.e., admits a protrusion decomposition). Lemma 6.1 follows directly from the following lemma.

LEMMA 6.2. *Let r be a positive integer, and let $G = (V, E)$ be a graph embedded in a surface Φ of Euler genus g that contains a set S of vertices, $|S| \leq k$, such that $\mathbf{R}^r_G(S) = V$. Then G has an $(\alpha k, \beta)$-protrusion decomposition for some constants α and β that depend only on r and g.*

Indeed, since a problem is r-coverable, there is a set S, $|S| \leq r \cdot k$, such that $\mathbf{R}^r_G(S) = V$. Then combinatorial property **B** holds for $c = r \cdot \max\{\alpha, \beta\}$.

The rest of this section is devoted to the proof of Lemma 6.2. We start with definitions and preliminary results. The first observation follows directly from the definition of protrusion decomposition.

OBSERVATION 4. *If G has an $(\alpha k, \beta)$-protrusion decomposition, then the same holds for every subgraph of G.*

The following proposition is a consequence of the result from Eppstein [2000] on the treewidth of graphs with bounded genus and diameter.

PROPOSITION 6.3. *There exists function $f_1 : \mathbb{Z}^+ \times \mathbb{Z}^+ \to \mathbb{Z}^+$ such that if $G = (V, E)$ is a graph of Euler genus at most g such that $V = \mathbf{B}^r_G(v)$ for some $v \in V$, then $\mathbf{tw}(G) \leq f_1(r, g)$.*

For the purposes of the proof of the next lemma, we permit the existence of multiple edges or loops in the embedding. Thus, contracting edges can create multiple edges or loops that we do not delete. We call a face *trivial* if it is incident with at most two edges. We call a loop *empty* if it is the boundary of some face of G.

A *walk* of length λ in a multigraph G is a sequence $\mathcal{C} = v_0e_1v_1 \cdots e_\lambda v_\lambda$ of alternating vertices and edges of G such that for every $i \in \{1, \ldots, \lambda\}$, the vertices v_{i-1} and v_i are the endpoints of edge e_i. Thus, an edge or a vertex can appear many times in a walk. If in the previous definition we additionally demand that $v_0 = v_\lambda$, then the walk is a *closed walk*.

We are ready to proceed with the proof of the lemma.

Proof of Lemma 6.2. We may assume that all faces in the embedding of G are *triangular*, meaning that they are incident with at most three edges and that G is connected. Indeed, if G is not triangulated, we can always triangulate it by adding edges in such a way that it does not increase radial distances between the vertices of G. Then, by Observation 4, we can obtain the required protrusion decomposition for G from the decomposition of its triangulation.

For every $v \in S$, we construct a breadth-first search tree T_v of depth at most r rooted at v. Because $\mathbf{B}^r_G(S) = V$, we have that every vertex of G is in some T_v for some $v \in S$. Although some vertices can be within distance r from several vertices of S, by suitably modifying these trees, we may assume that every vertex is assigned to exactly one tree. That way, the vertex sets of the trees in $\mathcal{T} = \{T_v \mid v \in S\}$ form a partition of V.

We denote by H the graph obtained from G after contracting all edges of the trees in $\mathcal{T}$. Notice that $V(H) = S$, and as G is triangulated, every face of H is incident to at most three edges. We further simplify H as follows:

—as long as there are two edges incident with a trivial face, we delete one of them, and
—As long as there is an empty loop, we delete it.

We denote the resulting graph by $\tilde{H}$. Again, every face of $\tilde{H}$ is incident to at most three edges. In addition, $V(\tilde{H}) = S$.

By making use of Euler's formula (31), we derive that $\tilde{H}$ has at most $2k + 2g - 4$ faces and at most $3k + 3g - 6$ edges. The edges of $\tilde{H}$ can be seen as the edges of G that were not contracted or deleted during the construction of $\tilde{H}$. For every edge $\tilde{e}$ of $\tilde{H}$, we denote by e the corresponding edge of G.

Let $\tilde{e}$ be an edge of $\tilde{H}$ with endpoints $u, v \in S$. Let x_u and x_v be the endpoints of the corresponding edge e in G. If $u = v$, then x_u and x_v are vertices of T_v. If $u \neq v$, then x_u is a vertex of T_u and x_v is a vertex of T_v. In both cases, there are unique paths P_{u,x_u} in T_u and P_{v,x_v} in T_v from u to x_u, and from v to x_v correspondingly. Each of these paths is of length at most r. We set $P_e = P_{u,x_u} \cup \{e\} \cup P_{v,x_v}$. Let us note that if $u = v$, then P_e is a closed walk, and if $u \neq v$, then it is a path. The length of P_e is at most $2r + 1$.

We construct graph $\tilde{G}$ from G by contracting for every edge $\tilde{e}$ of $\tilde{H}$ all edges except e in the corresponding walk P_e. Thus, besides S, the vertex set of $\tilde{G}$ contains all vertices of G not covered by walks P_e. By construction, $\tilde{G}[S] \supseteq \tilde{H}$. We take the drawing of $\tilde{G}$ in Φ and observe that $\tilde{G}[S]$ contains the drawing of $\tilde{H}$ in Φ. In the drawings of $\tilde{G}$ and $\tilde{H}$, every face f of $\tilde{H}$ covers a subset of vertices X_f of $\tilde{G}$. The set X_f is separated in $\tilde{G}$ by the vertices incident with f from the remaining vertices of the graph $\tilde{G}$.

In $\tilde{G}$, every vertex $v \notin S$ belongs to some set X_f. Thus, in G, every vertex is either in some X_f or belongs to some walk P_e. We define vertex subset R_0 of G as the union of the vertices of all walks corresponding to edges of $\tilde{H}$—that is,

$$R_0 = \bigcup_{\tilde{e} \in E(\tilde{H})} V(P_e).$$

Sets R_0 and X_f, $f \in \tilde{F}$, have the following properties.

Claim 1. $|R_0| \leq k + 2r(3k + 3g - 6)$.

Proof of Claim. There are at most $3k + 3g - 6$ edges in $\tilde{H}$, and each edge corresponds in G to a walk of length at most $2r + 1$ connecting vertices of S. There are at most k vertices in S, and thus $|R_0| \le k + 2r(3k + 3g - 6)$. □

Let $C_1, C_2, \ldots, C_\ell$ be the connected components of $G \backslash R_0$. We use the following properties of these connected components.

Claim 2.

$$\left|\{i \,:\, |N_G(C_i)| \ge 3\}\right| \le 2|R_0| + 2g - 4, \tag{32}$$

$$\sum_{\{i \,:\, |N_G(C_i)| \ge 3\}} |N_G(C_i)| \le 6|R_0| + 6g - 12. \tag{33}$$

Proof of Claim. We construct a new graph G' from G by deleting all components C_i such that $|N(C_i)| < 3$, contracting each component C_i with $|N(C_i)| \ge 3$ to a single vertex, removing all edges between vertices in R_0, and removing double edges and self loops. Thus, G' is a bipartite simple graph, and therefore every face of G' is incident with at least four edges.

Let $c = \left|\{i \,:\, |N_G(C_i)| \ge 3\}\right|$ and $r = |R_0|$. Additionally, let m be the number of edges and f be the number of faces in G'. Since every face of G' is incident with at least four edges, we have that $m \ge 2f$. This fact, together with Euler's formula (31), yields that

$$c + r - m + \frac{m}{2} \ge 2 - g.$$

Hence,

$$m \le 2(c + r) - 4 + 2g. \tag{34}$$

On the other hand, since every vertex of G' corresponding to C_i is incident with at least three edges, we have that $3c \le m$. Hence,

$$3c \le 2(c + r) - 4 + 2g, \tag{35}$$

and thus (32) follows.

Since $\sum_{\{i \,:\, |N_G(C_i)| \ge 3\}} |N_G(C_i)| \le m$, (33) follows from (34) and (35). □

Claim 3. *For each connected component C_i of $G \backslash R_0$, the treewidth of $G[N[C_i]]$ is at most $f_1(4r + 2, g)$.*

Proof of Claim. By construction of R_0, the component C_i is a subset of X_f for some face f of $\tilde{H}$. The face f is incident to at most three vertices, say x, y, and z. In the graph $\tilde{G}$, the neighborhood of X_f is a subset of $\{x, y, z\}$. Hence, in the graph G, the set $N_G(X_f)$ is a subset of vertices that were contracted to x, y, or z. Thus, also for C_i, it holds that $N_G(C_i)$ is a subset of the vertices that were contracted to x, y, or z.

For every vertex u in C_i, there is a path on at most r vertices starting in u and ending in S. This path must contain a vertex $u' \in N_G(C_i)$. The distance from u' to $\{x, y, z\}$ is at most r. Therefore, the distance from each vertex in C_i to $\{x, y, z\}$ is at most $2r$. Since the distance from x to y and to z is at most $2r + 1$, we have that $N[C_i]$ is covered by a ball of radius $4r + 2$ centered at x. Then, by Proposition 6.3, the treewidth of $G[N[C_i]]$ is at most $f_1(4r + 2, g)$. □

For each $i \le \ell$, define $G_i = G[N(C_i)]$. By Claim 3, we have that the treewidth of G_i is at most $t = f_1(4r + 2, g)$. Next, we claim the following.

Claim 4. *For every i, there exists a set $Y_i \subseteq V(G_i)$ such that*

—$N_G(C_i) \subseteq Y_i$,
—$|Y_i| \leq 2|N_G(C_i)|(t+1)$, *and*
—*every connected component of* $G_i \backslash Y_i$ *has at most* $2(t+1)$ *neighbors in* Y_i.

PROOF OF CLAIM. The proof of this claim is almost identical to the proof of Lemma 5.4. Here the role of the set Z is given to $N_G(C_i)$. We compute a nice tree decomposition of G_i and mark all uppermost forget nodes of the decomposition forgetting vertices of $N(C_i)$. We keep marking each lowest common ancestor of marked nodes as long as possible. The vertices contained in all marked bags form the set Y_i. □

We use Claim 4 to find sets Y_i for every G_i and define the set

$$R = R_0 \cup \bigcup_{\{i\,:\,|N(C_i)|\geq 3\}} Y_i.$$

We partition the remaining set of vertices $V(G)\backslash R$ into sets $Q_1, Q_2, \dots, Q_q$, where every Q_i is the union of connected components of $G\backslash R$ with the same neighborhood in R. We claim that $\mathcal{P} = (R, \{Q_i\}_{1\leq i\leq q})$ is the desired $(\alpha k, \beta)$-protrusion decomposition of G.

First, we have the following bound on $|R|$.

$$|R| \leq |R_0| + \sum_{\{i\,:\,|N(C_i)|\geq 3\}} |Y_i| \leq |R_0| + 2(t+1) \sum_{\{i\,:\,|N(C_i)|\geq 3\}} |N(C_i)| = O(k)$$

Here the last bound follows from (33) together with the bound of Claim 1 that $|R_0| = O(k)$.

There are at most $|R|$ sets Q_i such that $|N(Q_i)| = 1$. By Euler's formula, there are at most $3|R| + 6g - 6$ sets Q_i with exactly two neighbors in R. Again, by Euler's formula, exactly as in (32), the number of sets Q_i with at least three neighbors in R is at most $2|R| + 2g - 4$. Hence, $q \leq 6|R| + 7g = O(k)$.

By Claim 4, we have that $|N(Q_i)| \leq 2(t+1)$ for every i. Furthermore, for every i, we have that each connected component of $G[Q_i]$ is in fact C_j for some j, and hence by Claim 3, $G[Q_i]$ has treewidth at most t. Thus, $G[N[Q_i]]$ is a protrusion with treewidth at most $3t+2$ and boundary size at most $2(t+1)$. This completes the proof of Lemma 6.2. □

6.3. Decomposition Lemma for Quasi-Coverable Problems

In this section, we prove the following decomposition lemma.

LEMMA 6.4. *Every r-quasi-coverable problem has the weak protrusion decomposition property* $\mathbf{B}^*$.

Given the definition of r-quasi-coverability, Lemma 6.4 is a direct consequence of the following graph-theoretic result.

LEMMA 6.5. *There exist functions* ζ_1 *and* ζ_2 *such that the following holds: let* r, g, p, *and* k *be nonnegative integers, and let* $G = (V, E)$ *be a graph embedded in a surface* Φ *of Euler genus* g *such that*

—*G contains a set S of vertices, where* $|S| \leq k$ *and* $\mathbf{tw}(G \backslash \mathbf{R}^r_G(S)) \leq r$, *and*
—*for every* $\lambda \leq \zeta_1(r, g)$, *$G$ has no λ-protrusion of size at least p.*

Then G has a (ck, c)*-protrusion decomposition, where* $c = \zeta_2(g, r, p)$.

Indeed, we set $g = r$ in Lemma 6.5. Then combinatorial property $\mathbf{B}^*$ holds for $c' = \zeta_1(r, g)$ and $g(x) = \zeta_2(r, r, x)$.

The rest of this section is devoted to the proof of Lemma 6.5. Let us outline first the main ideas of the proof. Let S be a subset of V of size k such that removal of balls of radius r (in radial distance) around vertices of S from G results in a graph of treewidth at most r. We enlarge the set S by adding at most k new vertices, and we want the new set S' to satisfy the following property:

—Balls of radius μ (in radial distance) around vertices of S' cover all vertices of G, where μ is a constant depending on r, p and g.

If we succeed to find such a set S', then we can use Lemma 6.2 to obtain a (ck, c)-protrusion decomposition of G for some constant c. To find the required set S', we show how to construct a superset S' of S of size at most $2k$ such that for every vertex v at distance $\geq 2\mu$ from S' in the graph $G\backslash \mathbf{B}_G^{\mu}(v)$, there are at most two connected components containing vertices of S'. This construction is given in Lemma 6.6. To prove that S' is the required set, we have to prove that every vertex of G is at radial distance μ from some vertex of S'. The proof of this fact is based on the proof that in graphs embedded in a surface of bounded genus, two connected sets embedded at a large radial distance from each other and nonseparable by "small" separators form an obstruction for having "small" treewidth (Lemma 6.11). Because the treewidth of the graph $G\backslash \mathbf{R}_G^r(S')$ is at most r, we obtain that if there is a vertex v at distance $> \mu$ from S', then a ball of radius p around this vertex should be separated from the remaining graph by a small separator. This yields that G has a protrusion containing a ball of radius p around v and thus of size at least p. But by the assumption of the lemma, there is no such a protrusion. Thus, every vertex v is within distance $\leq \mu$ from S'.

We proceed with the proof of Lemma 6.5.

Constructing S' from S. Let G be a graph, H be a subgraph of G, and $S \subseteq V(G)$. An *S-component* of H is a connected component of H containing some of the vertices of S.

LEMMA 6.6. *Let μ be a positive integer, $G = (V, E)$ be a connected graph, and S be a subset of V. Then there is a set $S' \supseteq S$ such that*

—$|S'| \leq \max\{2|S| - 2, 1\}$, *and*
—*for every $v \in V\backslash \mathbf{B}_G^{2\mu}(S')$, graph $G\backslash \mathbf{B}_G^{\mu}(v)$ has at most two S'-components.*

PROOF. We use induction on $|S|$. As the lemma is obvious when $|S| \leq 2$, we assume that $|S| = k > 2$ and that the lemma holds for all sets S of smaller sizes. Suppose that G contains a vertex u such that $\mathbf{dist}_G(u, S) \geq 2\mu + 1$ and $G^- = G\backslash \mathbf{B}_G^{\mu}(u)$ has at least three S-components. (If there is no such vertex u, we are done.) We denote these components by $C_1, \dots, C_h$, $h \geq 3$, and we denote by $C_{h+1}, \dots, C_\ell$ the connected components of G^- not containing vertices from S. For $i \in \{1, \dots, \ell\}$, we define

$$S_i = (S \cap V(C_i)) \cup \{u\}$$

and

$$G_i = G\big[\mathbf{B}_G^{\mu}(u) \cup V(C_i)\big].$$

Notice that each S_i is a vertex subset of the connected graph G_i and that $1 \leq |S_i| \leq |S| - 1 = k - 1$. This means that the induction hypothesis holds for G_i and S_i. Thus, for every $i \in \{1, \dots, \ell\}$, there is a set $S_i' \supseteq S_i$ such that $|S_i'| \leq \max\{2|S_i| - 2, 1\}$, and

$$\forall v \in V(G_i)\backslash \mathbf{B}_{G_i}^{2\mu}(S_i'),\ \text{graph } G_i\backslash \mathbf{B}_{G_i}^{\mu}(v) \text{ has at most two } S_i'\text{-components.} \tag{36}$$

We now set $S' = \bigcup_{1\le i\le \ell} S'_i$. Clearly, $S' \supseteq S$. Notice also that u appears in every S'_i, whereas every other vertex of S' appears in exactly one of $S'_1, \ldots, S'_h$. Therefore,

$$\begin{aligned} |S'| &= \left(\sum_{i=1}^{h} |S'_i|\right) - (h-1) \\ &\le 2\cdot\left(\sum_{i=1}^{h} |S_i|\right) - 2h - h + 1 \\ &= 2\cdot\left(\sum_{i=1}^{h} |S_i\setminus\{u\}|\right) + 2h - 3h + 1 \\ &= 2|S| - h + 1 \le 2k - 2. \end{aligned}$$

(For the last inequality, we use the assumption that $h \ge 3$.)

We claim that for every $v \in V\setminus \mathbf{B}_G^{2\mu}(S')$, the graph $G\setminus \mathbf{B}_G^{\mu}(v)$ has at most two S'-components. Without loss of generality, let us assume that v belongs to the connected component C_1 of $G^- = G\setminus \mathbf{B}_G^{\mu}(u)$. By (36), in the corresponding graph G_1, the subgraph $G_1\setminus \mathbf{B}_{G_1}^{\mu}(v)$ has at most two S'_1-components, where $S'_1 = V(G_1) \cap S'$, and one of these components contains u. The distance from u to v is at least $2\mu + 1$, and hence the whole ball $\mathbf{B}_{G_v}^{\mu}(v)$ is contained in C_1. Therefore, every vertex $w \in S'\setminus S_1$ is connected with u in G by a path avoiding $\mathbf{B}_G^{\mu}(v)$. Thus, $G\setminus \mathbf{B}_G^{\mu}(v)$ has at most two S'-components. □

Treewidth obstructions. The main result of this section is Lemma 6.11. It can be seen as an extension of the following result: if a graph of bounded genus has two vertices that are far apart (in the radial distance) and cannot be separated by a small separator, then the treewidth of the graph is large [Mohar and Thomassen 2001]. However, for the purposes of the proof, we need an extension of this result for two "radially" connected and nonseparable vertex sets.

To prove Lemma 6.11, we need several combinatorial results. We use the following proposition from Juvan et al. [1996] (see also Proposition 4.2.7 in Mohar and Thomassen [2001].

PROPOSITION 6.7. *Let G be a graph embedded in a surface Φ of Euler genus g, $x, y \in V(G)$, and let $\mathcal{P}$ be a collection of pairwise internally vertex-disjoint paths from x to y such that no two of them are homotopic. Then $|\mathcal{P}| \le h(g)$, where*

$$h(g) = \begin{cases} g+1 & \textit{if } g \le 1 \\ 3g-2 & \textit{if } g \ge 2. \end{cases}$$

Let $G = (V, E)$ a graph, and let X, Y, and Z be pairwise disjoint subsets of V. We say that Y *separates* X and Z if X and Z are in different connected components of $G\setminus Y$. We say that Y is a *minimal* (X, Z)*-separator* if no subset of Y separates X and Z. For $S \subseteq V$, we say that S is *connected in* G if $G[S]$ is a connected graph.

The following properties of minimal separators of connected vertex sets in triangulated graphs are important for obtaining treewidth obstructions.

LEMMA 6.8. *Let G be a triangulated graph embedded in a surface Φ with Euler genus g, and let S be a minimal separator for connected vertex subsets X_1 and X_2 of G. Then S has at most $h(g)$ connected components.*

PROOF. Let $C_1, C_2, \ldots, C_r$ be the connected components of $G\setminus S$. Without loss of generality, we assume that C_1 contains X_1 and C_2 contains X_2. For each component C_i, we select a vertex $x_i \in C_i$, $i \in \{1, \ldots, r\}$. We call the vertices in S *separation vertices* and the vertices $\{x_1, x_2, \ldots, x_r\}$ *satellite* vertices. From G, we construct a graph H by exhaustively contracting or removing edges according to the following rules:

—We contract all edges except the edges with one endpoint being a satellite vertex and the other endpoint a separation vertex.
—We delete loops that are not boundaries of faces, and as long as possible, we delete one of the multiple edges incident with trivial faces (i.e., faces incident with two edges).

Notice that every connected component C_i is contracted to a single vertex x_i and every connected component of $G[S]$ is also contracted to a single vertex. In addition, each application of the preceding rules results in a triangulated graph, and thus H is triangulated. Let S' be the vertices of H that resulted in the contracting of $G[S]$. The vertices of S' form a minimal (x_1, x_2)-separator in H, and thus each of x_i, $i \in \{1, 2\}$, is adjacent to all vertices of S'. Hence, there exist $|S'|$ internally vertex-disjoint paths of length two from x_1 to x_2 in H. Because H is triangulated, these (x, y)-paths are pairwise nonhomotopic; otherwise, some edge in $H[S']$ could be further contracted or deleted. Combining this with Proposition 6.7, we deduce that $|S'| \leq h(g)$. The lemma now follows by observing that each connected component of S shrinks to a single vertex of S', and therefore S has $|S'| \leq h(g)$ connected components. □

We say that two vertex subsets X, Y of graph G *touch* if either $X \cap Y \neq \emptyset$ or there exist an edge of G with one endpoint in X and the other in Y. A *bramble* of G is a collection $\mathcal{B}$ of mutually touching connected subsets of $V(G)$. The *order* of a bramble $\mathcal{B}$ is the minimum size of a set S that intersects all of its elements. The *bramble* number of G is the maximum order that a bramble of G may have.

The following min-max characterization of treewidth was proved in Seymour and Thomas [1993].

Proposition 6.9. *The treewidth of a graph is one less than its bramble number.*

We define functions f_1, f_2 such that $f_1(x, y) = (x+1)y$ and $f_2(x, y) = x(\binom{(x+1)y}{x+1}) + 1$. The following lemma can be seen as a generalization of (3.2) in Seymour and Thomas.

Lemma 6.10. *Let q, t be nonnegative integers, and let $r_1 = f_1(t, q)$ and $r_2 = f_2(t, q)$. Let G be a graph, and let $\mathcal{X} = \{X_1, \ldots, X_{r_1}\}$ be a collection of mutually disjoint connected vertex sets of G. In addition, let $\mathcal{Y} = \{Y_1, \ldots, Y_{r_2}\}$ be a collection of mutually disjoint vertex sets of G, each with at most q connected components and such that for every $i \in \{1, \ldots, r_1\}$ and $j \in \{1, \ldots, r_2\}$, $X_i \cap Y_j \neq \emptyset$. Then $\mathbf{tw}(G) \geq t$.*

Proof. For every set $Y_j, j \in \{1, \ldots, r_2\}$, we select its connected component Y'_j intersecting the largest number of sets from $\mathcal{X}$. Because every Y_j has at most q connected components, set Y'_j intersects at least $t + 1 = r_1/q$ sets from $\mathcal{X}$.

Now let R be the intersection graph of sets $\mathcal{X}$ and $\mathcal{Y}' = \{Y'_1, \ldots, Y'_{r_2}\}$. Then R is a bipartite graph with bipartition $(\mathcal{X}, \mathcal{Y}')$, and every vertex from $\mathcal{Y}'$ has degree $\geq t + 1$ in R. We remove edges from R such that in the resulting graph, all vertices of $\mathcal{Y}'$ have degree exactly $t + 1$. In the new graph, the vertices from $\mathcal{Y}'$ have at most

$$\binom{|\mathcal{X}|}{t+1} = \binom{(t+1)q}{t+1}$$

distinct neighborhoods in $\mathcal{X}$. Since

$$|\mathcal{Y}'| = |\mathcal{Y}| = t\binom{(t+1)q}{t+1} + 1,$$

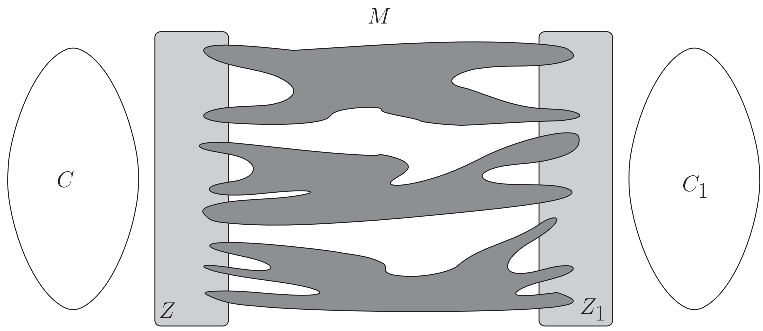

Fig. 1. Visualization of the statement of Lemma 6.11.

we deduce that there should be at least $t+1$ vertices of $\mathcal{Y}'$ with the same neighborhood in $\mathcal{X}$. Let $I_{\mathcal{Y}}$ be the indices of these vertices in $\mathcal{Y}$, and let $I_{\mathcal{X}}$ be the indices of their neighbors in $\mathcal{X}$.

It follows that for every $(i, j) \in I_{\mathcal{X}} \times I_{\mathcal{Y}}$, $X_i \cap Y'_j \neq \emptyset$, and, as both X_i and Y'_j are connected, $X_i \cup Y'_j$ is also a connected set. Moreover, because $|I_{\mathcal{X}}| = |I_{\mathcal{Y}}| = t+1$, it follows that for every set S of t vertices in G, there are $i \in I_{\mathcal{X}}$ and $j \in I_{\mathcal{Y}}$ such that $S \cap (X_i \cup Y'_j) = \emptyset$. We can now conclude that the collection $\{X_i \cup Y'_j \mid (i, j) \in I_{\mathcal{X}} \times I_{\mathcal{Y}}\}$ is a bramble in G of order $t+1$. Therefore, the bramble number of G is at least $t+1$, and the lemma follows from Proposition 6.9. □

Let G be a graph embedded in some surface Φ. We define the *radial completion* of G as the graph obtained from drawing of G in Φ together with its radial graph R_G. We denote the radial completion of G by W_G. Let us remark that W_G is triangulated and that R_G is a spanning subgraph of W_G. Notice that every two adjacent vertices in W_G have some common neighbor in R_G. This implies the following observation.

OBSERVATION 5. *Let G be a graph embedded in some surface Φ. Then for every pair $x, y \in V(R_G)$, it holds that $\mathbf{dist}_{W_G}(x, y) \leq \mathbf{dist}_{R_G}(x, y) \leq 2 \cdot \mathbf{dist}_{W_G}(x, y)$.*

Loosely speaking, the following lemma says that in a graph of small treewidth that is embedded on a surface of fixed genus, every two connected sets will be either radially close or separated by a small set. Let h be the function from Lemma 6.8, and let f_1, f_2 be the functions defined before Lemma 6.10.

LEMMA 6.11. *Let G be a graph embedded in a surface Φ of Euler genus g, t be a positive integer, and C, Z, Z_1, C_1 be disjoint subsets of $V(W_G)$ such that*

—*C and C_1 are connected in W_G,*
—*Z separates C from $Z_1 \cup C_1$ and Z_1 separates $C \cup Z$ from C_1 in W_G,*
—*$\mathbf{dist}_{W_G}(Z, Z_1) \geq 3 \cdot f_2(t+1, h(g)) + 3$, and*
—*G contains $f_1(t+1, h(g))$ internally vertex-disjoint paths from $C \cap V(G)$ to $C_1 \cap V(G)$.*

Then $\mathbf{tw}(G[V(M) \cap V(G)]) > t$, where M is the union of all connected components of $W_G \setminus (Z \cup Z_1)$ that have at least one neighbor in Z and at least one neighbor in Z_1 (Figure 1).

PROOF. We set $\mu = f_1(t+1, h(g))$ and $\lambda = f_2(t+1, h(g))$. Let $P_1, \ldots, P_\mu$ be internally vertex-disjoint paths in G from $C \cap V(G)$ to $C_1 \cap V(G)$. Each of these paths P_i contains at least one subpath with one endpoint in Z and the other in Z_1, and with all internal vertices in M. We denote by $P'_1, \ldots, P'_{\mu'}$ the set of such subpaths. Then $\mu' \geq \mu$.

For $j \in \{1, \ldots, 3\lambda + 2\}$, let A_j be the set of all vertices of W_G that are within distance exactly j from Z and belonging to M. Notice that each A_j is a (Z, Z_1)-separator and thus also a (C, C_1)-separator of W_G. Clearly, each A_j contains as a subset a minimal (C, C_1)-separator Y_j of W_G. As each Y_j is also a (Z, Z_1)-separator, it should contain at least one internal vertex of every path in $P'_1, \ldots, P_{\mu'}$. Moreover, by its definition, A_j should be a subset of M.

As W_G is triangulated, by Lemma 6.8, each $W_G[Y_j]$ contains at most $h(g)$ connected components. Recall that by the definition of W_G, for each vertex $x \in V(W_G)\backslash V(G)$, the graph induced by its neighborhood is a connected subgraph of G. Using this fact, we obtain that the subgraph of G induced by $Y_j^+ = \mathbf{B}^1_{W_G}(Y_j) \cap V(G)$ also has at most $h(g)$ connected components for $j \in \{2, \ldots, 3\lambda + 1\}$.

Let $I = \{1, \ldots, \lambda\}$, and notice that for any two distinct $h, l \in I$, sets Y^+_{3h} and Y^+_{3l} are disjoint. For $j \in \{1, \ldots, \mu'\}$, we define P''_j as the path obtained from P'_j after removing its endpoints. Observe now that $P''_1, \ldots, P''_{\mu'}$ are connected vertex-disjoint subgraphs of $G[V(M) \cap V(G)]$, and each of these graphs intersect all sets Y^+_{3j}. Applying Lemma 6.10 for μ graphs from $\{P''_1, \ldots, P''_{\mu'}\}$ and λ graphs from $\{Y^+_{3j} \mid j \in I\}$, we deduce that $\mathbf{tw}(G[V(M) \cap V(G)]) \geq t + 1 > t$, and the lemma follows. □

Final step. To conclude the proof of the main result of this section, we need the last lemma. The following lemma essentially says that if (G, k) is a YES-instance of a quasi-coverable problem Π where G has no big protrusion, then G has an r-dominating set of size $O(k)$ for some r that depends only on Π and g. Therefore ,(G, k) can be treated as a YES-instance of a coverable problem.

We define function $f_3(x, y) = 2 \cdot f_1(x + 1, h(y + 1))$, where h is the function of Lemma 6.8 and f_1 is the function defined before Lemma 6.10.

LEMMA 6.12. *Let $G = (V, E)$ be a graph embedded in a surface Φ of Euler genus g, and let p, t, and r be nonnegative integers such that*

—*there exists a set $S \subseteq V$ such that $\mathbf{tw}(G\backslash\mathbf{R}^r_G(S)) \leq t$, and*
—*for $\lambda \leq f_3(t, g)$, all λ-protrusions of G are of size less than p.*

Then there exist a set $S' \subseteq V$ and a constant μ (depending on p, g, and r only) such that

—$|S'| \leq 2|S|$ *and*
—$\mathbf{R}^\mu_G(S') = V$.

PROOF. To prove the lemma, we prove a slightly different statement: under the assumptions of the lemma, there is a set $S' \subseteq V(W_G)$ such that $|S'| \leq 2|S|$ and $\mathbf{B}^\mu_{W_G}(S') = V(W_G)$. Then the statement of the lemma can be deduced from this alternative statement by constructing set S'_{new} as follows: first set $S'_{\text{new}} \leftarrow S'$, and then replace each vertex in S' that does not belong to $V(G)$ with one of its neighbors from $V(G)$. It remains to observe that $\mathbf{R}^{\mu+1}_G(S'_{\text{new}}) \supseteq \mathbf{B}^\mu_{W_G}(S')$.

We put $\mu = 2p + 2r + 2 + 2\mu'$, where $\mu' = 3 \cdot f_2(t + 1, h(g)) + 3$, and proceed with the proof of the preceding alternative statement. We first apply Lemma 6.6 for W_G and S to obtain a set $S' \supseteq S$ of vertices, where $|S'| \leq 2|S|$ and such that for every $v \in W_G\backslash\mathbf{B}^{2\mu}_{W_G}(S')$, graph $W_G\backslash\mathbf{B}^\mu_{W_G}(v)$ has at most two S'-components. If $\mathbf{B}^{2\mu}_{W_G}(S') = V(W_G)$, then we are done. Otherwise, let $v \in W_G\backslash\mathbf{B}^{2\mu}_{W_G}(S')$. Let C_1, C_2 be S'-components of $W_G\backslash\mathbf{B}^\mu_{W_G}(v)$ (one of these components can be an empty set), and let $S_i = C_i \cap S'$, $i \in \{1, 2\}$. We also define subgraphs of W_G as follows: $W_1 = W_G\backslash C_2$ and $W_2 = W_G\backslash C_1$.

We claim that at least one of the sets C_i, $i \in \{1, 2\}$, cannot be separated in W_i from $C = B^{2p}_{W_G}(v)$ by a separator of size at most $\lambda/2$. Indeed, if it was the case, then in W_G, C is separable from $C_1 \cup C_2$ and thus from $\mathbf{B}^{2r}_{W_G}(S') \subseteq C_1 \cup C_2$ by a separator of size at most λ. By Observation 5, this means that in G, vertices $R^p_G(v)$ can be separated from $\mathbf{R}^r_G(S')$ by a separator of size at most λ. Because $\mathbf{tw}(G\backslash\mathbf{R}^r_G(S')) \leq t$, this yields that there is a λ-protrusion in G containing $\mathbf{R}^p_G(v)$. But $|\mathbf{R}^p_G(v)| \geq p$, and thus the size of this protrusion is at least p in G, which contradicts to the assumption of the lemma.

Without loss of generality, let us assume that C_1 is a S'-component of $W_G\backslash\mathbf{B}^{\mu}_{W_G}(v)$ that cannot be separated in W_1 from C by a separator of size $\lambda/2$. By Menger's theorem, in graph W_1 there are $\lambda/2$ internally vertex-disjoint paths from C to C_1. We define Z as the set of vertices at distance exactly $2p+1$ from v in W_1, and Z_1 as $N_{W_1}(C_1)$. Then Z separates C from $Z_1 \cup C_1$ and Z_1 separates C_1 from $Z \cup C$. The distance in W_1 between Z and Z_1 is at least μ'. Let M be the union of connected components of $W_1\backslash(Z_1 \cup Z_2)$ having at least one neighbor in Z and Z_1. By Lemma 6.11, the treewidth of the subgraph G_M of G induced by $M \cap V(G)$ is more than t. On the other hand, every vertex of M is at distance more than $r+1$ in W_G, and thus at radial distance at least $r+1$ in G, from each vertex of S', and thus of S. Hence, $\mathbf{tw}(G_M) \leq \mathbf{tw}(G\backslash\mathbf{R}^r_G(S))$, which is at most t by the assumption of the lemma. This contradiction concludes the proof of the lemma. □

Proof of Lemma 6.5. By applying Lemma 6.12 for $r = t$ and $\zeta_1 = f_3$, we have that G contains a set of vertices S', where $|S'| \leq 2k$ such that $\mathbf{R}^{\mu}_G(S') = V(G)$ and μ is the constant of Lemma 6.12. But then by Lemma 6.2, G has a (ck, c)-protrusion decomposition for some c depending on $g, r,$ and p, as required. □

7. CRITERIA FOR PROVING FII

To apply Theorem 1.3, to prove that a specific parameterized problem on graphs admits a linear kernel, we have to show that it has FII. This property is not always easy to prove directly. In this section, we give some general criteria for establishing FII. These tools are used in Section 8. Early results that establish that problems have FII were obtained by Bodlaender and de Fluiter [1996], Bodlaender and van Antwerpen-de Fluiter [2001], and de Fluiter [1997]; another criterion for FII was given in Section 11.2 of van Rooij [2011].

7.1. Strong Monotonicity

We first give a sufficient condition that implies that a large class of p-MIN/MAX-CMSO[ψ] problems has FII. We prove it here for vertex versions of p-MIN/MAX-CMSO[ψ] problems. By $\mathcal{U}_I$, we denote the set of all boundaried structures of type (graph, vertex set), whose boundaried graph has label set I.

Let Π be a p-MIN-CMSO[ψ] problem definable by some sentence ψ. We say that a boundaried structure (G', S') whose boundaried graph has label set I is *ψ-feasible for some boundaried graph G with label set I* if there exist some $S \subseteq V(G)$ such that $(G \oplus G', S \cup S') \models \psi$. For a boundaried graph G with label set I, we define the function $\zeta_G : \mathcal{U}_I \to \mathbb{Z}^+ \cup \{\infty\}$ as follows. For a structure $\alpha = (G', S') \in \mathcal{U}_I$, we set

$$\zeta_G(\alpha) = \begin{cases} \min\{|S| \mid S \subseteq V(G) \wedge (G \oplus G', S \cup S') \models \psi\} & \text{if } \alpha \text{ is } \psi\text{-feasible for G} \\ \infty & \text{otherwise.} \end{cases} \tag{37}$$

Similarly, for Π p-MAX-CMSO[ψ] problems, we define

$$\zeta_G(\alpha) = \begin{cases} \max\{|S| \mid S \subseteq V(G) \wedge (G \oplus G', S \cup S') \models \psi\} & \text{if } \alpha \text{ is } \psi\text{-feasible for G} \\ -\infty & \text{otherwise.} \end{cases}$$

Definition 7.1. A p-MIN-CMSO[ψ] problem Π is *strongly monotone* if there exists a function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that the following condition is satisfied. For every boundaried graph G with label set I, there exists a subset $W \subseteq V(G)$ such that for every $(G', S') \in \mathcal{U}_I$ such that $\zeta_G(G', S')$ is finite, it holds that $(G \oplus G', W \cup S') \models \psi$ and $|W| \leq \zeta_G(G', S') + f(|I|)$.

For completeness, in the following we give the maximization counterpart of Definition 7.1.

Definition 7.2. A p-MAX-CMSO[ψ] problem Π is *strongly monotone* if there exists a function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that the following condition is satisfied. For every boundaried graph G with label set I, there exists a subset $W \subseteq V(G)$ such that for every $(G', S') \in \mathcal{U}_I$ such that $\zeta_G(G', S')$ is finite, it holds that $(G \oplus G', W \cup S') \models \psi$ and $|W| \geq \zeta_G(G', S') - f(|I|)$.

7.2. FII for *p*-MIN/MAX-CMSO[ψ] Problems

LEMMA 7.3. *Every strongly monotone p-*MIN*-*CMSO[ψ] *and every strongly monotone p-*MAX*-*CMSO[ψ] *problem has FII.*

PROOF. We prove the lemma for a p-MIN-CMSO[ψ] problem; the proof for a p-MAX-CMSO[ψ] problem is similar. Let Π be a strongly monotone p-MIN-CMSO[ψ] problem, and let $I \subseteq \mathbb{Z}^+$. Let **MinRep**(ψ, I) be a set containing a representative (a boundaried structure of arity 2) for each equivalence class of $\equiv_{\sigma_\psi}$ with the minimum number of vertices in the graph of a structure. For brevity, we denote **MinRep**(ψ, I) by $\mathcal{S}$. From Lemma 3.2, we know that $|\mathcal{S}|$ is bounded by some function of $|\psi|$ and $|I|$.

Consider a boundaried graph G with label set I, and define $\zeta_G^{\mathcal{S}} : \mathcal{S} \to \mathbb{Z}^+ \cup \{\infty\}$ to be the function ζ_G with domain restricted to $\mathcal{S}$. Let $L_G^{\mathcal{S}} = \{\zeta_G^{\mathcal{S}}(\alpha) \mid \alpha \in \mathcal{S}\} \backslash \{\infty\}$. We first argue that if f is the function in the definition of the strong monotonicity of Π (i.e., Definition 7.1) and $L_G^{\mathcal{S}} \neq \emptyset$, then

$$\max L_G^{\mathcal{S}} - \min L_G^{\mathcal{S}} \leq f(|I|). \tag{38}$$

Since Π is strongly monotone, there exists $W \subseteq V(G)$ such that for every $(G', S') \in \mathcal{U}_I$ where $\zeta_G(G', S') \neq \infty$, it holds that

$$(G \oplus G', W \cup S') \models \psi \text{ and} \tag{39}$$

$$|W| \leq \zeta_G(G', S') + f(|I|). \tag{40}$$

Let $\alpha = (G', S') \in \mathcal{S}$ such that $\zeta_G^{\mathcal{S}}(\alpha) \neq \infty$. Then (39) implies that $\zeta_G^{\mathcal{S}}(\alpha) \leq |W|$. This, together with (40), yields that $|W| - f(|I|) \leq \zeta_G^{\mathcal{S}}(\alpha) \leq |W|$ and (38) holds. Hence, the minimum and the maximum finite values of $\zeta_G^{\mathcal{S}}$ can differ by at most $f(|I|)$.

We now assign for each boundaried graph G with label set I a *signature* $\chi_G : \mathcal{S} \to \{0, \ldots, f(|I|), \infty\}$ in a way that for each $\alpha \in \mathcal{S}$,

$$\chi_G(\alpha) = \zeta_G^{\mathcal{S}}(\alpha) - \min L_G^{\mathcal{S}}. \tag{41}$$

In (41), we make the agreement that infinite values remain infinite after subtracting an integer. Notice that it is possible that in (41), $\min L_G^{\mathcal{S}}$ may not exist, and this happens in the extreme case where $L_G^{\mathcal{S}} = \emptyset$. In such a case, we set $\chi_G(\alpha) = \infty$ for all $\alpha \in \mathcal{S}$.

We say that $G_1 \sim G_2$ if and only if $\chi_{G_1} = \chi_{G_2}$ and observe that $\sim$ is an equivalence relation. Observe that the number of different signatures of boundaried graphs with label set I is bounded by some function of $|\psi|$ and $|I|$. Therefore, the same holds for the number of equivalent classes of $\sim$. To prove that $\equiv_\Pi$ has FII, it is enough to prove that $\sim$ is a refinement of $\equiv_\Pi$, which means that if $G_1 \sim G_2$, then $G_1 \equiv_\Pi G_2$. For this, we claim that if $G_1 \sim G_2$, then there exists some constant $c \in \mathbb{Z}$ (depending on G_1 and G_2) such that

$$\forall (F,k) \in \mathcal{F} \times \mathbb{Z} (G_1 \oplus F, k) \in \Pi \Leftrightarrow (G_2 \oplus F, k+c) \in \Pi. \quad (42)$$

To prove the preceding statement, we first determine the constant c. As $G_1 \sim G_2$, we have that $\chi_{G_1} = \chi_{G_2}$. In the extreme case where $\chi_{G_1}(\alpha) = \chi_{G_2}(\alpha) = \infty$ for all $\alpha \in \mathcal{S}$, (42) holds trivially for $c = 0$, as $\forall (F,k) \in \mathcal{F} \times \mathbb{Z}^+$ both sides of the equivalence are false (for completeness, recall that according to the way we defined parameterized problems, $\forall (F,k) \in \mathcal{F} \times \mathbb{Z}^-$, both sides of the equivalence in (42) have the same value). From now onward, we assume that both $\min L^{\mathcal{S}}_{G_1}$ and $\min L^{\mathcal{S}}_{G_2}$ exist. Therefore, from (41), for each $\alpha \in \mathcal{S}$, $\zeta^{\mathcal{S}}_{G_2}(\alpha) = \zeta^{\mathcal{S}}_{G_1}(\alpha) - \min L^{\mathcal{S}}_{G_1} + \min L^{\mathcal{S}}_{G_2}$. We set $c = \min L^{\mathcal{S}}_{G_2} - \min L^{\mathcal{S}}_{G_1}$ and conclude that

$$\forall \alpha \in \mathcal{S}\; \zeta^{\mathcal{S}}_{G_2}(\alpha) = \zeta^{\mathcal{S}}_{G_1}(\alpha) + c. \quad (43)$$

Let $(F,k) \in \mathcal{F} \times \mathbb{Z}$, and assume that $(G_1 \oplus F, k) \in \Pi$. This means that there exists a set $S \subseteq V(G_1 \oplus F)$ such that $|S| \leq k$ and

$$(G_1 \oplus F, S) \models \psi. \quad (44)$$

Let $S_F = S \cap V(F)$ and $S_{G_1} = S \backslash S_F$, and observe that

$$|S_{G_1}| + |S_F| \leq k. \quad (45)$$

We rewrite (44) as follows:

$$(G_1, S_{G_1}) \oplus (F, S_F) \models \psi. \quad (46)$$

Let $(F', S'_F) \in \mathcal{S}$ be the representative of (F, S_F). As $(F, S_F) \equiv_{\sigma_\psi} (F', S'_F)$, (46) implies that

$$\begin{aligned} (G_1, S_{G_1}) \oplus (F', S'_F) &\models \psi \\ \iff (G_1 \oplus F', S_{G_1} \cup S'_F) &\models \psi. \end{aligned} \quad (47)$$

From (37), (47) implies that $\zeta_{G_1}(F', S'_F) \leq |S_{G_1}|$. From (43), we get $\zeta^{\mathcal{S}}_{G_2}(F', S'_F) \leq |S_{G_1}| + c$ which, again from (37), means that there exists S_{G_2}, where

$$(G_2 \oplus F', S_{G_2} \cup S'_F) \models \psi \text{ and} \quad (48)$$

$$|S_{G_2}| \leq |S_{G_1}| + c. \quad (49)$$

We rewrite (48) as follows:

$$(G_2, S_{G_2}) \oplus (F', S'_F) \models \psi. \quad (50)$$

As $(F', S'_F) \equiv_{\sigma_\psi} (F, S_F)$, (50) implies that

$$\begin{aligned} (G_2, S_{G_2}) \oplus (F, S_F) &\models \psi \\ \iff (G_2 \oplus F, S_{G_2} \cup S_F) &\models \psi. \end{aligned}$$

Moreover, $|S_{G_2} \cup S_F| \leq |S_{G_2}| + |S_F| \leq^{(49)} |S_{G_1}| + c + |S_F| \leq^{(45)} k + c$. We conclude that $(G_2 \oplus F, k+c) \in \Pi$, and we proved the one direction of (42). The other direction is symmetric. □

Remark 7.4. In Definitions 7.1 and 7.2, we defined the notion of strong monotonicity for p-MIN/MAX-CMSO[ψ] problems where S is a subset of the vertices of the input graph. If instead we ask S to be an edge subset, then an analogue of Lemma 7.3 can be proved in a similar manner.

Let $\mathcal{G}$ be a graph class. We say that $\mathcal{G}$ is CMSO *definable* if there exist a sentence ψ on graphs such that $\mathcal{G} = \{G \mid G \models \psi\}$, and in such a case, we say that ψ defines the class $\mathcal{G}$. Recall that, given a parameterized graph problem Π and a graph class $\mathcal{G}$, we denote by $\Pi \Cap \mathcal{G}$ the problem obtained by removing from Π all instances that encode graphs that do not belong to $\mathcal{G}$.

A necessary tool to adapt our results to problems on special graph classes is the following. The proof follows directly by the definitions.

LEMMA 7.5. *Let Π be a parameterized problem on graphs, and let $\mathcal{G}$ be a CMSO-definable graph class. Then if Π has FII, so does $\Pi \Cap \mathcal{G}$.*

8. IMPLICATIONS OF OUR RESULTS

In this section, we mention a few parameterized problems for which we can obtain either polynomial or linear kernel using Theorems 1.1, 1.2, and 1.3. In the appendix, we provide a full list of the problems amenable to our approach.

8.1. Preliminary Tools

All of our results concern problems defined on graphs of bounded genus. Recall that we denote by $\mathcal{G}_g$ the class of all graphs of Euler genus at most g. In this way, for every parameterized problem Π on graphs, we define the problem $\Pi_g = \Pi \Cap \mathcal{G}_g$, which contains only YES-instances of Π, encoding graphs of Euler genus at most g. We need to distinguish the two variants Π and Π_g. The reason for this is that in many cases, for some fixed value g, Π_g admits a polynomial kernel, whereas the general version Π is not even believed to be fixed parameter tractable. A typical example is PLANAR DOMINATING SET, which admits a vertex kernel of size $67k$, whereas the general DOMINATING SET problem is W[2]-complete [Downey and Fellows 1998].

The following lemma is a direct consequence of the definition of coverability and quasi-coverability.

LEMMA 8.1. *Let Π_1, Π_2 be graph problems whose instances are of the form (G, k). Then if $\Pi_1 \subseteq \Pi_2$ and Π_2 is r-(quasi)-coverable, then so is Π_1.*

The next lemma is useful when we work on graphs of bounded genus.

LEMMA 8.2. *Let Π be a parameterized problem on graphs. If Π has FII, then for every $g \in \mathbb{Z}^+$, Π_g has FII.*

PROOF. Let $\mathcal{O}_g$ be the set containing all minor-minimal elements of the class of graphs with Euler genus more than g. According to the results of Mohar [1999], $\mathcal{O}_g$ is finite for each fixed g. Notice that $\mathcal{G}_g = \{G \mid \forall_{H \in \mathcal{O}_g}\ H \not\preceq G\}$, and as minor checking can be expressed in CMSO, the class $\mathcal{G}_g$ is CMSO definable. Therefore, the lemma follows from Lemma 7.5. □

8.2. Covering Minors

A *minor model* of a graph H in a graph G is a minimal subgraph F of G that contains H as a minor. Notice that $H \preceq G$ if and only if G contains as a subgraph some minor model of H.

Next we give a generic problem that subsumes many problems in itself. Let $\mathcal{H}$ be a finite set of connected graphs containing at least one planar graph.

> p-$\mathcal{H}$-DELETION
> *Input*: A graph G and $k \in \mathbb{Z}^+$
> *Parameter*: k
> *Question*: Is there $S \subseteq V(G)$ such that $|S| \leq k$ and $G \setminus S$ do not contain any of the graphs from $\mathcal{H}$ as a minor?

Lemma 8.3. *If* $\Pi = p$-$\mathcal{H}$-Deletion*, then for every* $g \in \mathbb{Z}^+$, Π_g *is quasi-coverable.*

Proof. Let (G, k) be a YES-instance for Π_g. This means that there exists a set $S \subseteq V(G)$ of cardinality at most k such that none of the graphs in $\mathcal{H}$ is a minor of $G\backslash S$. Let H be a planar graph in $\mathcal{H}$. As $G\backslash S$ excludes H as a minor and H is planar, it follows from Robertson et al. [1994] that $\mathbf{tw}(G\backslash S) \leq c_H$ for some constant that depends only on H. Set $r = \max\{g, c_H\}$, and take an embedding of G in a surface of genus at most g. Observe that $G\backslash \mathbf{R}^r_G(S) \subseteq G\backslash S$; therefore, $\mathbf{tw}(G\backslash \mathbf{R}^r_G(S)) \leq \mathbf{tw}(G\backslash S)$. Thus, Π_g has the r-quasi-coverability property for some r depending on H and g. □

Lemma 8.4. *If* $\Pi = p$-$\mathcal{H}$-Deletion*, then for every* $g \in \mathbb{Z}^+$, Π_g *has FII.*

Proof. Let $\psi = [\forall H \in \mathcal{H}\ H \preceq (G\backslash S)]$. As minor checking is CMSO definable, ψ can be written as a CMSO sentence, and hence Π is a p-min-CMSO[ψ] problem. We now prove that Π has FII. By Lemmata 7.3 and 8.2, it suffices to prove that Π is strongly monotone. Let G be a boundaried graph with label set I and the boundary $\delta(G) = B$. Let S^- be a set of minimum size such that $(G\backslash B)\backslash S^-$ does not contain any of the graphs from $\mathcal{H}$ as a minor, and let $W = S^- \cup B$.

Let $(G', S') \in \mathcal{U}_I$ be a ψ-feasible structure. We first prove that $(G \oplus G', W \cup S') \models \psi$. For this, assume on the contrary that R is a minor model of some H from $\mathcal{H}$ contained in $(G \oplus G')\backslash(W \cup S')$. As H is connected and B is a separator of $G \oplus G'$, R should be either a subgraph of $G\backslash W = (G\backslash B)\backslash S^-$ or a subgraph of $(G'\backslash B)\backslash S'$. The first case contradicts to the choice of S^-. In the second case, R would be a subgraph of $(G'\backslash B)\backslash S'$, which contradicts the feasibility of (G', S').

We next prove that $|W| \leq \zeta_G(G', S') + f(|I|)$, where $f(|I|) = |I|$. For $(G', S') \in \mathcal{U}_I$, let $S^* \subseteq V(G)$ be a set of minimum size such that $(G \oplus G')\backslash(S^* \cup S')$ contains no graph from $\mathcal{H}$ as a minor. Thus, $|S^*| = \zeta_G(G', S')$. Notice that $G\backslash B$ does not contain vertices from S'. Therefore, for every $H \in \mathcal{H}$, every minor-model R of H in $G\backslash B$ should be intersected by vertices from S^*; otherwise, R would also be a subgraph of $(G \oplus G')\backslash(S^* \cup S')$, which is a contradiction. By the choice of S^-, we have $|S^-| \leq |S^*|$. We conclude that $|W| = |S^- \cup B| \leq |S^-| + |B| \leq |S^*| + |B| = \zeta_G(G', S') + f(|I|)$. □

p-$\mathcal{H}$-Deletion contains various problems as a special case. Some examples are presented next (all of them are parameterized by solution size k):

- p-Vertex Cover: In this problem, given an input graph G and a $k \in \mathbb{Z}^+$, the objective is to test whether it is possible to remove at most k vertices from G and obtain an edgeless graph. This problem is generated by taking $\mathcal{H} = \{K_2\}$.
- p-Feedback Vertex Set: In this problem, given an input graph G and a $k \in \mathbb{Z}^+$, the objective is to test whether it is possible to remove at most k vertices from G and obtain an acyclic graph. This problem is generated by taking $\mathcal{H} = \{K_3\}$.
- p-Diamond Hitting Set: In this problem, given an input graph G and a $k \in \mathbb{Z}^+$, the objective is to test whether it is possible to remove at most k vertices from G and obtain a graph where no edge is contained in more than one cycle. This problem is generated by taking $\mathcal{H} = \{K_4^-\}$, where K_4^- is the graph obtained from a K_4 after removing an edge.
- p-Almost Outerplanar: In this problem, given an input graph G and a $k \in \mathbb{Z}^+$, the objective is to test whether it is possible to remove at most k vertices from G and obtain an outerplanar graph. This problem is generated by taking $\mathcal{H} = \{K_4, K_{2,3}\}$.
- p-Almost-t-bounded treewidth: In this problem, given an input graph G and a $k \in \mathbb{Z}^+$, the objective is to test whether it is possible to remove at most k vertices from G and obtain a graph of treewidth bounded by some fixed constant t. This problem is generated by taking $\mathcal{H}$ to be the set of minor minimal graphs with treewidth $> t$

(from the results in Robertson et al. [1994], this set always contains a connected planar graph).

—p-ALMOST-t-BOUNDED PATHWIDTH: In this problem, given an input graph G and a $k \in \mathbb{Z}^+$, the objective is to test whether it is possible to remove at most k vertices from G and obtain a graph of pathwidth bounded by some fixed constant t. This problem is generated by taking $\mathcal{H}$ to be the set of minor minimal graphs with pathwidth bigger than t.

8.3. Packing Minors

We consider the following problem, which in a sense is dual to the one examined in Section 8.2. Again, let $\mathcal{H}$ be a finite set of connected graphs containing at least one planar graph.

> p-$\mathcal{H}$-PACKING
> *Input*: A graph G and $k \in \mathbb{Z}^+$
> *Parameter*: k
> *Question*: Does there exist k vertex-disjoint subgraphs $G_1, \dots, G_k$ of G such that each of them contains some graph from $\mathcal{H}$ as a minor?

For proving the quasi-coverability of p-$\mathcal{H}$-PACKING, we need to examine its relation to p-$\mathcal{H}$-DELETION.

LEMMA 8.5. *If* $\Pi =$$p$-$\mathcal{H}$-PACKING, *then for every* $g \in \mathbb{Z}^+$, Π_g *is quasi-coverable.*

PROOF. Given two graphs G and H, we define $\mathbf{cov}_H(G)$ as the minimum size of a set $S \subseteq V(G)$ of vertices such that $G \backslash S$ does not contain any minor model of H.
We also define

$$\mathbf{pack}_H(G) = \max\{k \mid \exists \text{ partition } V_1, \dots, V_k \text{ of } V(G) \text{ such that } \forall_{i \in \{1,\dots,k\}} G[V_i] \text{ is a minor model of } H\}.$$

Let H be a connected planar graph in $\mathcal{H}$. To prove that Π_g is quasi-coverable, we show that $\overline{\Pi}_g = ((\Sigma^* \times \mathbb{Z}^+)\backslash \Pi_g) \Cap \mathcal{G}_g$ has the quasi-coverability property. To do so, we prove that if $(G, k) \in \overline{\Pi}_g$ (i.e., $G \in \mathcal{G}_g$ and has no $\mathcal{H}$-packing into k sets), then (G, ck) is a YES-instance for Π_g^{hd}, where $\Pi^{\text{hd}} =$$p$-$\mathcal{H}$-DELETION for some constant c that depends only on g and H. By Lemma 8.5, p-$\mathcal{H}$-DELETION is r-quasi-coverable, and thus $\overline{\Pi}_g$ would posses a quasi-coverability property.

Suppose that $(G, k) \in \overline{\Pi}_g$. This implies that $\mathbf{pack}_H(G) < k$. According to the Erdős-Pósa type of result of Fomin et al. [2011], for every two graphs H and W, where H is planar and W is any graph, there exists a constant $c_{H,W}$ depending only on H and W such that for every graph G excluding W as a minor, $\mathbf{cov}_H(G) \leq c_{H,W} \cdot \mathbf{pack}_H(G)$. Let W be a graph of Euler genus $g+1$. As the class $\mathcal{G}_g$ is closed under taking of minors, we have that every graph in $\mathcal{G}_g$ excludes W as a minor. Applying the aforementioned result, we have that $\mathbf{cov}_H \leq c_{H,W} \cdot k$, and therefore $(G, c \cdot k)$ is a YES-instance for Π_g^{hd} for some c depending only on H and g, as required. This implies that $\overline{\Pi}_g$ has a quasi-coverability property, and hence Π_g is quasi-coverable. □

Notice that when $\mathcal{H} = \{K_3\}$, p-$\mathcal{H}$-PACKING is the p-CYCLE PACKING problem. Here, given an input graph G and a $k \in \mathbb{Z}^+$, the objective is to check whether G contains k vertex-disjoint cycles. Although the general problem has FII for every choice of $\mathcal{H}$, we present the proof for this special case to clearly explain the machinery that we use for such problems. After the end of the proof of Lemma 8.6, we outline how to extend the proof for the general case.

LEMMA 8.6. *If* $\Pi = p$-CYCLE PACKING, *then for every* $g \in \mathbb{Z}^+$, Π_g *has FII.*

PROOF. By Lemma 8.2, it is sufficient to prove that Π has FII. Let G be a boundaried graph with label set I and with boundary $\delta(G) = B^*$. The proof proceeds in three stages. The first stage defines some characteristic of the problem that depends on the boundary of the input boundaried graph. The second stage uses this characteristic to define an equivalence relation on boundaried graphs that will have finite index. The third stage proves that this equivalence relation is a refinement of $\equiv_\Pi$ and therefore has finitely many equivalence classes as well.

Characteristic. We define set $\mathcal{R}$ as the set of all matchings R (not necessarily maximal) of a complete graph on the vertex set B^*. Let us remark that matching $R \in \mathcal{R}$ is not necessarily a subgraph of G; each graph in $\mathcal{R}$ corresponds to a set of mutually disjoint pairs from B^*. We define $\zeta_G : \mathcal{R} \to \mathbb{Z}^+$ so that for every $R \in \mathcal{R}$, the value $\zeta_G(R)$ is the maximum number of cycles that can be contained in a subgraph J of G such that

—$\Delta(J) \leq 2$, and
—for every edge $\{x, y\}$ of R, J contains an (x, y)-path.

Let us remark that all (x, y)-paths of J are internally vertex disjoint. In case such a graph J does not exist, we set $\zeta_G(R) = -\infty$. Function ζ_G can be seen as a way to encode the tables of a dynamic programming for p-CYCLE PACKING on graphs of treewidth at most $|I|$. The proof that follows can be seen as an alternate way to prove that such a dynamic programming algorithm uses tables whose sizes depend only on $|I|$.

Definition of equivalence. Let x be the maximum number of vertex-disjoint cycles in G. Thus, for every $R \in \mathcal{R}$, we have $\zeta_G(R) \leq x$. We define the *signature* of G as the function $\chi_G : \mathcal{R} \to \{-|I|, \ldots, 0\} \cup \{-\infty\}$ such that

$$\chi_G(R) = \begin{cases} \zeta_G(R) - x & \text{if } x - |I| \leq \zeta_G(R) \leq x \\ -\infty & \text{otherwise.} \end{cases}$$

Notice that the number of different signatures is bounded by some function of $|I|$. Given two boundaried graphs G_1 and G_2, we say that $G_1 \sim G_2$ if and only if $\Lambda(G_1) = \Lambda(G_2)$ and $\chi_{G_1} = \chi_{G_2}$. Clearly, for every $I \subseteq \mathbb{Z}^+$, $\sim$ is an equivalence relation with finite number of equivalence classes.

Refinement proof. The result will follow if we prove that $\sim$ is a refinement of $\equiv_\Pi$. For this, we claim that if $G_1 \sim G_2$, then $G_1 \equiv_\Pi G_2$, or, equivalently, there is some constant c depending on G_1 and G_2 such that

$$\forall (F, k) \in \mathcal{F} \times \mathbb{Z} \quad (G_1 \oplus F, k) \in \Pi \Leftrightarrow (G_2 \oplus F, k + c) \in \Pi. \tag{51}$$

Suppose that $G_1 \sim G_2$. Let $(F, k) \in \mathcal{F} \times \mathbb{Z}$ such that $(G_1 \oplus F, k) \in \Pi$. Our target is to prove that $(G_2 \oplus F, k + c) \in \Pi$. (The proof for other direction of (51) is symmetric and thus omitted.) Let us also assume that G_1 and G_2 are boundaried graphs with label set I and $\delta(G_1) = B$.

The fact that $(G_1 \oplus F, k) \in \Pi$ means that $G_1 \oplus F$ contains a collection of k disjoint cycles. Let $\mathcal{C}$ be such a collection of maximum size in $G_1 \oplus F$. Clearly, $|\mathcal{C}| \geq k$. We partition $\mathcal{C}$ into four sets $\mathcal{C}_{G_1}$, $\mathcal{C}_B$, $\mathcal{C}_F^B$, and $\mathcal{C}_F$, where

—$\mathcal{C}_{G_1}$ are the cycles that are entirely inside G_1,
—$\mathcal{C}_B$ are the cycles of $\mathcal{C}$ that are not entirely in G_1 or F,
—$\mathcal{C}_F^B$ are the cycles that are entirely inside F and intersect the boundary B, and
—$\mathcal{C}_F$ are the cycles that are entirely inside F and do not intersect B.

Notice that $|\mathcal{C}_B| + |\mathcal{C}_F^B| \leq |I|$. Graph $G_1 \cap (\bigcup_{C \in \mathcal{C}_B} C)$ is a collection of internally disjoint paths between pairs of terminals in B. By replacing each of these paths by edges, we create graph $R \in \mathcal{R}$. Graph R represents the possibility of linking the pairs corresponding to the edges in $\mathcal{R}$ by disjoint paths inside G_1 in a way that these paths are disjoint from the disjoint cycles in $\mathcal{C}_{G_1}$.

For $i \in \{1, 2\}$, let $\mathcal{C}_i^*$ be a maximum size collection of cycles in G_i, and let $x_i = |\mathcal{C}_i^*|$. Notice that x_1 and x_2 depend only on G_1 and G_2. We claim that $x_1 - |I| \leq |\mathcal{C}_{G_1}|$. Indeed, $\mathcal{C}^* = \mathcal{C}_1^* \cup \mathcal{C}_F$ is also a cycle packing in $G_1 \oplus F$. If $|\mathcal{C}_{G_1}| < x_1 - |I| = |\mathcal{C}_1^*| - |I|$, then $|\mathcal{C}^*| = |\mathcal{C}_1^*| + |\mathcal{C}_F| > |\mathcal{C}_{G_1}| + |I| + |\mathcal{C}_F| \geq |\mathcal{C}_{G_1}| + |\mathcal{C}_B| + |\mathcal{C}_F^B| + |\mathcal{C}_F| = |C|$, contradicting the maximality of $\mathcal{C}$.

We set $c = x_2 - x_1$. By the definition of ζ_G, we have that $|\mathcal{C}_{G_1}| \leq \zeta_{G_1}(R) \leq x_1$. We conclude that $x_1 - |I| \leq \zeta_{G_1}(R) \leq x_1$, and thus $\chi_{G_1}(R) > -\infty$. As $G_1 \sim G_2$, we have that $\chi_{G_1}(R) = \chi_{G_2}(R)$, and therefore $\zeta_{G_2}(R) = \zeta_{G_1}(R) - x_1 + x_2 = \zeta_{G_1}(R) + c \geq |\mathcal{C}_{G_1}| + c$. This in turn means that G_2 contains a collection of disjoint cycles $\mathcal{C}_{G_2}$ and $|\mathcal{C}_{G_2}| = \zeta_{G_2}(R) \geq |\mathcal{C}_{G_1}| + c$ and $|E(R)|$ internally vertex-disjoint paths that are also disjoint from the cycles in $\mathcal{C}_{G_2}$, one for each pair of vertices represented by the edges of R.

Notice now that if we take the union of these paths with the graph $F \cap (\bigcup_{C \in \mathcal{C}_B} C)$, we obtain a collection $\mathcal{C}_B'$ of $|\mathcal{C}_B|$ vertex-disjoint cycles in $G_2 \oplus F$ that are also disjoint with the cycles from $\mathcal{C}_{G_2}$. The cycles from $\mathcal{C}_{G_2} \cup \mathcal{C}_B$ are disjoint from cycles $\mathcal{C}_F^B$ and $\mathcal{C}_F$. Therefore, $\mathcal{C}_{G_2} \cup \mathcal{C}_B' \cup \mathcal{C}_F^B \cup \mathcal{C}_F$ is a collection of cycles in $G_2 \oplus F$ that has size at least $|\mathcal{C}_{G_1}| + c + |\mathcal{C}_B| + |\mathcal{C}_F^B| + |\mathcal{C}_F| = k + c$. We conclude that $(G_2 \oplus F, k + c) \in \Pi$, as required. □

The proof that, in general, p-$\mathcal{H}$-PACKING has FII follows the same line as the proof of Lemma 8.5. Instead of cycles, we have minor models of graphs in $\mathcal{H}$, and instead of paths between terminals of the border, we have *partial models* that are parts of minor models of graphs in $\mathcal{H}$ that are cropped by G_1. The signature χ now encodes all of the ways such partial models might be "rooted" in the boundary. This can be done by the "folio" structure introduced in Robertson and Seymour [1995] for doing dynamic programming for the minor-checking problem and the disjoint paths problem on graphs of bounded treewidth. Variants of folios have been used for similar purposes in the work of Adler et al. [2008], Grohe et al. [2011], Kaminski and Thilikos [2012], and Fomin et al. [2012b].

8.4. Subgraph Covering and Packing

Let $\mathcal{S}$ be a finite set of connected graphs. We define the following two general problems.

> p-$\mathcal{S}$-COVERING
> *Input*: A graph G and $k \in \mathbb{Z}^+$
> *Parameter*: k
> *Question*: Is there a $S \subseteq V(G)$ such that $|S| \leq k$ and $G \backslash S$ contain no subgraph isomorphic to a graph from $\mathcal{S}$?

> p-$\mathcal{S}$-PACKING
> *Input*: A graph G and $k \in \mathbb{Z}^+$
> *Parameter*: k
> *Question*: Does there exist k vertex-disjoint subgraphs $G_1, \ldots, G_k$ of G such that each of them contains a subgraph isomorphic to a graph in $\mathcal{S}$?

Let us remark that it is not true, in general, that if $\Pi = p$-$\mathcal{S}$-COVERING or $\Pi = p$-$\mathcal{S}$-PACKING, then Π_g is coverable. However, the problems become coverable if we modify instances by applying the following simple preprocessing rule:

> Redundant Vertex Rule: For a graph G, although this is possible, delete a vertex that does not belong to any subgraph of G isomorphic to any graph in $\mathcal{S}$.

A graph G is RV-$\mathcal{S}$-*reduced* if each its vertex belongs to a subgraph isomorphic to a graph in $\mathcal{S}$. We denote by $\mathcal{R}(\mathcal{S})$ the set of all RV-$\mathcal{S}$-*reduced* graphs.

LEMMA 8.7. *Let Π be either p-$\mathcal{S}$-COVERING or p-$\mathcal{S}$-PACKING. There is a polynomial time algorithm transforming $(G, k) \in \Pi_g$ into an equivalent instance $(G', k) \in \Pi_g^{\mathsf{RV}} = \Pi_g \Cap \mathcal{R}(\mathcal{S})$.*

PROOF. Let s be the maximum diameter of a graph in $\mathcal{S}$, and let G be a graph of genus g. We can perform the Redundant Vertex Rule in $O(|V(G)|^2)$ time by checking for every vertex $v \in V(G)$ if the subgraph $G^s(v)$ induced by $\mathbf{B}_G^s(v)$ has a subgraph isomorphic to a graph in $\mathcal{S}$ containing vertex v. By Proposition 6.3, the treewidth of $G^s(v)$ is bounded by some function of s and g only, and thus for every v, such a check can be performed in time $O(|V(G)|)$ (e.g., see Eppstein [2000]). $\square$

We are now ready to prove the following lemma.

LEMMA 8.8. *Let Π be p-$\mathcal{S}$-COVERING or p-$\mathcal{S}$-PACKING. Then Π_g^{RV} is coverable.*

PROOF. Let s be the maximum diameter of a graph in $\mathcal{S}$, and let Υ = p-$\mathcal{S}$-COVERING. Let (G, k) be a YES-instance of Υ_g^{RV}, and let S be a vertex set of size at most k, such that each subgraph of G that is isomorphic to some graph in $\mathcal{S}$ intersects S. Consider an embedding of G in some surface of Euler genus at most g. As $G \in \mathcal{R}(\mathcal{S})$, every vertex in G is within distance at most s from S. Therefore, $\mathbf{B}_G^s(S) = V(G)$. By Observation 3, $\mathbf{R}_G^{2s}(S) \supseteq \mathbf{B}_G^s(S)$, and thus Υ_g^{RV} has the r-coverability property for $r = 2s$.

Assume now that Ψ =p-$\mathcal{S}$-PACKING. To prove the coverability of Ψ_g^{RV}, we will prove that $\bar{\Psi}_g^{\mathsf{RV}} = ((\Sigma^* \times \mathbb{Z}^+) \backslash \Psi_g^{\mathsf{RV}}) \Cap \mathcal{G}_g$ has the r-coverability property. Let c be the maximum number of vertices in a graph of $\mathcal{S}$. We claim that if (G, k) is a NO-instance for Ψ_g^{RV}, where $G \in \mathcal{G}_g$, then (G, ck) is a YES-instance of Υ_g^{RV}. Indeed, as (G, k) is a NO-instance, G does not contain k vertex-disjoint subgraphs from $\mathcal{S}$. A set S of vertices of size $\leq k \cdot c$ "hitting" all subgraphs of G isomorphic to graphs in $\mathcal{S}$ can be constructed by the following greedy procedure:

Initialize $S = \emptyset$ and, as long as G contains a subgraph that is isomorphic to some graph in $\mathcal{S}$, add all of its vertices to S and remove them from G.

Notice that the preceding procedure cannot be applied more than $k - 1$ times; otherwise, the removed graphs would constitute a vertex packing of graphs of $\mathcal{S}$ in G. When the procedure cannot be applied anymore, the set S intersects every subgraph of G that is isomorphic to some graph from $\mathcal{S}$ and $|S| \leq c \cdot (k - 1)$. Therefore, (G, ck) is a YES-instance of Υ_g^{RV}, which is already shown to be coverable. Now the coverability of Ψ_g^{RV} follows from Lemma 8.1. $\square$

Using a modification of the proof of Lemma 8.4, it is possible to show that p-$\mathcal{S}$-COVERING has FII. The proof that p-$\mathcal{S}$-PACKING has FII follows the same steps as in the proof of Lemma 8.6. The only difference in all cases is that we work with subgraphs instead of minors.

8.5. Domination and Its Variants

Given two integers $r, q \in \mathbb{Z}^+$, a graph G, and a set $S \subseteq V(G)$, we say that S is a (q, r)-*dominating set of* G if for every vertex x in $V(G) \backslash S$, there are at least q vertices in S

within distance at most r from x. We define a series of problems related to domination. In all of them, the input is a graph G and a parameter $k \in \mathbb{Z}^+$. In the following, we mention the variants and the questions corresponding to each of them:

—p-r-DOMINATING SET: Is there a $(1, r)$-dominating set S of size at most k in G? For $r = 1$, the problem is known as p-DOMINATING SET.
—p-q-THRESHOLD DOMINATING SET: Is there a $(q, 1)$-dominating set S of size at most k in G?
—p-EFFICIENT DOMINATING SET: Is there a $(1, 1)$-dominating set S of size at most k in G such that $G[S]$ is edgeless (i.e., S is an independent set) and each vertex from $V(G)\backslash S$ is adjacent to exactly one vertex in S? This problem is also known as p-PERFECT CODE.
—p-CONNECTED DOMINATING SET: Is there a $(1, 1)$-dominating set S of size at most k in G such that $G[S]$ is connected?

LEMMA 8.9. *If* Π *is one of the following problems,* p-r-DOMINATING SET*,* p-q-THRESHOLD DOMINATING SET*, or* p-EFFICIENT DOMINATING SET*, then for every* $g \in \mathbb{Z}^+$*,* Π_g *is coverable and has FII.*

PROOF. For all of these problems, Π_g is $2r$-coverable by definition because if S is a (q, r)-dominating set of G and G is embeddable in some surface of Euler genus at most g, then by Observation 3, $\mathbf{B}^r_G(S) \subseteq \mathbf{R}^{2r}_G(G)$.

By Lemma 8.2, it is enough to prove that each of the problems has FII. We start from p-r-DOMINATING SET. Since p-r-DOMINATING SET is a p-MIN-CMSO$[\psi]$ problem, then by Lemma 7.3, it is enough to prove that it is strongly monotone. For a boundaried graph G with label set I and boundary $\delta(G) = B$, let $S'' \subseteq V(G)$ be a minimum-size r-dominating set of G. We put $W = S'' \cup B$. For a boundaried structure $(G', S') \in \mathcal{U}_I$, let $S^* \subseteq V(G)$ be a set of minimum size such that $S^* \cup S'$ is an r-dominating set of $G \oplus G'$. Thus, $\zeta_G(G', S') = |S^*|$. Observe that $S^* \cup B$ is an r-dominating set of G, and hence $|S''| \leq |S^*| + |B|$. Therefore, $|W| = |S'' \cup B| \leq |S''| + |B| \leq |S^*| + 2|I| = \zeta_G(G', S') + 2|I|$. Additionally, observe that $W \cup B$ is an r-dominating set of G', and thus $W \cup S'$ is an r-dominating set of $G \oplus G'$. This implies that $(G \oplus G', S \cup S') \in \Pi$, and the strong monotonicity of p-r-DOMINATING SET follows.

The proof that p-q-THRESHOLD DOMINATING SET is strongly monotone is based on the same observations as the proof for p-r-DOMINATING SET and thus omitted. To prove that p-EFFICIENT DOMINATING SET has FII, we use the fact that

$$p\text{-EFFICIENT DOMINATING SET} = p\text{-1-DOMINATING SET} \Cap \mathcal{G}^{\text{eds}},$$

where $\mathcal{G}^{\text{eds}}$ is the class of all graphs that have an efficient dominating set. The equality follows from a theorem of Bange et al. [1988], asserting that if a graph G has an efficient dominating set, then the size of the minimum efficient dominating set is equal to the size of the minimum dominating set of G. As $\mathcal{G}^{\text{eds}}$ is CMSO definable, p-EFFICIENT DOMINATING SET has FII by Lemma 7.5. □

In the remaining part of this section, we prove that when Π is p-CONNECTED DOMINATING SET, then Π_g is coverable and has FII. For this, we first need some auxiliary definitions and results on connected domination. Given a graph G and a set $V(G)$, we say that a dominating set S is a *component-wise* connected dominating set of G if for every connected component C of G, $C[S \cap V(C)]$ is connected. In particular, if G is connected, then every component-wise dominating set of G is also a connected dominating set of G.

We need the following proposition attributed to Duchet and Meyniel [1982].

Proposition 8.10. *Let G be a connected graph, and let Q be a dominating set of G, such that $G[Q]$ has at most ρ connected components. Then there exists a set $Z \subseteq V(G)$ of size at most $2 \cdot (\rho - 1)$ such that $Q \cup Z$ is a connected dominating set in G.*

Lemma 8.11. *Let G be a graph, and let B be a subset of G. In addition, let R be a component-wise connected dominating set of G. Then there exists a set $S \supseteq R \cup B$ that is also a component-wise connected dominating set of G and has at most $|R| + 3|B|$ vertices.*

Proof. Let $\mathcal{C}$ be the set of connected components of G. For $C \in \mathcal{C}$, let $B_C = V(C) \cap B$ and $R_C = R \cap V(C)$. Observe that $C[B_C \cup R_C]$ cannot have more than $1 + |B_C|$ connected components. By Proposition 8.10, there exists a set $Z_C \subseteq V(C)$ such that $Z_C \cup R_C \cup B_C$ induces a connected subgraph of C such that $|Z_C| \leq 2|B_C|$. This means that $|B_C \cup R_C \cup Z_C| \leq |R_C| + 3|B_C|$. Moreover, as R_C is a dominating set of C, the same holds for its superset $B_C \cup R_C \cup Z_C$. Therefore, the set $S = \bigcup_{C \in \mathcal{C}} B_C \cup R_C \cup Z_C$ is a component-wise dominating set of G that contains $B \cup R$. It is now easy to check that $|S| \leq |R| + 3|B|$. □

Lemma 8.12. *Let G and G' be boundaried graphs with label set I and boundary $\delta(G) = B$. In addition, let $S^* \subseteq V(G)$ and $S' \subseteq V(G')$ such that $S^* \cup S'$ is a component-wise connected dominating set of $G \oplus G'$. Then G contains a component-wise connected dominating set S^+ of size at most $3|B| + |S^*|$.*

Proof. We first prove the lemma under the assumption that $H = G \oplus G'$ is a connected graph. Let us remark that G is not necessarily connected. Notice that $Q = S^* \cup B$ is a dominating set of G. Let $C_1, \ldots, C_\mu$ be the connected components of G and, for each $i \in \{1, \ldots, \mu\}$, let $Q_i^1, \ldots, Q_i^{\delta_i}$ be the vertex sets of the connected components of $C_i[V(C_i) \cap Q]$. We claim that $\sum_{1 \leq i \leq \mu} \delta_i \leq |B| + 1$. Indeed, if $S^* \cup S'$ does not intersect B, then since $H[S^* \cup S']$ is connected, we have that $G[S^* \cup S']$ is connected, and in this case, Q may have at most $|B| + 1$ connected components; therefore, $\sum_{1 \leq i \leq \mu} \delta_i \leq |B| + 1$. In case $S^* \cup S$ intersects B, then each connected component of Q should contain at least one vertex of B, and, again, we have $\sum_{1 \leq i \leq \mu} \delta_i \leq |B| < |B| + 1$.

We now apply Proposition 8.10 for the sets $Q_i^1, \ldots, Q_i^{\delta_i}$ of the graph C_i, for each $i \in \{1, \ldots, \mu\}$. That way, we find, for every $i \in \{1, \ldots, \mu\}$, a collection of sets $Z_1, \ldots, Z_\mu$, where Z_i is a connected dominating set of C_i. This means that $S^+ = \bigcup_{1 \leq i \leq \mu} Z_i$ is a component-wise connected dominating set of G. By Proposition 8.10, $|Z_i| \leq 2(\delta_i - 1) + |V(C_i) \cap Q|$. We now have that

$$\begin{aligned} |S^+| &= \sum_{i=1}^{\mu} |Z_i| \\ &\leq \sum_{i=1}^{\mu} 2(\delta_i - 1) + \sum_{i=1}^{\mu} |V(C_i) \cap Q| \\ &\leq 2|B| + |Q| = 3|B| + |S^*|, \end{aligned}$$

as required.

If $G \oplus G'$ is not a connected graph, then the required component-wise connected dominating set is the union of the component-wise connected dominating sets obtained if we apply the preceding proof for each of the connected components of $G \oplus G'$. □

We also need the following lemma. The proof is based on the definition of a connected dominating set and is omitted.

LEMMA 8.13. *Let G and G' be boundaried graphs with label set I and boundary $\delta(G) = B$ such that $C = G \oplus G'$ is connected. In addition, let $S^* \subseteq V(G)$ and $S' \subseteq V(G')$ be such that $S^* \cup S'$ is a connected dominating set of C. Let $S \subseteq V(G)$ be a component-wise dominating set of G such that $B \subseteq S$. Then $S \cup S'$ is a connected dominating set of $G \oplus G'$.*

LEMMA 8.14. *If Π = p-CONNECTED DOMINATING SET, then for every $g \in \mathbb{Z}^+$, Π_g is coverable and has FII.*

PROOF. The coverability of Π_g is trivial. To show that p-CONNECTED DOMINATING SET has FII, we define the following auxiliary problem:

$$\Pi' = \{(G, k) \mid G \text{ has a component-wise connected dominating set } S\}.$$

Notice that p-CONNECTED DOMINATING SET $= \Pi' \Cap \mathcal{G}_{\text{con}}$, where $\mathcal{G}_{\text{con}}$ is the class of all connected graphs. Let us remark that $\mathcal{G}_{\text{con}}$ is CMSO definable and Π' is a p-MIN-CMSO$[\psi]$ problem.

Let G be a boundaried graph with label set I and boundary $\delta(G) = B$. Let R be a minimum-size component-wise dominating set of G. By Lemma 8.11, G has a component-wise connected dominating set W that contains the boundary of G ($B \subseteq W$) as a subset and $|W| \leq |R| + 3|I|$.

For a boundaried structure $(G', S') \in \mathcal{U}_I$, let $S^* \subseteq V(G)$ be a set of minimum-size subset of G such that $S^* \cup S'$ is a component-wise connected dominating set of $G \oplus G'$. Thus, $\zeta_G(G', S') = |S^*|$. From Lemma 8.12, G contains a component-wise connected dominating set S^+ of size at most $|S^*| + 3|I|$. By the definition of R, we have that $|R| \leq |S^+| \leq |S^*| + 3|I| = \zeta_G(G', S') + 3|I|$, and therefore $|S| \leq |R| + 3|I| \leq \zeta_G(G', S') + 6|I|$.

To prove that $(G \oplus G', W \cup S') \in \Pi'$, we have to show that $W \cup S'$ is a component-wise connected dominating set of $G \oplus G'$. Let $\mathcal{C}$ be the set of the connected components of $G \oplus G'$, and for every $C \in \mathcal{C}$, we set $G_C = G[V(C)]$, $G'_C = G'[V(C)]$, $S^*_C = S^* \cap V(C)$, $W_C = W \cap V(C)$, $S'_C = S' \cap V(C)$, and $B_C = B \cap V(C)$. Notice that $C = G_C \oplus G'_C$. As $S^* \cup S'$ is a component-wise dominating set of $G \oplus G'$, we have that the set $S^*_C \cup S'_C$ is a connected dominating set of C. Moreover, the fact that S is a component-wise dominating set of G implies that W_C is also a component-wise dominating set of G_C. Recall that the boundary of G is contained in W, and therefore $B \subseteq W$, which implies that $B_C \subseteq W_C$. From Lemma 8.13, $W_C \cup S'_C$ is a connected dominating set of C. Therefore, $W \cup S' = \bigcup_{C \in \mathcal{C}} W_C \cup S'_C$ is a component-wise connected dominating set of $G \oplus G'$, as required. □

Using ideas similar to those in the proof of Lemma 8.9, it is possible to prove that other problems, such as p-CONNECTED VERTEX COVER, p-EDGE DOMINATING SET, or p- CYCLE DOMINATION, have FII.

8.6. Scattered Sets

Given an $r \in \mathbb{Z}^+$, a graph G, and a set $S \subseteq V(G)$, we say that S is an *r-independent set* if every two vertices in S have distance greater than r.

We consider the following problem.

p-r-SCATTERED SET
Input: A graph G and a $k \in \mathbb{Z}^+$
Parameter: k
Question: Is there an r-independent set in G of size at least k?

LEMMA 8.15. *For every positive integer r and every $g \in \mathbb{Z}^+$, if Π^r = r-SCATTERED SET, then Π^r_g is coverable.*

Proof. To prove the coverability of Π_g^r, we will prove that $\Psi_g = ((\Sigma^* \times \mathbb{Z}^+)\setminus\Pi_g^r) \Cap \mathcal{G}_g$ has the r-coverability property for some constant c that depends on g and r. Let (G, k) be a NO-instance of Π_g^r. This means that G does not contain any r-independent set of size k. According to the result in Dvorak [2013], G has an r-dominating set of size $c \cdot k$, where c is a constant depending on the Euler genus of G (actually, the result of Dvorak [2013] holds for much more general classes of sparse graphs that include graphs of bounded Euler genus). Recall that from Observation 3, given an embedding of G in a surface of Euler genus $\leq g$, we have that $\mathbf{R}_G^{2r} \subseteq \mathbf{B}_G^r(S)$, and therefore Ψ_g has the c-coverability property for $c = \max\{r, g\}$. □

We present in detail the proof of the following lemma, as it is based on slightly different ideas than the one used in Lemma 8.6.

Lemma 8.16. *For every positive integer r, if* $\Pi^r = p$-r-Scattered Set, *then* Π_g *has FII.*

Proof. Using Lemma 8.2, we prove instead that Π^r has FII. In the following, we prove this fact by adapting the three-stage machinery of the proof of Lemma 8.6.

Characteristic. Let G be a boundaried graph with label set I and the boundary $\delta(G) = B$. Furthermore, let $\ell_G : I \times I \to \{0, \dots, r\}$ be a function that for $i, j \in I$ defines

$$\ell_G(i, j) = \min\left\{\mathbf{dist}_G\big(\lambda^{-1}(i), \lambda^{-1}(j)\big), r\right\}.$$

In addition, let $\mathcal{S}$ be the set containing all functions mapping the integers of I to integers in $\{0, \dots, r\} \cup \{\infty\}$. Given a $\sigma \in \mathcal{S}$, we define $\zeta_G(\sigma)$ as the maximum size of an r-independent set S in G with the property that for every $i \in I$, the distance in G between $\lambda^{-1}(i)$ and every vertex in S is at least $\sigma(i)$. As the empty set is always such a set, it holds that $\forall_{\sigma\in\mathcal{S}}\ \zeta_G(\sigma) \geq 0$.

Definition of equivalence. Let $\sigma^{(0)} \in \mathcal{S}$ such that $\forall_{i\in\lambda(B)}\ \sigma^{(0)}(i) = 0$. We also set $x_G = \zeta_G(\sigma^{(0)})$. We have that $\forall_{\sigma\in\mathcal{S}}\ \zeta_G(\sigma) \leq x_G$. We define a function $\chi_G : \mathcal{S} \to \{-\infty\}\cup\{-2t, \dots, 0\}$ as follows:

$$\chi_G(\sigma) = \begin{cases} \zeta_G(\sigma) - x_G & \text{if } x_G - 2t \leq \zeta_G(\sigma) \leq x_G \\ -\infty & \text{otherwise.} \end{cases}$$

Given two boundaried graphs G_1 and G_2, we say that $G_1 \sim G_2$ if $\Lambda(G_1) = \Lambda(G_2)$, $\ell_{G_1} = \ell_{G_2}$, and $\chi_{G_1} = \chi_{G_2}$. Notice that for every finite $I \subseteq \mathbb{Z}^+$, $\sim$ is an equivalence relation with finitely many equivalence classes.

Refinement proof. The result will follow if we prove that $\equiv_{\Pi^r}$ is a refinement of $\sim$. For this, we claim that if $G_1 \sim G_2$, then $G_1 \equiv_{\Pi^r} G_2$, or, equivalently, that there is some constant c, depending on G_1 and G_2, such that

$$\forall (F, k) \in \mathcal{F} \times \mathbb{Z} \quad (G_1 \oplus F, k) \in \Pi^r \Leftrightarrow (G_2 \oplus F, k + c) \in \Pi^r. \tag{52}$$

Suppose that $G_1 \sim G_2$. This implies that $\Lambda(G_1) = \Lambda(G_2)$. Let $\Lambda(G_1) = \Lambda(G_2) = I$ and $|I| = t$. Let $(F, k) \in \mathcal{F} \times \mathbb{Z}$ such that $(G_1 \oplus F, k) \in \Pi^r$. Our target is to prove that $(G_2 \oplus F, k + c) \in \Pi^r$ (the other direction of (52) is symmetric).

The fact that $(G_1 \oplus F, k) \in \Pi^r$ means that $(G_1 \oplus F)$ contains an r-independent set S, where $|S| \geq k$. Let B be the boundary of G_1 (i.e., $\delta(G_1) = B$), and let $S_1 = S \cap V(G_1)$ and $S_F = S\setminus S_1$. In addition, let λ_1 and λ_2 be the labelings of boundaries of G_1 and G_2, respectively. We define σ as follows: for $i \in I$, set $\sigma(i)$ to be the minimum distance of a vertex of S_1 from $\lambda_1^{-1}(i)$ in G_1. By the definition of ζ_{G_1}, we have that $\zeta_{G_1}(\sigma) \geq |S_1|$. Before we proceed, we need to prove the following claim:

Claim: $|S_1| \geq x_{G_1} - 2t$. Let S'_1 be an r-independent set of G_1 such that $|S'_1| = x_{G_1}$. Mark in S'_1 all vertices that are within distance at most $\lfloor \frac{r}{2} \rfloor$ from B, and denote by S^*_1 the set of the nonmarked vertices of S'_1. Notice that S^*_1 is an r-independent set of G_1. The proof of the claim is a consequence of the following two subclaims:

Subclaim 1: $|S^*_1| \geq x_{G_1} - t$. For this, it is enough to prove that no more than $|B|$ vertices can be marked from S'_1. Indeed, if this is not the case, then there should exist two vertices x and y in S'_1 that are within distance at most $\lfloor \frac{r}{2} \rfloor$ from some vertex z of B. Then the distance between x and y should be less than $2 \cdot \lfloor \frac{r}{2} \rfloor \leq r$, a contradiction to the fact that S'_1 is an r-independent set of G_1.

Subclaim 2: $|S_1| \geq |S^*_1| - t$. For this, we mark in S the vertices of $G_1 \oplus F$ that are within distance at most $\lfloor \frac{r}{2} \rfloor$ from some vertex of B. As earlier, the marked vertices cannot be more than $|B|$. Let S^- be the set obtained from S after removing the marked vertices. Notice that $|S^-| \geq |S| - t$, and therefore $|S^- \cap V(G_1)| + |S^- \setminus V(G_1)| \geq |S| - t$. Notice that $S^- \cap V(G_1)$ is an r-independent set of G_1, and therefore $|S^- \cap V(G_1)| \leq x_G$. Notice that $S^*_1 \cup (S^- \setminus V(G_1))$ is an r-independent set of $G_1 \oplus F$. Indeed, if there are two vertices $x \in S^*_1$ and $y \in S^- \setminus V(G_1)$ within distance r, then either x or y would be within distance $\lfloor \frac{r}{2} \rfloor$ from some vertex in B, a contradiction. We obtain that $|S^*_1| + |S^- \setminus V(G_1)| = |S^*_1 \cup (S^- \setminus V(G_1))| \leq |S| \leq |S^-| + t = |S^- \cap V(G_1)| + |S^- \setminus V(G_1)| + t$, and therefore $|S^*_1| \leq |S^- \cap V(G_1)| + t \leq |S_1| + t$.

We just proved that $\zeta_{G_1}(\sigma) \geq |S_1| \geq x_{G_1} - 2t$. This means that $\chi_G(\sigma) > -\infty$. As $G_1 \sim G_2$, we have that $\ell_{G_1} = \ell_{G_2}$ and $\chi_{G_1}(\sigma) = \chi_{G_2}(\sigma)$. By the definition of χ_G, we obtain that $\zeta_{G_2}(\sigma) = \zeta_{G_1}(\sigma) - \zeta_{G_1}(\sigma^{(0)}) + \zeta_{G_2}(\sigma^{(0)}) = \zeta_{G_1}(\sigma) + c \geq |S_{G_1}| + c$, where c is a constant depending only on G_1 and G_2. This implies that there exists an r-independent set S_{G_2} in G_2 with least $|S_{G_1}| + c$ vertices, and for every $i \in \lambda_2(B)$, the distance in G_2 between $\lambda_2^{-1}(i)$ and the vertices in S_2 is at least $\sigma(i)$. The facts that $\ell_{G_1} = \ell_{G_2}$ and $\chi_{G_1}(\sigma) = \chi_{G_2}(\sigma)$ together imply that $S_{G_2} \cup S_F$ is an r-independent set of $G_2 \oplus F$ of size $|S_{G_2} \cup S_F| = |S_{G_2}| + |S_F| \geq |S_{G_1}| + |S_F| + c \geq |S_1| + |S_F| + c \geq k + c$. We conclude that $(G_2, k + c) \in \Pi^r$, as required. □

8.7. Problems on Directed Graphs

Our results also apply to problems on directed graphs whose underlying undirected graph is of bounded genus. In this direction, we mention three problems considered in the literature. In all cases, the input is a directed graph $D = (V, A)$, where V is the set of its vertices and A is the set of its directed edges (i.e., $A \subseteq V \times V$):

—p-Directed Domination [Alber et al. 2006b]: Is there a subset $S \subseteq V$ of size at most k such that for very vertex $u \in V \setminus S$ there is a vertex $v \in S$ such that $(u, v) \in A$? Such a set S is called a *directed dominating set* of D.
—p-Independent Directed Domination[2] [Gutin et al. [2005]: Is there a subset $S \subseteq V$ of size at most k such that S is an independent set and for every vertex $u \in V \setminus S$ there is a vertex $v \in S$ such that $(u, v) \in A$?
—p-Maximum Internal Out-branching [Gutin et al. 2009]: Does D contain a directed rooted spanning tree, an out-branching, with at least k internal vertices?

To formally state our results, we extend the notion of coverability to directed graphs by applying the definitions to their underlying undirected graphs.

Lemma 8.17. *The following statements hold:*

[2]In the literature, it is known as p-Kernels. We refer to it differently here to avoid confusion with problem kernels.

—*Let* Π *be either* p-Independent Directed Domination *or* p-Maximum Internal Out-branching. *Then* Π_g *is a coverable* p-min-CMSO[ψ] *problem.*
—*Let* Π *be* p-Directed Domination. *Then* Π_g *is a coverable problem and has FII.*

Proof. Problems p-Independent Directed Domination and p-Directed Domination can easily be seen to be p-min-CMSO[ψ] problems, whereas p-Maximum Internal Out-branching can be proved to be a p-max-CMSO[ψ] problem. The strong monotonicity of p-Directed Domination can be proved using the same arguments as in the proof of Lemma 8.9. This, together with Lemmata 7.3 and 8.2, implies that for Π=p-Directed Domination, Π_g has FII.

p-Independent Directed Domination and p-Directed Domination are coverable by definition. Let Π=p-Maximum Internal Out-branching. We claim that if $(D, k) \notin \Pi$, then the underlying undirected graph of D has a dominating set of size at most $k - 1$. For this, let $k_0 = \max\{k' \mid (D, k') \in \Pi\}$ and observe that $k_0 < k$. Moreover, it also holds that $(D, k_0) \in \Pi$, whereas $(D, k_0 + 1) \notin \Pi$. These two facts together imply that D has a rooted directed spanning tree with *exactly* k_0 internal vertices and all other vertices of D being its leaves. These internal vertices form a dominating set for the underlying undirected graph of D. As $k_0 < k$, the underlying undirected graph of D has a dominating set of size at most $k - 1$. Then the coverability of Π_g follows from the coverability of p-Dominating Set and Lemma 8.1. □

8.8. A Direct Proof of FII for a Minimization Problem

Although Lemma 7.3 is very useful for showing that a concrete problem has FII, sometimes a minimization problem may have FII even though it may not be strongly monotone. For example, consider the following problem. Let $s \geq 3$ be an integer.

> s-Cycle Transversal
> *Input*: A graph G and a $k \in \mathbb{Z}^+$
> *Parameter*: k
> *Question*: Is there an edge subset $S \subseteq E(G)$ such that $G' = G \backslash S$ does not contain any cycle of length at most s (i.e., G' has girth more than s)?

Notice that for each integer $s \geq 3$, the preceding problem is the edge deletion counterpart of Edge-$\mathcal{S}$-Covering when $\mathcal{S}$ contains the cycles of size at least 3 and at most s.

Lemma 8.18. *If* Π^s =s-Cycle Transversal*, then* Π^s_g *has FII.*

Proof. Using Lemma 8.2, we prove instead that Π^s has FII. We present the proof in three stages, as we did in the cases of Lemmata 8.6 and 8.16.

Characteristic. Let G be a boundaried graph with label set I and the boundary $\delta(G) = B$. Let $|I| = t$. We use the term s-*cycle* for a cycle of length at most s. Let X be the set of unordered pairs of distinct indices in I and $\mathcal{H}$ be the set containing all functions from X to $\{0, \ldots, s\}$. We define the function $\zeta_G : \mathcal{H} \rightarrow \mathbb{Z}^+$ such that, given a function $f \in \mathcal{H}$, $\zeta_G(f)$ is the size of a minimum set of edges S in G such that the following hold:

—the graph $G \backslash S$ has girth $> s$, and
—for every $\{i, j\} \in I$, the distance in $G' = G \backslash S$ between $\lambda^{-1}(i)$ and $\lambda^{-1}(j)$ is at least $f(i, j) + 1$. In other words, $\mathbf{dist}_{G'}(\lambda^{-1}(i), \lambda^{-1}(j)) \geq f(i, j) + 1$.

In case a set satisfying the preceding conditions does not exist, we set $\zeta_G(f) = \infty$.

Definition of equivalence. We denote by $f^{\min}$ the function in $\mathcal{H}$ where for all $\{i, j\} \in X$, $f^{\min}(\{i, j\}) = 0$. Notice that $\zeta_G(f^{\min}) < \infty$ (just take $S = E(G)$). We set $x_G = \zeta_G(f^{\min})$.

The definition of ζ_G implies that

$$\forall f \in \mathcal{H} \quad x_G \leq \zeta_G(f). \tag{53}$$

We now define the *signature* of G as the function $\chi_G : \mathcal{H} \to \{0, \ldots, 3\binom{t}{2}\} \cup \{\infty\}$, where

$$\chi_G(f) = \begin{cases} \zeta_G(f) - x_G & \text{if } x_G \leq \zeta_G(f) \leq x_G + 3\binom{t}{2} \\ \infty & \text{otherwise.} \end{cases} \tag{54}$$

We say that $G_1 \sim G_2$ if $\Lambda(G_1) = \Lambda(G_2)$ and $\chi_{G_1} = \chi_{G_2}$. Notice that the number of different signatures is bounded by some function of t and s. Clearly, for every $I \subseteq \mathbb{Z}^+$, $\sim$ is an equivalent relation with finitely many equivalence classes.

Refinement proof. The result will follow if we prove that $\sim$ is a refinement of $\equiv_\Pi$. For this, we claim that if $G_1 \sim G_2$, then $G_1 \equiv_\Pi G_2$, or, equivalently, that there is some constant c, depending on G_1 and G_2, such that

$$\forall (F, k) \in \mathcal{F} \times \mathbb{Z} \quad (G_1 \oplus F, k) \in \Pi \Leftrightarrow (G_2 \oplus F, k + c) \in \Pi. \tag{55}$$

Suppose that $G_1 \sim G_2$. Let $(F, k) \in \mathcal{F} \times \mathbb{Z}$ such that $(G_1 \oplus F, k) \in \Pi$. Our target is to prove that $(G_2 \oplus F, k + c) \in \Pi$ (the other direction of (55) is symmetric and is omitted).

The fact that $(G_1 \oplus F, k) \in \Pi$ means that there is a set $S \subseteq E(G_1 \oplus F)$ of edges such that all cycles in $(G_1 \oplus F)\backslash S$ have length $> s$. Recall that λ_G is an injective labeling from the boundary of the graph to I. We denote by λ_1, λ_2, and λ_F the labelings of the boundaried graphs G_1, G_2, and F, respectively. Let $B = \lambda_1^{-1}(\Lambda(G_1) \cap \Lambda(F))$ and $B' = \lambda_2^{-1}(\Lambda(G_2) \cap \Lambda(F))$. Since G_1, G_2, and F are boundaried graphs with label set I, we have that $|B|, |B'| = |I| = t$. In addition, let $S_{G_1} = E(G_1) \cap S$ and $S_F = E(F) \cap S$. The set $\mathcal{C}$ of s-cycles in $G_1 \cup F$ is partitioned into three sets:

—$\mathcal{C}_1$ are the cycles in $\mathcal{C}$ that are entirely inside G_1,
—$\mathcal{C}_F$ are the cycles in $\mathcal{C}$ that are entirely inside F, and
—$\mathcal{C}_B$ are the cycles in $\mathcal{C}$ that contain both edges that are not in G_1 and edges that are not in F (i.e., $\mathcal{C}_B = \mathcal{C}\backslash(\mathcal{C}_{G_1} \cup \mathcal{C}_F)$).

Observe that S_F intersects all s-cycles in $\mathcal{C}_F$ and the set S_{G_1} intersects all s-cycles in $\mathcal{C}_1$. Observe that $S_{G_1} \cap S_F$ contains only edges with both endpoints in B, and therefore $|S_{G_1} \cap S_F| \leq \binom{t}{2}$. This implies that

$$|S_{G_1}| + |S_F| - \binom{t}{2} \;\leq\; |S|. \tag{56}$$

Recall that $x_{G_1} = \zeta_{G_1}(f^{\min})$. We prove the following claim. Let x_{G_1} denote the cardinality of a minimum sized subset of $E(G_1)$ intersecting all s-cycles in G_1:

Claim: $|S_{G_1}| \leq x_{G_1} + 3\binom{t}{2}$.

Proof of Claim: Let $S^*_{G_1}$ be a minimum size subset of $E(G_1)$ intersecting all s-cycles in G_1. By definition, $|S^*_{G_1}| = x_{G_1}$. Notice that the set $S^*_{G_1} \cup S_F$ meets all cycles in $\mathcal{C}_1 \cup \mathcal{C}_F$. Let $\mathcal{C}^\bullet_B$ be the cycles of $\mathcal{C}_B$ that are not met by $S^*_{G_1} \cup S_F$.

Our first aim is to find a set S_B of at most $2\binom{t}{2}$ edges that interest all cycles of $\mathcal{C}^\bullet_B$. Observe that each cycle in $\mathcal{C}^\bullet_B$ meets at least two vertices in B. Let W be the set of pairs in X that are met by the cycles in $\mathcal{C}^\bullet_B$. For each pair $p = \{x, y\}$, we denote by $\mathcal{Q}^{\text{left}}_p$ (respectively, $\mathcal{Q}^{\text{right}}_p$) the set of all (x, y)-paths in G_1 that belong to cycles in $\mathcal{C}^\bullet_B$. We claim that for each $p = \{x, y\}$, where $x, y \in B$, at most one of the (x, y)-paths in $\mathcal{Q}^{\text{left}}_p$ can have length at most $s/2$. Suppose in contrary that P_1, P_2 are two (x, y)-paths of G_1 of length $\leq s/2$. The union of P_1 and P_2 contains a cycle $C_{x,y}$ that is entirely in G_1.

By the definition of $\mathcal{C}_B^{\bullet}$, we have that $C_{x,y}$ does not contain any edge e from $S^*_{G_1}$. This contradicts the fact that $S^*_{G_1}$ intersects all s-cycles in G_1. Therefore, for each $p = \{x, y\}$, where $x, y \in B$, at most one, say Q_p^{right}, of the (x, y)-paths in $\mathcal{Q}_p^{\text{right}}$ can have length at most $s/2$. Using the same arguments on F, instead of G_1, it follows that for each $p = \{x, y\}$, where $x, y \in B$, at most one, say Q_p^{left}, of the (x, y)-paths in $\mathcal{Q}_p^{\text{left}}$ can have length at most $s/2$.

We now construct the set S_B by adding to it, for each pair $p \in X$, one edge from the Q_p^{right} and one edge from Q_p^{left}. As there are at most $\binom{t}{2}$ pairs in X, we obtain that $|S_B| \leq 2\binom{t}{2}$. Next we prove that S_B meets all cycles in $\mathcal{C}_B^{\bullet}$. For this, let C be a cycle in $\mathcal{C}_B^{\bullet}$. Clearly, there are at least two internally vertex-disjoint paths contained in C (these two paths may not contain all vertices on C) that are entirely inside G_1 or F and have their endpoints in B. Since C is an s-cycle, we have that at least one, say Q, of these paths should have length $\leq s/2$. Let x and y be the endpoints of Q and $p = \{x, y\}$. Clearly, Q belongs in one of $\mathcal{Q}_p^{\text{left}}$ or $\mathcal{Q}_p^{\text{right}}$. Without loss of generality, suppose that Q belongs in $\mathcal{Q}_p^{\text{left}}$. As Q has length at most $s/2$, then Q is the unique path in $\mathcal{Q}_p^{\text{left}}$ that has such a length. By its construction, S_B intersects Q, and as Q is a path of C, S_B intersects C as well.

We just proved that S_B intersects all s-cycles in $\mathcal{C}'_B$ and contains at most $2\binom{t}{2}$ edges. This implies that $S^*_{G_1} \cup S_B \cup S_F$ is intersecting all s-cycles in $\mathcal{C}$. By the definition of S, we have that $|S| \leq |S^*_{G_1} \cup S_B \cup S_F| \leq |S^*_{G_1}| + |S_B| + |S_F|$. Therefore, $|S_{G_1}| + |S_F| - \binom{t}{2} \leq^{(56)} |S| \leq |S^*_{G_1}| + |S_B| + |S_F| \leq x_{G_1} + 2\binom{t}{2} + |S_F|$. We conclude that $|S_{G_1}| \leq x_{G_1} + 2\binom{t}{2} + \binom{t}{2}$, and the claim follows. □

For every pair $\{i, j\} \in X$, let $s(i, j)$ be equal to s minus the distance between $\lambda_F^{-1}(i)$ and $\lambda_F^{-1}(j)$ in F. We define the function $f \in \mathcal{F}$ as follows. For every pair $\{i, j\} \in X$, if $\{\lambda_1^{-1}(i), \lambda_1^{-1}(j)\}$ is an edge of $S_{G_1} \cap S_F$, then define

$$f(i, j) = \max\{1, s(i, j), \},$$

else define $f(i, j) = s(i, j)$. The choice of f and the definition of ζ_{G_1} imply that

$$\zeta_{G_1}(f) \leq |S_{G_1}|. \tag{57}$$

From (53), we have that $x_{G_1} \leq \zeta_{G_1}(f)$. Moreover, from (57) and the preceding claim, we obtain $\zeta_{G_1}(f) \leq x_{G_1} + 3\binom{t}{2}$. By (54), $\chi_{G_1}(f) = \zeta_{G_1}(f) - x_{G_1}$. Recall now that $G_1 \sim G_2$, and hence $\chi_{G_2}(f) = \chi_{G_2}(f)$. This means that $\zeta_{G_2}(f) = \zeta_{G_1}(f) + c$, where $c = x_{G_2} - x_{G_1}$, and clearly c depends only on G_1 and G_2.

Let S_{G_2} be a subset of $E(G_2)$ such that $\zeta_{G_2}(f) = |S_{G_2}|$. By the definition of ζ_{G_2},S_{G_2} has the following properties:

(A) the graph $G_2 \backslash S_{G_2}$ has girth $> s$, and
(B) for every $\{i, j\} \in X$, the distance in $G_2 \backslash S_{G_2}$ between $\lambda_2^{-1}(i)$ and $\lambda_2^{-1}(j)$ is at least $f(i, j) + 1$.

By the definition of f, and Properties (A) and (B), all s-cycles in $G_2 \oplus F$ that are not entirely in F are intersected by S_{G_2}. Hence, $S' = S_{G_2} \cup S_F$ intersects all cycles in $G_2 \oplus F$. Moreover, by the definition of f, we obtain that $S_{G_1} \cap S_F \subseteq S_{G_2}$. This implies that $S' = S_{G_2} \cup S_F = S_{G_2} \cup (S_{G_1} \cap S_F) \cup (S_F \backslash (S_{G_1} \cap S_F)) = S_{G_2} \cup (S_F \backslash (S_{G_1} \cap S_F))$.

We now have that $|S'| \leq |S_{G_2}| + |S_F \backslash (S_{G_1} \cap S_F)| = \zeta_{G_2}(f) + |S_F \backslash (S_{G_1} \cap S_F)| = \zeta_{G_1}(f) + c + |S_F \backslash (S_{G_1} \cap S_F)| \leq^{(57)} |S_{G_1}| + |S_F \backslash (S_{G_1} \cap S_F)| + c = |S_{G_1} \cup S_F| + c = |S| + c \leq k + c$. Therefore, $(G_2 \oplus F, k + c) \in \Pi$, and the lemma follows. □

8.9. Summary of Consequences of Our Results

In this section, we discuss some of the consequences of our main meta-algorithmic results, namely Theorems 1.1 and 1.3.

We start with the consequences of Theorem 1.3 to minimization problems that have FII.

COROLLARY 8.19. *If $g \in \mathbb{Z}^+$ and if Π is one of the following problems, p-*VERTEX COVER*, p-*FEEDBACK VERTEX SET*,* ALMOST OUTERPLANAR*, p-*DIAMOND HITTING SET*, p-*ALMOST-*t*-BOUNDED TREEWIDTH*, p-*ALMOST-*t*-BOUNDED PATHWIDTH*, p-$\mathcal{H}$-*DELETION*, p-*EDGE DOMINATING SET*, p-*MINIMUM-VERTEX FEEDBACK EDGE SET*, p-*DOMINATING SET*, p-r-*DOMINATING SET*, p-q-*THRESHOLD DOMINATING SET*, p-*EFFICIENT DOMINATING SET*, p-*CONNECTED DOMINATING SET*, p-*CONNECTED VERTEX COVER*, p-*CYCLE DOMINATION*, p-*DIRECTED DOMINATION*, p-$\mathcal{S}$-*COVERING*, p-*MINIMUM PARTITION INTO CLIQUES*, p-*EDGE CLIQUE COVER*, and p-s-*CYCLE TRANSVERSAL*, then Π_g admits a linear kernel.*

PROOF. The definitions of *p*-VERTEX COVER, *p*-FEEDBACK VERTEX SET, *p*-ALMOST OUTERPLANAR, *p*-DIAMOND HITTING SET, *p*-ALMOST-*t*-BOUNDED TREEWIDTH, and *p*-ALMOST-*t*-BOUNDED PATHWIDTH have been given in Section 8.2, and all of them are special cases of the *p*-$\mathcal{H}$-DELETION problem. They all have FII because of Lemma 8.4, and the quasi-coverability of Π_g follows from Lemma 8.3. We remark that not all of these problems are coverable.

p-EDGE DOMINATING SET asks whether a graph G contains a set F of at most k edges such that every other edge shares a common endpoint with some edge in F. The coverability of Π_g follows by the fact that the endpoints of the edges in F form a dominating set of G. Moreover, the *p*-EDGE DOMINATING SET problem can be easily expressed as a *p*-MIN-CMSO[ψ] problem (with edge quantification), and the proof of its strong monotonicity is similar to the one of Lemma 8.9. Therefore, it has FII as well. Using similar arguments, one can prove that if Π=MINIMUM-VERTEX FEEDBACK EDGE SET, given an undirected graph G and a positive integer k, the task is to find a spanning tree T of G in which at most k vertices have a degree smaller than in G, then Π_g is quasi-coverable (however, it is not coverable). Moreover, MINIMUM-VERTEX FEEDBACK EDGE SET has FII because it can be expressed as a *p*-MIN-CMSO[ψ] problem and can be proved to be strongly monotone with a proof that uses the ideas of Lemma 8.9.

p-DOMINATING SET, *p*-*r*-DOMINATING SET, *p*-*q*-THRESHOLD DOMINATING SET, and *p*-EFFICIENT DOMINATING SET are defined in Section 8.5. All of these problems are coverable and have FII because of Lemma 8.9. Notice that for the first three problems, the FII property follows by expressing them as *p*-MIN-CMSO[ψ] problems and proving that they are are strongly monotone. However, *p*-EFFICIENT DOMINATING SET is *not* strongly monotone, and the proof that it has FII uses a different idea.

p-CONNECTED DOMINATING SET is also defined in Section 8.5. The coverability of Π_g and the FII property is proved in Lemma 8.14. Using similar ideas, the same results can also be proved for CONNECTED VERTEX COVER.

The CYCLE DOMINATION problem asks whether a graph G contains a set S of at most k vertices such that the removal of S together with its neighbors from G results in an acyclic graph. This problem can be seen as a common extension of *p*-FEEDBACK VERTEX SET and *p*-DOMINATING SET. Π_g can be proven to be quasi-coverable with arguments similar to those in the case of *p*-FEEDBACK VERTEX SET (*p*-CYCLE DOMINATION is not a coverable problem). The problem is easily expressible as a *p*-MIN-CMSO[ψ] problem, and the proof that it is strongly monotone is a blend of the ideas of the proofs of Lemmata 8.4 and 8.9.

p-DIRECTED DOMINATION is defined in Section 8.7. The coverability and the FII property of Π_g are proved in Lemma 8.17.

p-$\mathcal{S}$-Covering has been defined in Section 8.4. The existence of a linear kernel for this problem makes use of the Redundant Vertex Rule (Lemma 8.7), Lemma 8.8 (for coverability), and the ideas in the proof of Lemma 8.4 (for the FII property).

The p-Minimum Partition Into Cliques problem asks whether the vertex set of a graph G can be partitioned into at most k sets, each inducing a clique in G (in other words, we are asking for a k coloring of the complement of G). Let S be a set containing a vertex from each clique. Notice that S is a dominating set of G. Therefore, Π_g is a coverable problem. To prove that it also has FII, one needs to express it as a p-min-CMSO[ψ] problem and then use arguments similar to those of Lemma 8.9 to prove that it is strongly monotone.

The p-Edge Clique Cover problem asks whether a graph G contains a collection of at most k cliques such that for every edge of G, both its endpoints belongs to some of those cliques. We observe first that Π_g is quasi-coverable. To see this, just notice that if we consider a set with one vertex from each such clique, then the removal of the closed neighborhood of this set from G results in an edgeless graph. The proof that the problem has FII is omitted in this article.

Finally, p-s-Cycle Transversal has been defined in Section 8.8. Although this problem is not strongly monotone, it has FII because of Lemma 8.18. To prove that it has a linear kernel, first one needs to apply to its instances the following preprocessing routine: remove each vertex that does not appear in some cycle of G of length $\leq s$. This routine can be seen as a special case of the Redundant Vertex Rule presented in Section 8.4, and with a proof similar to the one of Lemma 8.7, one can show that it produces equivalent instances. Under these circumstances, the coverability of Π_g can be proved following the arguments of Lemma 8.8. □

We continue with the consequences of Theorem 1.3 to maximization problems that have FII.

Corollary 8.20. *If $g \in \mathbb{Z}^+$ and if Π is one of the following problems,* p-r-Scattered Set, p-Independent Set, p-Induced Matching, p-Triangle Edge Packing, p-Maximum Internal Spanning Tree, p-Maximum Full-Degree Spanning Tree, p-Cycle Packing, p-$\mathcal{H}$-Packing, p-Triangle Vertex Packing, p-$\mathcal{S}$-Packing, *and* p-Edge Cycle Packing, *then Π_g admits a linear kernel.*

Proof. The p-r-Scattered Set problem has been defined in Section 8.6. The coverability of Π_g^r is proved in Lemma 8.15, whereas the problem has FII because of Lemma 8.16. We stress that the p-r-Scattered Set problem is, in general, not a strongly monotone problem. The p-Independent Set problem asks whether a graph G contains a set of at least k mutually nonadjacent vertices. If Π=p-Independent Set, then Π_g is coverable using an argument that is very similar to the one of Lemma 8.15. Likewise, one may use the arguments of Lemma 8.16 to prove that the problem has FII. Alternatively, one may express p-Independent Set as a p-max-CMSO[ψ] problem and then prove that it is strongly monotone.

The p-Induced Matching problem asks whether a graph G contains a set of at least k edges such that no vertex in G has as neighbors endpoints of more than one edges in this set. The problem is quasi-coverable because every NO-instance without isolated vertices has a $(1, 3)$-dominating of size at most k. Moreover, the FII property uses ideas of the proof of 8.16. We stress that p-Induced Matching is not a strongly monotone problem.

The p-Triangle Edge Packing problem asks whether a graph G contains at least k triangles such that no two of them have any edge in common. The existence of a linear kernel for this problem makes use of the Redundant Vertex Rule and is based on suitable adaptations of the proofs of Lemma 8.8 (for coverability) and Lemma 8.4 (for the FII property).

The p-MAXIMUM INTERNAL SPANNING TREE problem asks whether a graph G has a spanning tree with at least k internal vertices. The coverability of Π_g follows by observing that a NO-instance has a connected dominating set of less than k vertices. The problem is not strongly monotone, and proving that it has FII requires a direct proof that we omit in this article.

The p-MAXIMUM FULL-DEGREE SPANNING TREE problem asks whether a graph G has a spanning tree T containing at least k vertices of full degree (a vertex v of T has *full degree* if $N_T(v) = N_G(v)$). Clearly, a NO-instance of Π_g cannot have a 2-independent set of size at least k; otherwise, we can grow a spanning tree with $\geq k$ full-degree vertices by starting from the neighborhoods of the vertices in such a set. But then, using the arguments of the proof of Lemma 8.15, G has a dominating set of size $c \cdot k$, where c is a constant that depends on the Euler genus g of G. This implies the coverability of Π_g. For the FII property, we only mention that the problem is not strongly monotone, and a specialized proof is required that is omitted in this article.

The p-CYCLE PACKING problem asks whether a graph contains at least k mutually vertex-disjoint cycles. This is a special case of the p-$\mathcal{H}$-PACKING problem, where $\mathcal{H} = \{K_3\}$. For both problems, the quasi-coverability of Π_g follows from Lemma 8.5. The FII property of p-CYCLE PACKING follows from Lemma 8.6, and this proof can be extended for the general case of the p-$\mathcal{H}$-PACKING problem, as mentioned at the end of Section 8.3. Notice that both problems are neither strongly monotone nor coverable.

The p-TRIANGLE VERTEX PACKING problem asks whether a graph G contains a set of at least k triangles where no two such triangles share some common vertex. p-TRIANGLE VERTEX PACKING is a special case of the p-$\mathcal{S}$-PACKING problem, where $\mathcal{S} = \{K_3\}$. The existence of a linear kernel for these problem makes use of the Redundant Vertex Rule (Lemma 8.7), Lemma 8.8 (for coverability), and the ideas in the proof of Lemma 8.6 (for the FII property).

p-EDGE CYCLE PACKING asks whether a graph G contains a collection of at least k mutually edge-disjoint cycles. To prove the quasi-coverability of Π_g, observe that a NO-instance cannot contain a collection of k vertex-disjoint cycles. But then, by the application of Erdős-Pósa property on bounded genus graphs (e.g., see Fomin et al. [2011] and Kloks et al. [2002]), G contains a set of at most $c \cdot k$ vertices meeting all the cycles of G, where c is a constant depending on the Euler genus g of G. The proof that the problem has FII is omitted. □

Corollaries 8.19 and 8.20 unify and generalize results presented in Alber et al. [2004, 2006b], Bodlaender and Penninkx [2008], Bodlaender et al. [2008], Chen et al. [2007], Fomin and Thilikos [2004], Guo and Niedermeier [2007b], Guo et al. [2010], Kanj et al. [2011], Lokshtanov et al. [2011], Moser and Sikdar [2009], and Xia and Zhang [2011].

We conclude this section with some consequences of Theorem 1.1 for problems that do not have FII.

COROLLARY 8.21. *If $g \in \mathbb{Z}^+$ and if Π is one of the following problems,* p-INDEPENDENT DOMINATING SET, p-ACYCLIC DOMINATING SET, p-INDEPENDENT DIRECTED DOMINATION, p-MAXIMUM INTERNAL OUT-BRANCHING, p-ODD SET, *and* p-EDGE-$\mathcal{S}$-COVERING, *then Π_g admits a polynomial kernel.*

PROOF. The p-INDEPENDENT DOMINATING SET problem asks whether a graph G contains a dominating set of at most k mutually nonadjacent vertices. The p-ACYCLIC DOMINATING SET problem asks whether a graph G contains a dominating set S of at most k vertices such that $G[S]$ is acyclic. Although these problems do not have FII, they can be both expressed as p-MIN-CMSO[ψ] problems and are obviously coverable.

Problems p-Independent Directed Domination and p-Maximum Internal Out-branching have been defined in Section 8.7, and they do not have FII. According to Lemma 8.17, in both cases, Π_g is a coverable p-min-CMSO[ψ] problem.

The p-Odd Set problem asks whether a graph G contains a set S of at most k vertices such that for every vertex of G, the number of its neighbors in S is odd. Clearly, such a set is a dominating set, and therefore Π_g is coverable. p-Odd Set does not have FII. However, it can be expressed as a p-min-CMSO[ψ] problem (notice that here we have to use the "counting" expressive power of CMSO).

Given some fixed finite collection of graphs $\mathcal{S}$, the p-Edge-$\mathcal{S}$-Covering problem asks whether a graph G contains a set of at most k edges meeting every subgraph of G that is isomorphic to a graph in $\mathcal{S}$. For this problem, a linear kernel requires the application of the Redundant Vertex Rule. The coverability of Π_g follows similarly to the proof of Lemma 8.8. Edge-$\mathcal{S}$-Covering does not have, in general, FII (although it has FII if $\mathcal{S}$ contains only cliques). However, it is possible to formulate it as a p-min-CMSO[ψ] problem. □

Concluding this section, we mention that there are several problems that do not satisfy the conditions of Theorems 1.3 and 1.1.

Apart from the problems mentioned in Corollary 8.20, other examples of p-max-CMSO problems that do not have FII include p-Maximum Cut, p-Longest Path, and p-Longest Cycle (see de Fluiter [1997]). Notice that p-Maximum Cut is (trivially) quasi-coverable, whereas p-Longest Path and p-Longest Cycle are not. In fact, p-Maximum Cut admits a trivial $2k$ kernel on general graphs, whereas p-Longest Path, and p-Longest Cycle do not admit polynomial kernels unless coNP $\subseteq$ NP/poly [Bodlaender et al. 2009a].

As an example of a problem that has FII but is neither coverable or quasi-coverable, we mention p-Hamiltonian Path Completion (asking whether the addition of at most k edges in a graph can make it Hamiltonian). This problem can be expressed as a p-min-CMSO[ψ], and it is possible to prove that it is strongly monotone. Therefore, it has FII. However, none of our results apply to this problem, as it is not quasi-coverable. In fact, p-Hamiltonian Path Completion cannot have a kernel, unless P=NP, as such a kernelization algorithm, for $k = 1$, would be a polynomial algorithm for the Hamiltonian Path problem.

9. OPEN PROBLEMS AND FURTHER DIRECTIONS

This article gives the first meta-theorems on kernelization, where logical and combinatorial properties of problems lead to kernels of polynomial or linear sizes. Our results are quite general in the sense that they can be applied to a large number of combinatorial problems on graphs on fixed surfaces and generalize a large collection of known results. Still, there are several directions in which our results could possibly be extended. We conclude with some new problems and further research directions opened by our results.

Further extensions. The first natural question for further research is if our logical and combinatorial properties can be extended to larger classes of problems. The property that problems should satisfy some kind of coverability or quasi-coverability cannot be omitted. For instance, even though the problem of finding a path of length k is expressible in first-order logic, it does not admit a polynomial kernel on planar graphs unless coNP $\subseteq$ NP/poly [Bodlaender et al. 2009a]. An interesting question for further research is the following:

—Do all quasi-coverable CMSO problems admit a linear kernel on graphs of bounded genus?

This question is interesting even restricting ourselves to planar graphs.

It is very natural to ask whether our results can be extended to more general classes of graphs. The most natural candidates for such extensions are graphs of bounded local treewidth [Frick and Grohe 2001] and graphs of bounded expansion [Nešetřil and de Mendez 2008]. The first step in this direction is done in Fomin et al. [2010].

Practical considerations. Our meta-theorems provide simple criteria to decide whether a problem admits a polynomial or linear kernel on graphs of bounded genus. It is expected that for concrete problems, tailor-made kernels will have much smaller constant factors than what would follow from a direct application of our results. However, our approach might be useful for computer-aided design of kernelization algorithms: a computer program can in some cases output a set of rules that transform each protrusion to a minimum-size representative and estimate the obtained kernel size. This seems to be an interesting and far from trivial algorithm-engineering problem. In general, finding linear kernels with reasonably small constant factors for concrete problems on planar graphs or graphs with small genus remains a worthy topic of further research.

Some concrete open problems. We conclude with some concrete problems that cannot be resolved by our approach. These include p-Directed Feedback Vertex Set [Chen et al. 2008] and p-Odd Cycle Transversal [Reed et al. 2004], to name a few. These problems are expressible in CMSO, but none of them are known to be quasi-coverable. For p-Directed Feedback Vertex Set, no polynomial kernel is known even on planar graphs. For p-Odd Cycle Transversal, a randomized kernel for general graphs was obtained recently in Kratsch and Wahlström [2014], but the existence of a deterministic kernel even on planar graphs is open.

Impact. The protrusion replacement technique for kernelization was introduced in the preliminary conference version of this article [Bodlaender et al. 2009b] and appears to be useful in different algorithmic approaches. They were used to obtain kernels for a wide set of bidimensional problems on H-minor-free graphs [Fomin et al. 2010, 2012], vertex removal problems on general and unit disc graphs [Fomin et al. 2011a], and problems on graphs excluding a fixed graph as a topological minor [Fomin et al. 2013; Kim et al. 2016]. It was also used in the design of fast parameterized algorithms and approximation algorithms [Fomin et al. 2011b, 2012a, 2012b; Joret et al. 2014; Kim et al. 2015, 2016].

APPENDIX

A. PROBLEM COMPENDIUM

In this compendium, we present the kernelization status of all problems that have been mentioned in this article.

A.1. Minimization Problems That Have FII and Are Quasi-Coverable—*Linear Kernels for Graphs of Bounded Genus*.

p-Vertex Cover, p-Feedback Vertex Set, p-Almost Outerplanar, p-Diamond Hitting Set, p-Almost-t-bounded treewidth, p-Almost-t-bounded pathwidth, p-$\mathcal{H}$-Deletion, p-Edge Dominating Set, p-Minimum-Vertex Feedback Edge Set, p-Dominating Set, p-r-Dominating Set, p-q-Threshold Dominating Set, p-Efficient Dominating Set*, p-Connected Dominating Set, p-Connected Vertex Cover, p-Cycle Domination, p-Directed Domination, p-$\mathcal{S}$-Covering, p-Minimum Partition Into Cliques, p-Edge Clique Cover*, and p-s-Cycle Transversal*.

A.2. Maximization Problems That Have FII and Are Quasi-coverable—*Linear Kernels for Graphs of Bounded Genus*.

p-r-Scattered Set*, p-Independent Set, p-Induced Matching*, p-Triangle Edge Packing+, p-Maximum Internal Spanning Tree*, p-Maximum Full-Degree Spanning Tree*, p-Cycle Packing*, p-$\mathcal{H}$-Packing*, p-Triangle Vertex Packing+, p-$\mathcal{S}$-Packing+, and p-Edge Cycle Packing*,

For all problems with an asterisk "*", a direct proof that they have FII is required. For the rest, FII property follow by expressing them as a p-min/max-CMSO problem and proving strong monotonicity. For the problems with a cross "+", the linear kernel assumes the application of some preprocessing routine.

A.3. Problems That Do Not Have FII and Are Coverable *p*-min/max-CMSO—*Polynomial Kernels for Graphs of Bounded Genus*.

p-Independent Dominating Set, p-Acyclic Dominating Set, p-Independent Directed Domination, p-Maximum Internal Out-branching, p-Odd Set, and p-Edge-$\mathcal{S}$-Covering.

A.4. A Problem That Has FII but Is Not Quasi-Coverable.

p-Hamiltonian Path Completion.

A.5. A Quasi-Coverable Problem That Has No FII.

p-Maximum Cut.

A.6. Problems That Do Not Have FII and Are Not Quasi-Coverable.

p-Longest Path and p-Longest Cycle.

ACKNOWLEDGMENT

We thank Jiong Guo, Ge Xia, and Yong Zhang for sending us the full versions of Guo and Niedermeier [2007b] and Xia and Zhang [2011]. We also thank the anonymous reviewers of FOCS '09 and JACM for their valuable comments on previous versions of this article.

REFERENCES

Karl Abrahamson and Michael Fellows. 1993. Finite automata, bounded treewidth and well-quasiordering. In *Proceedings of the AMS Summer Workshop on Graph Minors, Graph Structure Theory*. 539–563. DOI:http://dx.doi.org/10.1090/conm/147/01199

Isolde Adler, Martin Grohe, and Stephan Kreutzer. 2008. Computing excluded minors. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*. 641–650. http://portal.acm.org/citation.cfm?id=1347082.1347153

Jochen Alber, Nadja Betzler, and Rolf Niedermeier. 2006a. Experiments on data reduction for optimal domination in networks. *Annals of Operations Research* 146, 1, 105–117.

Jochen Alber, Britta Dorn, and Rolf Niedermeier. 2006b. A general data reduction scheme for domination in graphs. In *SOFSEM 2006: Theory and Practice of Computer Science*. Lecture Notes in Computer Science, Vol. 3831. Springer, Berlin, 137–147.

Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. 2004. Polynomial-time data reduction for dominating sets. *Journal of the ACM* 51, 363–384.

Noga Alon and Shai Gutner. 2008. *Kernels for the Dominating Set Problem on Graphs With an Excluded Minor*. Technical Report TR08-066. Electronic Colloquium on Computational Complexity.

Stefan Arnborg, Bruno Courcelle, Andrzej Proskurowski, and Detlef Seese. 1993. An algebraic theory of graph reduction. *Journal of the ACM* 40, 1134–1164.

Stefan Arnborg, Jens Lagergren, and Detlef Seese. 1991. Easy problems for tree-decomposable graphs. *Journal of Algorithms* 12, 308–340.

D. W. Bange, A. E. Barkauskas, and P. J. Slater. 1988. Efficient dominating sets in graphs. In *Applications of Discrete Mathematics*. SIAM, Philadelphia, PA, 189–199.

Hans L. Bodlaender. 1996. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing* 25, 6, 1305–1317.

Hans L. Bodlaender and Babette de Fluiter. 1996. Reduction algorithms for constructing solutions in graphs with small treewidth. In *Computing and Combinatorics*. Lecture Notes in Computer Science, Vol. 1090. Springer, 199–208.

Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. 2009a. On problems without polynomial kernels. *Journal of Computer and System Sciences* 75, 8, 423–434.

Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. 2009b. (Meta) kernelization. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS'09)*. IEEE, Los Alamitos, CA, 629–638.

Hans L. Bodlaender and Torben Hagerup. 1998. Parallel algorithms with optimal speedup for bounded treewidth. *SIAM Journal on Computing* 27, 1725–1746.

Hans L. Bodlaender and Eelko Penninkx. 2008. A linear kernel for planar feedback vertex set. In *Parameterized and Exact Computation*. Lecture Notes in Computer Science, Vol. 5018. Springer, 160–171. http://dl.acm.org/citation.cfm?id=1789694.1789710

Hans L. Bodlaender, Eelko Penninkx, and Richard B. Tan. 2008. A linear kernel for the k-disjoint cycle problem on planar graphs. In *Algorithms and Computation*. Lecture Notes in Computer Science, Vol. 5369. Springer, Berlin, 306–317.

Hans L. Bodlaender and Babette van Antwerpen-de Fluiter. 2001. Reduction algorithms for graphs of small treewidth. *Information and Computation* 167, 86–119.

Richard B. Borie, R. Gary Parker, and Craig A. Tovey. 1992. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica* 7, 555–581.

Jianer Chen, Henning Fernau, Iyad A. Kanj, and Ge Xia. 2007. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM Journal on Computing* 37, 1077–1106.

Jianer Chen, Iyad A. Kanj, and Weijia Jia. 2001. Vertex cover: Further observations and further improvements. *Journal of Algorithms* 41, 2, 280–301. DOI:http://dx.doi.org/10.1006/jagm.2001.1186

Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. 2008. A fixed-parameter algorithm for the directed feedback vertex set problem. *Journal of the ACM* 55, 5, Article No. 21. DOI:http://dx.doi.org/10.1145/1411509.1411511

Bruno Courcelle. 1990. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation* 85, 12–75.

Bruno Courcelle. 1992. The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues. *RAIRO: Theoretical Informatics and Applications* 26, 3, 257–286.

Bruno Courcelle. 1997. The expression of graph properties and graph transformations in monadic second-order logic. In *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 1*. World Scientific Publishing, River Edge, NJ, 313–400. DOI:http://dx.doi.org/10.1142/9789812384720_0005

Bruno Courcelle and Joost Engelfriet. 2012. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge University Press.

Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.

Anuj Dawar, Martin Grohe, and Stephan Kreutzer. 2007. Locally excluding a minor. In *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science (LICS'07)*. IEEE, Los Alamitos, CA, 270–279.

Babette de Fluiter. 1997. *Algorithms for Graphs of Small Treewidth*. Ph.D. Dissertation. Utrecht University, Utrecht, Netherlands.

Rodney G. Downey and Michael R. Fellows. 1998. *Parameterized Complexity*. Springer, Berlin, Germany.

Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer.

P. Duchet and H. Meyniel. 1982. On Hadwiger's number and the stability number. In *Graph Theory*. North-Holland Mathematics Studies, Vol. 62. North-Holland, Amsterdam, Netherlands, 71–73.

Zdenek Dvorak. 2013. Constant-factor approximation of the domination number in sparse graphs. *European Journal of Combinatorics* 34, 5, 833–840.

David Eppstein. 2000. Diameter and treewidth in minor-closed graph families. *Algorithmica* 27, 275–291.

Michael R. Fellows and Michael A. Langston. 1989. An analogue of the Myhill-Nerode theorem and its use in computing finite-basis characterizations (extended abstract). In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS'89)*. IEEE, Los Alamitos, CA, 520–525.

Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer-Verlag, Berlin, Germany.

Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. 2011a. Hitting forbidden minors: Approximation and kernelization. In *Proceedings of the 8th International Symposium on Theoretical Aspects of Computer Science (STACS'11)*. 189–200.

Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. 2012b. Planar f-deletion: Approximation, kernelization and optimal FPT algorithms. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS'12)*. IEEE, Los Alamitos, CA, 470–479.

Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. 2011b. Bidimensionality and EPTAS. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'11)*. 748–759.

Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. 2012a. Bidimensionality and geometric graphs. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'12)*. 1563–1575.

F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. 2010. Bidimensionality and kernels. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*. ACM, New York, NY, 503–510.

Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. 2012. Linear kernels for (connected) dominating set on H-minor-free graphs. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'12)*. 82–93.

Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. 2013. Linear kernels for (connected) dominating set on graphs with excluded topological subgraphs. In *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS'13)*. 92–103. DOI:http://dx.doi.org/10.4230/LIPIcs.STACS.2013.92

Fedor V. Fomin and Saket Saurabh. 2014. Kernelization methods for fixed-parameter tractability. In *Tractability*. Cambridge University Press, Cambridge, MA, 260–282.

Fedor V. Fomin, Saket Saurabh, and Dimitrios M. Thilikos. 2011. Strengthening Erdős-Pósa property for minor-closed graph classes. *Journal of Graph Theory* 66, 3, 235–240.

Fedor V. Fomin and Dimitrios M. Thilikos. 2004. Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up. In *Automata, Languages and Programming*. Lecture Notes in Computer Science, Vol. 3142. Springer, Berlin, 581–592.

Markus Frick and Martin Grohe. 2001. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM* 48, 6, 1184–1206.

Martin Grohe. 2007. Logic, graphs, and algorithms. In *Logic and Automata-History and Perspectives*, J. Flum, E. Gradel, and T. Wilke (Eds.). Amsterdam University Press, Amsterdam, Netherlands, 357–422.

Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. 2011. Finding topological subgraphs is fixed-parameter tractable. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC'11)*. ACM, New York, NY, 479–488.

Jiong Guo and Rolf Niedermeier. 2007a. Invitation to data reduction and problem kernelization. *ACM SIGACT News* 38, 1, 31–45.

Jiong Guo and Rolf Niedermeier. 2007b. Linear problem kernels for NP-hard problems on planar graphs. In *Automata, Languages and Programming*. Lecture Notes in Computer Science, Vol. 4596. Springer, 375–386.

Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. 2010. Fixed-parameter tractability results for full-degree spanning tree and its dual. *Networks* 56, 2, 116–130.

Gregory Gutin, Ton Kloks, Chuan Min Lee, and Anders Yeo. 2005. Kernels in planar digraphs. *Journal of Computer and System Sciences* 71, 2, 174–184.

Gregory Gutin, Igor Razgon, and Eun Jung Kim. 2009. Minimum leaf out-branching and related problems. *Theoretical Computer Science* 410, 45, 4571–4579.

Gwenaël Joret, Christophe Paul, Ignasi Sau, Saket Saurabh, and Stéphan Thomassé. 2014. Hitting and harvesting pumpkins. *SIAM Journal on Discrete Mathematics* 28, 3, 1363–1390. DOI:http://dx.doi.org/10.1137/120883736

M. Juvan, A. Malnič, and B. Mohar. 1996. Systems of curves on surfaces. *Journal of Combinatorial Theory, Series B* 68, 1, 7–22. DOI:http://dx.doi.org/10.1006/J.~Combin. Theory Ser. B.1996.0053

Marcin Kaminski and Dimitrios M. Thilikos. 2012. Contraction checking in graphs on surfaces. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS'12)*. 182–193. DOI:http://dx.doi.org/10.4230/LIPIcs.STACS.2012.182

Iyad A. Kanj, Michael J. Pelsmajer, Marcus Schaefer, and Ge Xia. 2011. On the induced matching problem. *Journal of Computer and System Sciences* 77, 6, 1058–1070.

Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. 2016. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Transactions on Algorithms* 12, 2, 21. DOI:http://dx.doi.org/10.1145/2797140

Eun Jung Kim, Christophe Paul, and Geevarghese Philip. 2015. A single-exponential FPT algorithm for the K_4-minor cover problem. *Journal of Computer and System Sciences* 81, 1, 186–207. DOI:http://dx.doi.org/10.1016/j.jcss.2014.05.001

T. Kloks, C. M. Lee, and J. Liu. 2002. New algorithms for k-face cover, k-feedback vertex set, and k-disjoint cycles on plane and planar graphs. In *Graph-Theoretic Concepts in Computer Science*. Lecture Notes in Computer Science, Vol. 2573. Springer, 282–295.

Stefan Kratsch and Magnus Wahlström. 2014. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Transactions on Algorithms* 10, 4, 20:1–20:15. DOI:http://dx.doi.org/10.1145/2635810

Stephan Kreutzer. 2011. Algorithmic meta-theorems. In *Finite and Algorithmic Model Theory*. London Mathematical Society Lecture Notes Series, Vol. 379. Cambridge University Press, Cambridge, England, 177–270.

Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. 2012. Kernelization–preprocessing with a guarantee. In *The Multivariate Algorithmic Revolution and Beyond*. Springer, 129–161.

Daniel Lokshtanov, Matthias Mnich, and Saket Saurabh. 2011. A linear kernel for a planar connected dominating set. *Theoretical Computer Science* 412, 23, 2536–2543.

Bojan Mohar. 1999. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM Journal on Discrete Mathematics* 12, 1, 6–26.

Bojan Mohar and Carsten Thomassen. 2001. *Graphs on Surfaces*. Johns Hopkins University Press, Baltimore, MD.

Hannes Moser and Somnath Sikdar. 2009. The parameterized complexity of the induced matching problem. *Discrete Applied Mathematics* 157, 4, 715–727.

Jaroslav Nešetřil and Patrice Ossona de Mendez. 2008. Grad and classes with bounded expansion II. Algorithmic aspects. *European Journal of Combinatorics* 29, 3, 777–791.

Rolf Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications, Vol. 31. Oxford University Press, Oxford, England.

Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. 2012. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Transactions on Algorithms* 9, 1, 11.

W. V. Quine. 1952. The problem of simplifying truth functions. *American Mathematical Monthly* 59, 521–531.

Bruce Reed, Kaleigh Smith, and Adrian Vetta. 2004. Finding odd cycle transversals. *Operations Research Letters* 32, 4, 299–301. DOI:http://dx.doi.org/10.1016/j.orl.2003.10.009

N. Robertson and P. D. Seymour. 1995. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B* 63, 1, 65–110.

Neil Robertson, Paul D. Seymour, and Robin Thomas. 1994. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 323–348.

Paul D. Seymour and Robin Thomas. 1993. Graph searching and a minimax theorem for tree-width. *Journal of Combinatorial Theory, Series B* 58, 239–257.

Stéphan Thomassé. 2010. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms* 6, 2, 32:1–32.8.

Johan M. M. van Rooij. 2011. *Exact Exponential-Time Algorithms for Domination Problems in Graphs*. Ph.D. Dissertation. Utrecht University, Utrecht, Netherlands.

Ge Xia and Yong Zhang. 2011. On the small cycle transversal of planar graphs. *Theoretical Computer Science* 412, 29, 3501–3509.

Received May 2010; revised March 2016; accepted July 2016