Twin-width I: tractable FO model checking

Édouard Bonnet
Univ Lyon, CNRS, ENS de Lyon,
Université Claude Bernard Lyon 1, LIP UMR5668,
Lyon, France
Email: edouard.bonnet@ens-lyon.fr

Eun Jung Kim
Université Paris-Dauphine, PSL University,
CNRS UMR7243, LAMSADE,
Paris, France
Email: eun-jung.kim@dauphine.fr

Stéphan Thomassé
Univ Lyon, CNRS, ENS de Lyon,
Université Claude Bernard Lyon 1, LIP UMR5668,
Lyon, France
Institut Universitaire de France
Email: stephan.thomasse@ens-lyon.fr

Rémi Watrigant
Univ Lyon, CNRS, ENS de Lyon,
Université Claude Bernard Lyon 1, LIP UMR5668,
Lyon, France
Email: remi.watrigant@ens-lyon.fr

***Abstract*—Inspired by a *width* invariant defined on permutations by Guillemot and Marx [SODA '14], we introduce the notion of twin-width on graphs and on matrices. Proper minor-closed classes, bounded rank-width graphs, map graphs, K_t-free unit d-dimensional ball graphs, posets with antichains of bounded size, and proper subclasses of dimension-2 posets all have bounded twin-width. On all these classes (except map graphs without geometric embedding) we show how to compute in polynomial time a *sequence of d-contractions*, witness that the twin-width is at most d. We show that FO model checking, that is deciding if a given first-order formula ϕ evaluates to true for a given binary structure G on a domain D, is FPT in $|\phi|$ on classes of bounded twin-width, provided the witness is given. More precisely, being given a d-contraction sequence for G, our algorithm runs in time $f(d, |\phi|) \cdot |D|$ where f is a computable but non-elementary function. We also prove that bounded twin-width is preserved by FO interpretations and transductions (allowing operations such as squaring or complementing a graph). This unifies and significantly extends the knowledge on fixed-parameter tractability of FO model checking on non-monotone classes, such as the FPT algorithm on bounded-width posets by Gajarský et al. [FOCS '15].**

***Keywords*-Twin-width; FO model checking; fixed-parameter tractability;**

I. Introduction

Measuring how complex a class of structures is often depends on the context. Complexity can be related to algorithms (are computations easier on the class?), counting (how many structures exist per slice of the class?), size (can structures be encoded in a compact way?), decomposition (can structures be built with easy operations?), and so on. The most successful and central complexity invariants like treewidth and VC-dimension tick many of these boxes and, as such, stand as cornerstone notions in both discrete mathematics and computer science.

In 2014, Guillemot and Marx [1] solved a long-standing question by showing that detecting a fixed pattern in some input permutation can be done in linear time. This result came as a surprise: Many researchers thought the problem was W[1]-hard since all known techniques had failed so far. In their paper, Guillemot and Marx observed that their proof introduces a parameter and a dynamic programming scheme of a new kind and wondered whether a graph-theoretic generalization of their permutation parameter could exist.

The starting point of our paper is to answer that question positively, by generalizing their width parameter to graphs and even matrices. This new notion, dubbed *twin-width*, proves remarkably well connected to other areas of computer science, logic, and combinatorics. In the course of the paper, we show that graphs of bounded twin-width define a very natural class with respect to computational complexity (FO model checking is linear), to model theory (they are stable under first-order interpretations), and to decomposition methods (as a generalization of both proper minor-closed and bounded rank-width/clique-width classes).

A. A dynamic generalization of cographs

When it comes to graph decompositions, arguably one of the simplest graph classes is the class of *cographs*. Starting from a single vertex, cographs can be built by iterating disjoint unions and complete sums. Another way to decompose cographs is to observe that they always contain *twins*, that is two vertices u and v with the same neighborhood outside $\{u, v\}$ (hence contracting u, v is equivalent to deleting u).

This work was supported by the grant from French National Agency under PRC program (project Digraphs, ANR-19-CE48-0013-01).

Eun Jung Kim is supported by the grant from French National Agency under JCJC program (project ASSK, ANR-18-CE40-0025-01).

This work was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX- 0007) operated by the French National Research Agency (ANR)

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00062

Cographs are then exactly graphs which can be contracted to a single vertex by iterating contractions of twins. Generalizing the decomposition by allowing more complex bipartitions provides the celebrated notions of clique-width and rank-width, which extends treewidth to dense graphs. However, bounded rank-width do not capture simple graphs such as unit interval graphs which have a simple linear structure, and allow polynomial-time algorithms for various problems. Also, bounded rank-width does not capture large 2-dimensional grids, on which we know how to design FPT algorithms.

The goal of this paper is to propose a width parameter which is not only bounded on d-dimensional grids, proper minor-closed classes and bounded rank-width graphs, but also provides a very versatile and simple scheme which can be applied to many structures, for instance, patterns of permutations, hypergraphs, and posets. The idea is very simple: a graph has bounded twin-width if it can be iteratively contracted to a singleton, where each contracted pair consists of near-twins (two vertices whose neighborhoods differ only on a bounded number of elements). The crucial ingredient to add to this simplified picture is to keep track of the errors with another type of edges, that we call *red edges*, and to require that the degree in red edges remains bounded by a threshold, say d.

In a nutshell (a more formal definition will be given in Section III), we consider a sequence of graphs $G_n, G_{n-1}, \ldots, G_2, G_1$, where G_n is the original graph G, G_1 is the one-vertex graph, G_i has i vertices, and G_{i-1} is obtained from G_i by performing a single contraction of two (non-necessarily adjacent) vertices. For every vertex $u \in V(G_i)$, let us denote by $u(G)$ the vertices of G which have been contracted to u along the sequence $G_n, \ldots, G_i$. The red edges mentioned previously consist of all pairs uv of vertices of G_i such that $u(G)$ and $v(G)$ are not homogeneous[1] in G. If the red degree of every G_i is at most d, then $G_n, G_{n-1}, \ldots, G_2, G_1$ is called a *sequence of d-contractions*, or *d-sequence*. The twin-width of G is the minimum d for which there exists a sequence of d-contractions. Hence, graphs of twin-width 0 are exactly the cographs (since a red edge never appears along the sequence when contracting twins). See Figure 1 for an illustration of a 2-sequence.

This basic definition proves to be extremely rich. The main algorithmic application presented in this paper is the design of a linear-time FPT algorithm for FO model checking on binary structures with bounded twin-width, provided a sequence of d-contractions is given.

B. FO model checking

A natural algorithmic question given a graph class $\mathcal{C}$ (i.e., a set of graphs closed under taking induced subgraphs) is

[1] Two disjoint sets of vertices are homogeneous if, between them, there are either all possible edges or no edge at all.

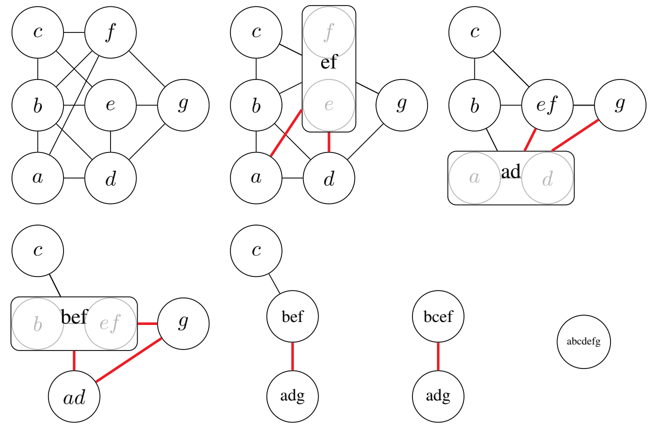

Figure 1. A 2-sequence of contractions to a single vertex shows that the original graph has twin-width at most 2.

whether or not deciding first-order formulas φ on graphs $G \in \mathcal{C}$ can be done in time whose superpolynomial blow-up is a function of $|\varphi|$ and $\mathcal{C}$ only. A line of works spanning two decades settled this question for monotone (that is, closed under taking subgraphs) graph classes. It was shown that one can decide first-order (FO) formulas in fixed-parameter time (FPT) in the formula size on bounded-degree graphs [2], planar graphs, and more generally, graphs with locally bounded treewidth [3], H-minor free graphs [4], locally H-minor free graphs [5], classes with (locally) bounded expansion [6], and finally nowhere dense classes [7]. The latter result generalizes all previous ones, since nowhere dense graphs contain all the aforementioned classes. Let us observe that the dependency on $|V(G)|$ of the FPT model checking algorithm on classes with bounded expansion is linear [6], while it is almost linear (i.e., $|V(G)|^{1+\varepsilon}$ for every $\varepsilon > 0$) for nowhere dense classes [7]. In sharp contrast, if a monotone class $\mathcal{C}$ is not nowhere dense then FO model checking on $\mathcal{C}$ is AW[$*$]-complete [8], hence highly unlikely to be FPT. Thus the result of Grohe et al. [7] gives a final answer in the case of monotone classes.

Since then, the focus has shifted to the complexity of model checking on (dense) non-monotone graph classes. Our main result is that FO model checking is FPT on classes with bounded twin-width. More precisely, we show that:

Theorem 1: Given an n-vertex (di)graph G, a sequence of d-contractions $G = G_n, G_{n-1}, \ldots, G_1 = K_1$, and a first-order formula φ, we can decide $G \models \varphi$ in time $f(|\varphi|, d) \cdot n$ for some computable, yet non-elementary, function f.

This unifies and extends known FPT algorithms for

- H-minor free graphs [4],
- posets of bounded width (i.e., size of the largest antichain) [9],

- permutations avoiding a fixed pattern [1][2] and proper subclass of permutation graphs,
- bounded rank-width or bounded clique-width [10],[3]

since we will establish that these classes have bounded twin-width, and that, on them, a sequence of d-contractions can be found efficiently. By transitivity, this also generalizes the FPT algorithm for L-interval graphs [11], and may shed a new unified light on geometric graph classes for which FO model checking is FPT [12]. In that direction we show that a large class of geometric intersection graphs with bounded clique number, including K_t-free unit d-dimensional ball graphs, admits such an algorithm. We also show that map graphs have bounded twin-width but we only provide a d-contraction sequence when the input comes with a planar embedding of the map. FO model checking was proven FPT on map graphs even when no geometric embedding is provided [13]. See Figure 2 for the Hasse diagram of classes with a fixed-parameter tractable FO model checking.

Permutation patterns can be represented as posets of dimension 2. Then any proper (hereditary) subclass of posets of dimension 2 contains all permutations avoiding a fixed pattern. Posets can in turn be encoded by directed graphs (or digraphs). Thus we formulated Theorem 1 with graphs and digraphs, to cover all the classes of bounded twin-width listed after the theorem. Twin-width and the applicability of Theorem 1 is actually broader: one may replace "an n-vertex (di)graph G" by "a binary structure G on a domain of size n" in the statement of the theorem, where a binary structure is a finite set of binary relations.

Roadmap for the proof of Theorem 1.: Instead of deciding "$G \models \varphi$" for a specific formula φ, we build in FPT time a tree $MT'_\ell(G)$ which contains enough information to answer all the queries of the form *is ϕ true on G?*, for every prenex sentence ϕ on ℓ variables. A prenex sentence ϕ starts with a quantification (existential and universal) over the ℓ variables, followed, in the case of graphs, by a Boolean combination $\phi'(x_1, \dots, x_\ell)$ of atoms of the form $x = y$ (interpreted as: vertex x is vertex y) and $E(x, y)$ (interpreted as: there is an edge between x and y). A simple but important insight is that once Existential and Universal players have chosen the assignment $v_1, \dots, v_\ell$, the truth of $\phi'(v_1, \dots, v_\ell)$ only depends on the induced subgraph $G[\{v_1, \dots, v_\ell\}]$ and the pattern of equality classes of the tuple $(v_1, \dots, v_\ell)$. Indeed the latter pair carries the truth value of each possible atom.

Imagine now the complete tree $MT_\ell(G)$ of all the possible "moves" assigning vertex v_i to variable x_i. This tree, called *morphism-tree*, has arity $|V(G)|$ and depth ℓ. Thus $MT_\ell(G)$ is too large to be explicitly computed. However, up to labeling its different levels with $\exists$ and $\forall$, it trivially contains what is needed to evaluate any ℓ-variable prenex formula on G. In light of the previous paragraph, $MT_\ell(G)$ contains way too much information. Assume, for instance, that two of its leaves v_ℓ, v'_ℓ with the same parent node define the same induced subgraph $G[\{v_1, \dots, v_{\ell-1}, v_\ell\}] \cong G[\{v_1, \dots, v_{\ell-1}, v'_\ell\}]$ and the same pattern of equality classes. Then it is safe to delete the "move v'_ℓ" from the possibilities of whichever player shall play at level ℓ. Indeed "move v_ℓ" is perfectly equivalent: As it sets to true the same list of atoms, it will satisfy the exact same formulas ϕ', irrelevant of the nature of the quantifier preceding x_ℓ. We generalize this notion to any pair of sibling nodes at any level of the morphism-tree, and we call *reduction* a morphism-tree obtained after removing equivalent sibling nodes (and their subtree). It can be observed that a *reduct*, that is, a reduction that cannot be reduced further, has size bounded by ℓ only. Thus it all boils down to computing a reduct $MT'_\ell(G)$ in FPT time.

Now the contraction sequence comes in. Actually, more convenient here than the successive graphs $G = G_n, G_{n-1}, \dots, G_1$, we consider the equivalent partition sequence: $\mathcal{P}_n, \mathcal{P}_{n-1}, \dots, \mathcal{P}_1$, where $\mathcal{P}_i$ is the partition of $V(G)$ whose parts correspond to the vertices of $V(G_i)$ ($v(G) \in \mathcal{P}_i$ is the set of all the vertices of $V(G)$ contracted to form $v \in V(G_i)$). Recall that two parts of $\mathcal{P}_i$ are *homogeneous* if they are fully adjacent or fully non-adjacent in G. Let $G_{\mathcal{P}_i}$ be the graph whose vertices are the parts of $\mathcal{P}_i$, and edges link every pair of non-homogeneous parts. It corresponds to the red edges of G_i. We also extend morphism-trees to partitioned graphs: $MT_\ell(G, \mathcal{P}_i)$ denotes the morphism-tree $MT_\ell(G)$ where reductions are only allowed between two vertices of the same part of $\mathcal{P}_i$. And for $X \in \mathcal{P}_i$, $MT_\ell(G, \mathcal{P}_i, X)$ is the morphism-tree $MT_\ell(G, \mathcal{P}_i)$ restricted to parts of $\mathcal{P}_i$ in the "vicinity" of X. Again $MT'_\ell(G, \mathcal{P}_i, X)$ denotes the reduct of $MT_\ell(G, \mathcal{P}_i, X)$.

By dynamic programming, we will maintain for i going from n down to 1, reducts $MT'_\ell(G, \mathcal{P}_i, X_j)$ for every $X_j \in \mathcal{P}_i$. $\mathcal{P}_n$ is a partition into singletons $\{v\}$ (for each $v \in V(G)$), so we initialize the reducts to paths of length ℓ labeled by v. $\mathcal{P}_1$ is the trivial partition $\{V(G)\}$, so the eventually computed reduct $MT'_\ell(G, \{V(G)\}, V(G))$ is exactly the reduct $MT'_\ell(G)$ that we were looking for. Say that, to go from $\mathcal{P}_{i+1}$ to $\mathcal{P}_i$, we fuse two sets X'_i, X''_i into X_i. We shall now update the reducts $MT'_\ell(G, \mathcal{P}_i, X_j)$ for every $X_j \in \mathcal{P}_i$, being given the reducts $MT'_\ell(G, \mathcal{P}_{i+1}, X_j)$ for every $X_j \in \mathcal{P}_{i+1}$. For parts X_j at distance more than 3^ℓ of X_i in $G_{\mathcal{P}_i}$, nothing happens: we set $MT'_\ell(G, \mathcal{P}_i, X_j) := MT'_\ell(G, \mathcal{P}_{i+1}, X_j)$. The value 3^ℓ is chosen so that two parts Y, Y' further apart than this threshold cannot "interact" via non-homogeneous pairs of parts. This implies that the choice of a precise vertex in Y does not affect in any way the choice of a precise vertex in Y'.

[2]Guillemot and Marx show that PERMUTATION PATTERN (not FO model checking in general) is FPT when the host permutation avoids a pattern, then a win-win argument proper to PERMUTATION PATTERN allows them to achieve an FPT algorithm for the class of *all* permutations.

[3]for this class, even deciding MSO_1 is FPT, which is something that we do not capture.

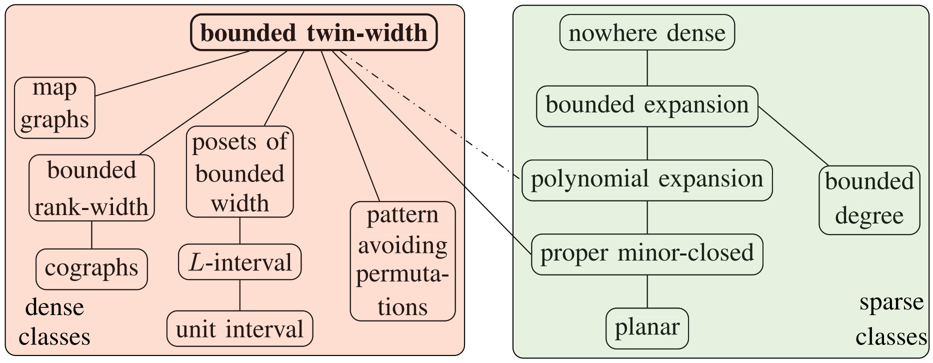

Figure 2. Hasse diagram of classes on which FO model checking is FPT, with the newcomer twin-width. The dash-dotted edge means that polynomial expansion may well be included in bounded twin-width. Bounded twin-width and nowhere dense classes roughly subsume all the current knowledge on the fixed-parameter tractability of FO model checking. Do they admit a natural common superclass still admitting an FPT algorithm for FO model checking?

We therefore focus on the at most $d^{3^\ell+1}$ parts of X_j at distance at most 3^ℓ of X_i in $G_{\mathcal{P}_i}$. We first combine, by a so-called *shuffle* operation, a bounded number of $MT'_\ell(G, \mathcal{P}_{i+1}, Y)$ for $Y \in \mathcal{P}_{i+1}$ sufficiently close to X_i in $G_{\mathcal{P}_i}$, then strategically prune redundant nodes, and reduce further the obtained morphism-tree (T, m). The aggregation of the two former steps is dubbed *pruned shuffle* and is the central operation of our algorithm. To define $MT'_\ell(G, \mathcal{P}_i, X_j)$ we finally *project* (or prune further) (T, m) on the nodes that are inherently *rooted at* X_j. To be formalized the latter requires to introduce an auxiliary graph, called *tuple graph*, and a notion of *local root*. These objects are instrumental in handling overlap or redundant information.

A crucial aspect of the algorithm relies on the following fact. If two connected components, say X and Y, of $G_{\mathcal{P}_{i+1}}$ are united in $G_{\mathcal{P}_i}$, then reductions of morphism-trees on $X' \cup Y'$ with $X' \subseteq X$ and $Y' \subseteq Y$ are obtained by just interleaving (actually *shuffling*) $MT'_\ell(G, \mathcal{P}_{i+1}, X')$ and $MT'_\ell(G, \mathcal{P}_{i+1}, Y')$. Indeed X' and Y' are by construction homogeneous to each other, so the precise choices of vertices in X' and in Y' are totally independent. We can finally observe that at each step i, we are updating a bounded number of reducts of bounded size. Therefore the overall algorithm takes linear FPT time (see bottom part of Figure 3).

C. How to compute the contraction sequences?

Given an arbitrary graph or binary structure, it seems tremendously hard to compute a good –let alone, optimum– contraction sequence. Fortunately on classes with bounded twin-width, for which this endeavor is algorithmically useful (in light of Theorem 1), we can often exploit structural properties of the class to achieve our goal. In the full version [14, Section 4] we present a simple polynomial-time algorithm outputting a $(2^{k+1}-1)$-contraction sequence on graphs of boolean-width at most k and a linear-time algorithm for a $3d$-contraction sequence of (subgraphs of) the d-dimensional grid of side-length n. The bottleneck for the former algorithm would lie in finding the boolean-width decomposition in the first place. The latter result enables to find in polynomial time $3^d k$-contraction sequences for unit d-dimensional ball graphs with clique number k, provided the geometric representation is given.

For other classes, such as planar graphs, directly finding the sequence proves challenging. Therefore we design in Section IV a framework that reduces this task to finding an ordering σ –later called *mixed-free order*– of the n vertices such that the adjacency matrix A written compliantly to σ is simple. Here by "simple" we mean that A cannot be divided into a large number of blocks of consecutive rows and columns, such that no cell of the division is horizontal (repetition of the same row subvector) or vertical (repetition of the same column subvector). An important local object to handle this type of division is the notion of *corner*, namely a consecutive 2-by-2 submatrix which is neither horizontal nor vertical. The principal ingredient to show that simple matrices have bounded twin-width is the use of a theorem by Marcus and Tardos [15] which states that $n \times n$ 0,1-matrices with at least cn 1 entries (for a large enough constant c) admit large divisions with at least one 1 entry in each cell. This result is at the core of Guillemot and Marx's algorithm [1] to solve PERMUTATION PATTERN in linear FPT time. As we now apply Marcus-Tardos theorem to the corners (and not the 1 entries), we bring this engine to the dense setting. Indeed the matrix can be packed with 1 entries, and yet we learn something non-trivial from the number of corners.

By Marcus-Tardos theorem the number of corners cannot be too large, otherwise the matrix would not be simple. From this fact, we are eventually able to find two rows or two columns with sufficiently small Hamming distance. Therefore they can be contracted. Admittedly some technicalities are involved to preserve the simplicity of the matrix throughout the contraction process. So we adopt a two-step algorithm: In the first step, we build a sequence of partition coarsenings over the matrix, and in the second step, we extract the actual sequence of contractions. The overall algorithm taking A (or σ) as input, and outputting the contraction sequence, takes polynomial time in n. It can be implemented in quadratic time, or even faster if instead of the raw matrix, we get a list of pointers to corners of A.

We shall now find mixed-free orders. Section V is devoted to this task for three different classes. Dealing with permutations avoiding a fixed pattern (equivalently, a proper subclass of posets of dimension 2), the order is easy to find: it is imposed. For posets of bounded width (that is, maximum size of an antichain or minimum size of a chain partition), a mixed-free order is attained by putting the chains in increasing order, one after the other. Finally for K_t-minor free graphs, a hamiltonian path would provide a good order. As we cannot always expect to find a hamiltonian path, we simulate it by a specific Lex-DFS. The top part of Figure 3 provides a visual summary of this section.

D. How general are classes of bounded twin-width?

As announced in the previous section, we will show that proper minor-closed classes have bounded twin-width. As far as we know, all classes of polynomial expansion may also have bounded twinwidth. However on the one hand, cubic graphs have unbounded twin-width, whereas on the other hand, cliques have twin-width 0. Thus bounded twin-width is incomparable with bounded degree, bounded expansion, and nowhere denseness.

Nowhere dense classes are *stable*, that is, no arbitrarily-long total order can be first-order interpreted from graphs of this class. In particular, unit interval graphs are not FO interpretations (even FO *transductions*, where in addition copying the structure and *coloring* it with a constant number of unary relations is allowed) of nowhere dense graphs. Thus even any class of first-order transductions of nowhere dense graphs is incomparable with bounded twin-width graphs. We will show that bounded twin-width is preserved by FO interpretations and transductions, which makes it a robust class as far as first-order model checking is concerned. Despite cubic graphs having unbounded twin-width, some particular classes with bounded degree, such as subgraphs of d-dimensional grids, have bounded twin-width. This showcases the ubiquity of bounded twin-width, and the wide scope of Theorem 1. As we will generalize twin-width to matrices, in order to handle permutations, posets, and digraphs, we can potentially define a twin-width notion on hypergraphs, groups, and lattices.

E. Organization of the paper

Section II gives the necessary graph-theoretic and logic background. In Section III we formally introduce contraction sequences and the twin-width of a graph. At this point, the interested reader is referred to Section 4 of the full version [14] for classes of graphs shown of bounded twin-width by direct arguments. In particular we show there that bounded rank-width graphs, d-dimensional grids, and unit d-dimensional ball graphs with bounded clique number, have bounded twin-width. In Section IV we extend twin-width to matrices and show a grid-minor-like theorem, which informally states that a graph has large twin-width if and only if all its vertex orderings yield an adjacency matrix with a complex large submatrix. This turns out to be a useful characterization for the next section. In Section V we show how, thanks to this characterization, we can compute a witness of bounded twin-width, for permutations avoiding a fixed pattern, comparability graphs with bounded independence number (equivalently, bounded-width posets), and K_t-minor free graphs. In Section VI we give the statement of the linear-time FPT algorithm for FO model checking on graphs given with a witness of bounded twin-width. We also see that FO interpretations (even transductions) of classes of bounded twin-width still have bounded twin-width.

All the proofs are omitted from the current paper due to space restrictions. These proofs can be found in the full version [14].

II. Preliminaries

We denote by $[i, j]$ the set of integers $\{i, i+1, \dots, j-1, j\}$, and by $[i]$ the set of integers $[1, i]$. If $\mathcal{X}$ is a set of sets, we denote by $\cup\mathcal{X}$ the union of them.

A. Graph definitions and notations

All our graphs are undirected and simple (no multiple edge nor self-loop). We denote by $V(G)$, respectively $E(G)$, the set of vertices, respectively of edges, of the graph G. For $S \subseteq V(G)$, we denote the *open neighborhood* (or simply *neighborhood*) of S by $N_G(S)$, i.e., the set of neighbors of S deprived of S, and the *closed neighborhood* of S by $N_G[S]$, i.e., the set $N_G(S) \cup S$. For singletons, we simplify $N_G(\{v\})$ into $N_G(v)$, and $N_G[\{v\}]$ into $N_G[v]$. We denote by $G[S]$ the subgraph of G induced by S, and $G - S := G[V(G) \setminus S]$. For $A, B \subseteq V(G)$, $E(A, B)$ denotes the set of edges in $E(G)$ with one endpoint in A and the other one in B. Two distinct vertices u, v such that $N(u) = N(v)$ are called *false twins*, and *true twins* if $N[u] = N[v]$. In particular, true twins are adjacent. Two vertices are *twins* if they are false twins or true twins. If G is an n-vertex graph and σ is a total ordering of $V(G)$, say, $v_1, \dots, v_n$, then $A_\sigma(G)$ denotes the adjacency matrix of G in the order σ. Thus the entry in the i-th row and j-th column is a 1 if $v_i v_j \in E(G)$ and a 0 otherwise.

The length of a path in an unweighted graph is simply the number of edges of the path. For two vertices $u, v \in V(G)$, we denote by $d_G(u, v)$, the distance between u and v in G, that is the length of the shortest path between u and v. The diameter of a graph is the longest distance between a pair of its vertices. In all the above notations with a subscript, we omit it whenever the graph is implicit from the context.

An *edge contraction* of two adjacent vertices u, v consists of merging u and v into a single vertex adjacent to

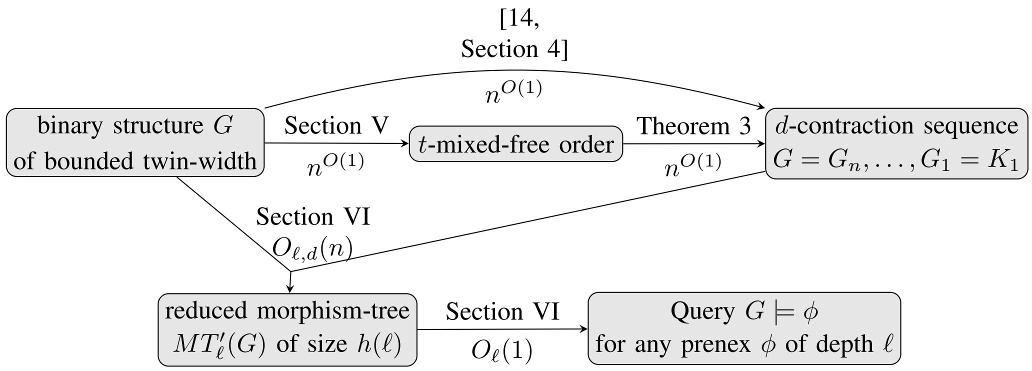

Figure 3. The overall workflow. Two paths are possible to get a d-contraction sequence from a bounded twin-width structure G. Either a direct polytime algorithm as for bounded boolean-width, or via a domain-ordering yielding a t-mixed free matrix followed by Theorem 3 which converts it into a d-contraction sequence. From there, a tree of constant size (function of ℓ only) can be computed in linear FPT time. This tree captures the evaluation of all prenex sentences ϕ on ℓ variables for G. Queries "$G \models \phi$" can then be answered in constant time.

$N(\{u, v\})$ (and deleting u and v). A graph H is a *minor* of a graph G if H can be obtained from G by a sequence of vertex and edge deletions, and edge contractions. A graph G is said *H-minor free* if H is not a minor of G. Importantly we will overload the term "contraction". In this paper, we call *contraction* the same as an edge contraction without the requirement that the two vertices u and v are adjacent. This is sometimes called an *identification*, but we stick to the shorter *contraction* since we will use that word often. In the very rare cases in which we actually mean the classical (edge) contraction, the context will lift the ambiguity. We will also somewhat overload the term "minor". Indeed, in Section IV we introduce the notions of "d-grid minor" and "d-mixed minor" on matrices. They are only loosely related to (classical) graph minors, and it will always be clear which notion is meant.

B. First-order logic, model checking, FO interpretations/transductions

For our purposes, we define first-order logic without function symbols. A finite *relational signature* is a set τ of relation (or *predicate*) symbols given with their arity $\{R^1_{a_1}, \ldots, R^h_{a_h}\}$; that is, relation $R^i_{a_i}$ has arity a_i. A first-order formula $\phi \in \mathrm{FO}(\tau)$ over τ is any string generated from letter ψ by the grammar:

$$\psi \to \exists x \psi,\ \forall x \psi,\ \psi \vee \psi,\ \psi \wedge \psi,\ \neg\psi,\ (\psi),\ R^1_{a_1}(x, \ldots, x),$$

$$\ldots,\ R^h_{a_h}(x, \ldots, x),\ x = x,\ \text{and}$$

$$x \to x_1, x_2, \ldots \ \text{an infinite set of fresh variable labels.}$$

For the sake of simplicity, we will further impose that the same label cannot be reused for two different variables. A variable x_i is then said *quantified* if it appears next to a quantifier ($\forall x_i$ or $\exists x_i$), and *free* otherwise. We usually denote by $\phi(x_{f_1}, \ldots, x_{f_h})$ a formula whose free variables are precisely $x_{f_1}, \ldots, x_{f_h}$. A formula without quantified variables is said *quantifier-free*. A *sentence* is a formula without free variables. With our simplification that the same label is not used for two distinct variables, when a formula ϕ contains a subformula $Q x_i \phi'$ (with $Q \in \{\exists, \forall\}$), all the occurrences of x_i in ϕ lie in ϕ'.

Model checking.: A first-order (FO) formula is purely syntactical. An *interpretation*, *model*, or *structure* $\mathcal{M}$ of the FO language FO(τ) specifies a *domain of discourse* D for the variables, and a relation $\mathcal{M}(R^i_{a_i}) = R^i \subseteq D^{a_i}$ for each symbol $R^i_{a_i}$. $\mathcal{M}$ is sometimes called a *τ-structure*. $\mathcal{M}$ is a *binary structure* if τ has only relation symbols of arity 2. It is said *finite* if the domain D is finite. A sentence ϕ interpreted by $\mathcal{M}$ is *true*, denoted by $\mathcal{M} \models \phi$, if it evaluates to true with the usual semantics for quantified Boolean logic, the equality, and $R^i_{a_i}(d_1, \ldots, d_{a_i})$ is true if and only if $(d_1, \ldots, d_{a_i}) \in \mathcal{M}(R^i_{a_i})$. For a fixed interpretation, a formula ϕ with free variables $x_{f_1}, \ldots, x_{f_h}$ is *satisfiable* if $\exists x_{f_1} \cdots \exists x_{f_h} \phi$ is true.

In the FO model checking problem, given a first-order sentence $\phi \in \mathrm{FO}(\tau)$ and a finite model $\mathcal{M}$ of FO(τ), one has to decide whether $\mathcal{M} \models \phi$ holds. The input size is $|\phi| + |\mathcal{M}|$, the number of bits necessary to encode the sentence ϕ and the model $\mathcal{M}$. The brute-force algorithm decides $\mathcal{M} \models \phi$ in time $|\mathcal{M}|^{|\phi|}$, by building the tree of all possible assignments. We will consider ϕ to be fixed or rather small compared to $|\mathcal{M}|$. Therefore we wish to find an FPT algorithm for FO model checking parameterized by $|\phi|$, that is, running in time $f(|\phi|)|\mathcal{M}|^{O(1)}$, or even better $f(|\phi|)|D|$.

We restrict ourselves to FO model checking on finite binary structures, for which twin-width will be eventually defined. For the most part, we will consider FO model checking on graphs (and we may omit the signature τ). Let us give a simple example. Let $\tau = \{E_2\}$ be a signature with a single binary relation. Finite models of the language FO(τ) correspond to finite directed graphs with possible self-loops. Let ϕ be the sentence $\exists x_1 \exists x_2 \cdots \exists x_k \bigwedge_{i<j} \neg(x_i =$

$x_j) \wedge \bigwedge_{i \neq j} \neg E(x_i, x_j)$. Let G be a τ-structure or graph. $G \models \phi$ holds if and if G has an independent set of size k. This problem parameterized by $|\phi|$ (or equivalently k) is W[1]-hard on general graphs. However it may admit an FPT algorithm when G belongs to a specific class of graphs, as in the case, for instance, of planar graphs or bounded-degree graphs.

FO interpretations and transductions.: An FO interpretation of a τ-structure $\mathcal{M}$ is a τ-structure $\mathcal{M}'$ such that for every relation R of $\mathcal{M}'$, $R(a_1, \dots, a_h)$ is true if and only if $\mathcal{M} \models \phi_R(a_1, \dots, a_h)$ for a fixed formula $\phi_R(x_1, \dots, x_h) \in \text{FO}(\tau)$. Informally every relation of $\mathcal{M}'$ can be characterized by a formula evaluated on $\mathcal{M}$.

Again we shall give some example on graphs since it is our main focus. Let G be a simple undirected graph (in particular, $E(x, y)$ holds whenever $E(y, x)$ holds). Then the *FO (ϕ-)interpretation* $I_\phi(G)$ is a graph H with vertex-set $V(G)$ and $uv \in E(H)$ if and only if $G \models \phi(x, y) \wedge \phi(y, x)$. If for instance $\phi(x, y)$ is the formula $\neg E(x, y)$, then $I_\phi(G)$ is the complement of G. If instead $\phi(x, y)$ is $E(x, y) \vee \exists z E(x, z) \wedge E(z, y)$, then $I_\phi(G)$ is the square of G. The FO (ϕ-)interpretation of a class $\mathcal{C}$ of graphs is the set of all graphs that are ϕ-interpretations of graphs in $\mathcal{C}$, namely $I_\phi(\mathcal{C}) := \{H \mid H = I_\phi(G),\ G \in \mathcal{C}\}$. It is not very satisfactory that $I_\phi(\mathcal{C})$ is not hereditary. We will therefore either close $I_\phi(\mathcal{C})$ by taking induced subgraphs, or use the more general notion of FO transductions (see for instance [16]).

An FO transduction is an enhanced FO interpretation. A graph augmented by unary relations $U^1, \dots, U^h$ is called a *colored* graph. The vertex $v \in V(G)$ can be thought as having color i if $v \in U_i$ (note that the same vertex may have one color, several colors, or no color at all). The *(ϕ, γ, h)-transduction $\mathcal{T}_{\phi,\gamma,h}(G)$ of a graph G* is the set of all graphs H that are $(\phi, U^1, \dots, U^h)$-interpretations of any graph obtained by coloring an induced subgraph of G with some unary relations $U^1, \dots, U^h$ and duplicating the result γ times. Note that the colors are only relevant to define H, so we may forget about them and think of H as an uncolored graph. Similarly to FO interpretations of classes, we define $\mathcal{T}_{\phi,\gamma,h}(\mathcal{C}) := \{H \mid H \in \mathcal{T}_{\phi,\gamma,h}(G),\ G \in \mathcal{C}\}$. Now $\mathcal{T}_{\phi,\gamma,h}(\mathcal{C})$ is by definition a (hereditary) class of graphs.

III. Sequence of contractions and twin-width

We say that two vertices u and v are *twins* if they have the same neighborhood outside $\{u, v\}$. A natural operation is to contract (or identify) them and try to iterate the process. If this algorithm leads to a single vertex, the graph was initially a *cograph*. Many intractable problems become easy on cographs. It is thus tempting to try and extend this tractability to larger classes. One such example is the class of graphs with bounded clique-width (or equivalently bounded rank-width) for which any problem expressible in MSO_1 logic can be solved in polynomial-time [10]. A perhaps more direct generalization (than defining clique-width) would be to allow contractions of near twins, but the cumulative effect of the errors[4] stands as a barrier to algorithm design.

An illuminating example is provided by a bipartite graph G, with bipartition (A, B), such that for every subset X of A there is a vertex $b \in B$ with neighborhood X in A. Surely G is complex enough so that we should not entertain any hope of solving a problem like, say, k-Dominating Set significantly faster on any class containing G than on general graphs. For one thing, graphs like G contain all the bipartite graphs as induced subgraphs. Nonetheless G can be contracted to a single vertex by iterating contractions of vertices whose neighborhoods differ on only one vertex. Indeed, consider $a \in A$ and contract all pairs of vertices of B differing exactly at a. Applying this process for every $a \in A$, we end up by contracting the whole set B, and we can eventually contract A.

Thus the admissibility of a contraction sequence should not solely be based on the current neighborhoods. The key idea is to keep track of the past errors in the contraction history and always require all the vertices to be involved in only a limited number of mistakes. Say the errors are carried by the edges, and an erroneous edge is recorded as *red*. Note that in the previous contraction sequence of G, after contracting all pairs of vertices of B differing at a, all the edges incident to a are red, and vertex a witnesses the non-admissibility of the sequence. Let us now get more formal.

It appears, from the previous paragraphs, that the appropriate structure to define twin-width is a graph in which some edges are colored red. A *trigraph* is a triple $G = (V, E, R)$ where E and R are two disjoint sets of edges on V: the (usual) edges and the *red edges*. An informal interpretation of a red edge $uv \in R$ is that some errors have been made while handling G and the existence of an edge between u and v, or lack thereof, is uncertain. A trigraph (V, E, R) such that (V, R) has maximum degree at most d is a *d-trigraph*. We observe that any graph (V, E) may be interpreted as the trigraph $(V, E, \emptyset)$.

Given a trigraph $G = (V, E, R)$ and two vertices u, v in V, we define the trigraph $G/u, v = (V', E', R')$ obtained by *contracting*[5] u, v into a new vertex w as the trigraph on vertex-set $V' = (V \setminus \{u, v\}) \cup \{w\}$ such that $G - \{u, v\} = (G/u, v) - \{w\}$ and with the following edges incident to w:

- $wx \in E'$ if and only if $ux \in E$ and $vx \in E$,
- $wx \notin E' \cup R'$ if and only if $ux \notin E \cup R$ and $vx \notin E \cup R$, and
- $wx \in R'$ otherwise.

In other words, when contracting two vertices u, v, red edges stay red, and red edges are created for every vertex

[4]By *error* we informally refer to the elements in the (non-empty) symmetric difference in the neighborhoods of the contracted vertices.

[5]Or *identifying*. Let us insist that u and v do not have to be adjacent.

x which is not joined to u and v at the same time. We say that $G/u, v$ is a *contraction of* G. If both G and $G/u, v$ are d-trigraphs, $G/u, v$ is a *d-contraction*. We may denote by $V(G)$ the vertex-set, $E(G)$ the set of *black* edges, and $R(G)$ the set of *red* edges, of the trigraph G.

A (tri)graph G on n vertices is *d-collapsible* if there exists a sequence of d-contractions which contracts G to a single vertex. More precisely, there is a *d-sequence* of d-trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_{i-1} is a contraction of G_i (hence G_1 is the singleton graph). See Figure 1 for an example of a sequence of 2-contractions of a 7-vertex graph. The minimum d for which G is d-collapsible is the *twin-width* of G, denoted by $\mathrm{tww}(G)$.

If v is a vertex of G_i and $j \geqslant i$, then $v(G_j)$ denotes the subset of vertices of G_j eventually contracted into v in G_i. Two disjoint vertex-subsets A, B of a trigraph are said *homogeneous* if there is no red edge between A and B, and there are not both an edge and a non-edge between A and B. In other words, A and B are fully linked by black edges or there is no (black or red) edge between them. Observe that in any contraction sequence $G = G_n, \ldots, G_i, \ldots, G_1$, there is a red edge between u and v in G_i if and only if $u(G)$ and $v(G)$ are not homogeneous. We may sometimes (abusively) identify a vertex $v \in G_i$ with the subset of vertices of G contracted to form v.

One can check that cographs have twin-width 0 (the class of graphs with twin-width 0 actually coincides with cographs), paths of length at least three have twin-width 1, red paths have twin-width at most 2, and trees have twin-width 2. Indeed, they are not 1-collapsible, as exemplified by the 1-subdivision of $K_{1,3}$, and they admit the following 2-sequence. Choose an arbitrary root and contract two leaves with the same neighbor, or, if not applicable, contract the highest leaf with its neighbor. We observe that in this 2-sequence, every G_i only contains red edges which are adjacent to leaves. In particular, red edges are either isolated or are contained in a path of length two.

The definition of twin-width readily generalizes to directed graphs, where we create a red edge whenever the contracted vertices u, v are not linked to x in the same way. This way we may speak of the twin-width of a directed graph or of a partial order. One could also wish to define twin-width on graphs "colored" by a constant number of unary relations. To have a unifying framework, in the next section we work with matrices.

IV. The grid theorem for twin-width

In this section, we will deal with matrices instead of graphs. Our matrices have their entries on a finite alphabet with a special additional value r (for red) representing errors made along the computations. This is the analog of the red edges of the previous section.

A. Twin-width of matrices, digraphs, and binary structures

The *red number* of a matrix is the maximum number of red entries taken over all rows and all columns. Given an $n \times m$ matrix M and two columns C_i and C_j, the *contraction* of C_i and C_j is obtained by deleting C_j and replacing every entry $m_{k,i}$ of C_i by r whenever $m_{k,i} \neq m_{k,j}$. The same contraction operation is defined for rows. A matrix M has *twin-width* at most k if one can perform a sequence of contractions starting from M and ending in some 1×1 matrix in such a way that all matrices occurring in the process have red number at most k. Note that when M has twin-width at most k, one can reorder its rows and columns in such a way that every contraction will identify consecutive rows or columns. The reordered matrix is then called *k-twin-ordered*. The *symmetric twin-width* of an $n \times n$ matrix M is defined similarly, except that the contraction of rows i and j (resp. columns i and j) is immediately followed by the contraction of columns i and j (resp. rows i and j).

We can now extend the twin-width to digraphs, which in particular capture posets. Unsurprisingly the twin-width of a digraph is defined as the symmetric twin-width of its adjacency matrix; only we write the adjacency matrix in a specific way. Say, the vertices are labeled $v_1, \ldots, v_n$. If there is an arc v_iv_j (but no arc v_jv_i), we place a 1 entry in the i-th row j-column of the matrix and a -1 entry in the j-th row i-th column. If there are two arcs v_iv_j and v_jv_i, we place a 2 entry in both the i-th row j-column and j-th row i-th column. If there is no arc v_iv_j nor v_jv_i, we place a 0 entry in both the i-th row j-column and j-th row i-th column. We then further extend twin-width to a binary structure S with binary relations $E^1, \ldots, E^h$. When building the adjacency matrix, the entry at v_i, v_j is now $(e_1, \ldots, e_h)$ where $e_p \in \{-1, 0, 1, 2\}$ is chosen accordingly to the encoding of the "digraph E^p". Again the twin-width of a binary structure is the symmetric twin-width of the so-built adjacency matrix. One can show that adding k unary relations can at most multiply the twin-width by 2^k (see full version [14]). Thus we will not consider additional unary relations in this section.

Given a total order σ on the domain of a binary structure G, we denote by $A_\sigma(G)$ the adjacency matrix encoded accordingly to the previous paragraph and following the order σ. Denoting $M := A_\sigma(G) = (m_{ij} = (e_1^{ij}, \ldots, e_h^{ij}))_{i,j}$, the matrix M satisfies the important following property, mixing symmetry and skew-symmetry. If $e_p^{ij} \in \{0, 2\}$ then $e_p^{ij} = e_p^{ji}$, and if $e_p^{ij} \in \{-1, 1\}$ then $e_p^{ij} = -e_p^{ji}$. We call this property *mixed-symmetry* and M is said *mixed-symmetric*. This will be useful to find *symmetric* sequences of contractions.

B. Partition coarsening, contraction sequence, and error value

Here we present an equivalent way of seeing the twin-width with a successive coarsening of a partition, instead of explicitly performing the contractions with deletion.

A partition $\mathcal{P}$ of a set S *refines* a partition $\mathcal{P}'$ of S if every part of $\mathcal{P}$ is contained in a part of $\mathcal{P}'$. Conversely we say that $\mathcal{P}'$ is a coarsening of $\mathcal{P}$, or *contains* $\mathcal{P}$. When every part of $\mathcal{P}'$ contains at most k parts of $\mathcal{P}$, we say that $\mathcal{P}$ *k-refines* $\mathcal{P}'$. Given a partition $\mathcal{P}$ and two distinct parts P, P' of $\mathcal{P}$, the *contraction* of P and P' yields the partition $\mathcal{P} \setminus \{P, P'\} \cup \{P \cup P'\}$.

Given an $n \times m$ matrix M, a *row-partition* (resp. *column-partition*) is a partition of the rows (resp. columns) of M. A (k, ℓ)*-partition* (or simply *partition*) of a matrix M is a pair $(\mathcal{R} = \{R_1, \ldots, R_k\}, \mathcal{C} = \{C_1, \ldots, C_\ell\})$ where $\mathcal{R}$ is a row-partition and $\mathcal{C}$ is a column-partition. A *contraction* of a partition $(\mathcal{R}, \mathcal{C})$ of a matrix M is obtained by performing one contraction in $\mathcal{R}$ or in $\mathcal{C}$.

We distinguish two extreme partitions of an $n \times m$ matrix M: the *finest partition* where $(\mathcal{R}, \mathcal{C})$ have size n and m, respectively, and the *coarsest partition* where they both have size one. The finest partition is sometimes called the *partition of singletons*, since all its parts are singletons, and the coarsest partition is sometimes called the *trivial partition*. A *contraction sequence* of an $n \times m$ matrix M is a sequence of partitions $(\mathcal{R}^1, \mathcal{C}^1), \ldots, (\mathcal{R}^{n+m-1}, \mathcal{C}^{n+m-1})$ where

- $(\mathcal{R}^1, \mathcal{C}^1)$ is the finest partition,
- $(\mathcal{R}^{n+m-1}, \mathcal{C}^{n+m-1})$ is the coarsest partition, and
- for every $i \in [n+m-3]$, $(\mathcal{R}^{i+1}, \mathcal{C}^{i+1})$ is a contraction of $(\mathcal{R}^i, \mathcal{C}^i)$.

Given a subset R of rows and a subset C of columns in a matrix M, the *zone* $R \cap C$ denotes the submatrix of all entries of M at the intersection between a row of R and a column of C. A *zone* of a partition pair $(\mathcal{R}, \mathcal{C}) = (\{R_1, \ldots, R_k\}, \{C_1, \ldots, C_\ell\})$ is any $R_i \cap C_j$ for $i \in [k]$ and $j \in [\ell]$. A zone is *constant* if all its entries are identical. The *error value* of R_i is the number of non constant zones among all zones in $\{R_i \cap C_1, \ldots, R_i \cap C_\ell\}$. We adopt a similar definition for the *error value* of C_j. The *error value* of $(\mathcal{R}, \mathcal{C})$ is the maximum error value taken over all R_i and C_j.

We can now restate the definition of twin-width of a matrix M as the minimum t for which there exists a contraction sequence of M consisting of partitions with error value at most t. The following easy technical lemma will be used later to upper bound twin-width.

Lemma 1: If $(\mathcal{R}^1, \mathcal{C}^1), \ldots, (\mathcal{R}^s, \mathcal{C}^s)$ is a sequence of partitions of a matrix M such that:

- $(\mathcal{R}^1, \mathcal{C}^1)$ is the finest partition,
- $(\mathcal{R}^s, \mathcal{C}^s)$ is the coarsest partition,
- $\mathcal{R}^i$ r-refines $\mathcal{R}^{i+1}$ and $\mathcal{C}^i$ r-refines $\mathcal{C}^{i+1}$, and
- all $(\mathcal{R}^i, \mathcal{C}^i)$ have error value at most t,

then the twin-width of M is at most rt.

C. Matrix division and Marcus-Tardos theorem

In a contraction sequence of a matrix M, one can always reorder the rows and the columns of M in such a way that all parts of all partitions in the contraction sequence consist of consecutive rows or consecutive columns. To mark this distinction, a *row-division* is a row-partition where every part consists of consecutive rows; with the analogous definition for *column-division*. A (k, ℓ)*-division* (or simply *division*) of a matrix M is a pair $(\mathcal{R}, \mathcal{C})$ of a row-division and a column-division with respectively k and ℓ parts. A *fusion* of a division is obtained by contraction of two consecutive parts of $\mathcal{R}$ or of $\mathcal{C}$. Fusions are just contractions preserving divisions. A *division sequence* is a contraction sequence in which all partitions are divisions.

We now turn to the fundamental tool which is basically only applied once but is the cornerstone of twin-width. Given a $0,1$-matrix $M = (m_{i,j})$, a *t-grid minor* in M is a (t, t)-division $(\mathcal{R}, \mathcal{C})$ of M in which every zone contains a 1. We say that a matrix is *t-grid free* if it does not have a *t-grid minor*. A celebrated result by Marcus and Tardos [15] (henceforth *Marcus-Tardos theorem*) asserts that every $0,1$-matrix with large enough linear density has a t-grid minor. Precisely:

Theorem 2: [15] For every integer t, there is some c_t such that every $n \times m$ $0,1$-matrix M with at least $c_t \max(n, m)$ entries 1 has a t-grid minor.

Marcus and Tardos established this theorem with $c_t = 2t^4\binom{t^2}{t}$. Fox [17] subsequently improved the bound to $3t2^{8t}$. He also showed that c_t has to be superpolynomial in t (at least $2^{\Omega(t^{1/4})}$). Then Cibulka and Kynčl [18] decreased c_t further down to $8/3(t+1)^2 2^{4t}$.

Matrices with enough 1 entries are complex in the sense that they contain large t-grids minors. However here the role of 1 is special compared to 0, and this result is only interesting for sparse matrices. We would like to extend this notion of complexity to the dense case, that is to say for all matrices. In Marcus-Tardos theorem zones are *not simple* if they contain a 1, that is, if they have rank at least 1. A natural definition would consist of substituting "rank at least 1" by "rank at least 2" in the definition of a t-grid minor. Since we mostly deal with $0,1$-matrices, and exclusively with discrete objects, we adopt a more combinatorial approach.

D. Mixed minor and the grid theorem for twin-width

A matrix $M = (m_{i,j})$ is *vertical* (resp. *horizontal*) if $m_{i,j} = m_{i+1,j}$ (resp. $m_{i,j} = m_{i,j+1}$) for all i, j. Observe that a matrix which is both vertical and horizontal is constant. We say that M is *mixed* if it is neither vertical nor horizontal. A *t-mixed minor* in M is a division $(\mathcal{R}, \mathcal{C}) = (\{R_1, \ldots, R_t\}, \{C_1, \ldots, C_t\})$ such that every zone $R_i \cap C_j$ is mixed. A matrix without t-mixed minor is *t-mixed free*. For instance, the $n \times n$ matrix with all entries equal to 1 is 1-mixed free but admits an n-grid minor. See Figure 4 for some more examples.

$$
\left[\begin{array}{cc|cc|cc|cc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]
\left[\begin{array}{cc|ccc|ccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}\right]
$$

Figure 4. To the left a 4-grid minor: every zone contains at least one 1. To the right a 3-mixed minor on the same matrix: no zone is horizontal or vertical.

The main result of this section is that t-mixed free matrices are exactly matrices with bounded twin-width, modulo reordering the rows and columns. More precisely:

Theorem 3: Let α be the alphabet size for the matrix entries, and $c_t := 8/3(t+1)^2 2^{4t}$.

- Every t-twin-ordered matrix is $2t+2$-mixed free.
- Every t-mixed free matrix has twin-width at most $4c_t\alpha^{4c_t+2} = 2^{2^{O(t)}}$.

E. Corners

The proof of Theorem 3 will crucially rely on the notion of *corner*. Given a matrix $M = (m_{i,j})$, a *corner* is any 2-by-2 mixed submatrix of the form $(m_{i,j}, m_{i+1,j}, m_{i,j+1}, m_{i+1,j+1})$. Corners will play the same role as the 1 entries in Marcus-Tardos theorem, as they localize the property of being mixed:

Lemma 2: A matrix is mixed if and only if it contains a corner.

F. Mixed zones, cuts, and values

Let $\mathcal{R} = \{R_1, \ldots, R_k\}$ be a row-division of a matrix M and let C be a set of consecutive columns. We call *mixed zone* of C on $\mathcal{R}$ any zone $R_i \cap C$ which is a mixed matrix. We call *mixed cut* of C on $\mathcal{R}$ any index $i \in [k-1]$ for which the 2-by-$|C|$ zone defined by the last row of R_i, the first row of R_{i+1}, and C is a mixed matrix. Now the *mixed value* of C on $\mathcal{R}$ is the sum of the number of mixed cuts and the number of mixed zones. Analogously we define the mixed value of a set R of consecutive rows on a column-division $\mathcal{C}$.

Lemma 3: The contraction of two consecutive parts of $\mathcal{R}$ does not increase the mixed value of C on $\mathcal{R}$.

The *mixed value* of a division $(\mathcal{R}, \mathcal{C}) = (\{R_1, \ldots, R_k\}, \{C_1, \ldots, C_\ell\})$ is the maximum mixed value of R_i on $\mathcal{C}$, and of C_j on $\mathcal{R}$, taken over all $R_i \in \mathcal{R}$ and $C_j \in \mathcal{C}$. Observe that the finest division has mixed value 0 and the coarsest division has mixed value at most 1.

G. Finding a division sequence with bounded mixed value

Leveraging Marcus-Tardos theorem, we are ready to compute, for any t-mixed free matrix, a division sequence with bounded mixed value. This division sequence is not necessarily yet a contraction sequence with bounded error value (indeed a non-constant horizontal or vertical zone counts for 0 in the mixed value but for 1 in the error value). But this division sequence will serve as a crucial frame to find the eventual contraction sequence.

Lemma 4: Every t-mixed free matrix M has a division sequence in which all divisions have mixed value at most $2c_t$ (where c_t is the one of Theorem 2).

H. Finding a contraction sequence with bounded error value

We are now equipped for the main result of this section, which is the second item of Theorem 3. The division sequence with small mixed value, provided by Lemma 4, will guide the construction of a contraction sequence (not necessarily a division sequence) of bounded error value. This two-layered mechanism is also present in the proof of Guillemot and Marx, albeit in a simpler form since they have it tailored for sparse matrices, and importantly they start from a permutation matrix.

The second item of Theorem 3 has the following consequence, which reduces the task of bounding the twin-width of G and finding a contraction sequence to merely exhibiting a *mixed free order*, that is a domain-ordering σ such that the matrix $A_\sigma(G)$ is t-mixed free for a bounded t.

Theorem 4: Let G be a (di)graph or even a binary structure. If there is an ordering $\sigma : v_1, \ldots, v_n$ of $V(G)$ such that $A_\sigma(G)$ is k-mixed free, then $\text{tww}(G) = 2^{2^{O(k)}}$.

V. Classes with bounded twin-width

In this section we show that some classical classes of graphs and matrices have bounded twin-width. Let us start with the origin of twin-width, which is the method proposed by Guillemot and Marx [1] to understand permutation matrices avoiding a certain pattern.

A. Pattern-avoiding permutations

We associate to a permutation σ over $[n]$ the $n \times n$ matrix $M_\sigma = (m_{ij})_{i,j}$ where $m_{i\sigma(i)} = 1$ and all the other entries are set to 0. A permutation σ is a *pattern* of a permutation τ if M_σ is a submatrix of M_τ. A central open question was the design of an algorithm deciding if a pattern σ appears in a permutation τ in time $f(|\sigma|) \cdot |\tau|^{O(1)}$. The brilliant idea of Guillemot and Marx, reminiscent of treewidth and grid minors, is to observe that permutations avoiding a pattern σ can be iteratively decomposed (or collapsed), and that the decomposition gives rise to a dynamic-programming scheme. This lead them to a linear-time $f(|\sigma|) \cdot |\tau|$ algorithm for permutation pattern recognition. In Sections III and IV we generalized their decomposition to graphs and arbitrary (dense) matrices, and leveraged Marcus-Tardos theorem, also in the dense setting. Section IV would in principle readily apply here: If a permutation matrix M_τ does not contain a fixed pattern of size k, then it is certainly k-mixed free since otherwise the k-mixed minor would contain any pattern of size k. Hence by Theorem 3, M_τ has bounded twin-width.

B. Posets of bounded width

The versatility of the grid minor theorem for twin-width is also illustrated with posets. Let $P = (X, \leqslant)$ be a poset of *width* k, that is, its maximum antichain has size k. For $x_i, x_j \in X$, $x_i < x_j$ denotes that $x_i \leqslant x_j$ and $x_i \neq x_j$. We claim that the twin-width of P is bounded by a function of k. By Dilworth's theorem, P can be partitioned into k total orders (or *chains*) $T_1, \ldots, T_k$. Now one can enumerate the vertices precisely in this order, say σ, that is, increasingly with respect to T_1, then increasingly with respect to T_2, and so on. We rename the elements of X so that in the order σ, they read $x_1, x_2, \ldots, x_n$, with $n := |X|$. Let us write the adjacency matrix $A = (a_{ij}) := A_\sigma(P)$ of P: $a_{ij} = 1$ if $x_i \leqslant x_j$, $a_{ij} = -1$ if $x_j < x_i$, and $a_{ij} = 0$ otherwise. Recall that this is consistent with how we defined the adjacency matrix for the more general digraphs in Section IV. We assume for contradiction that A has a $3k$-mixed minor.

By the pigeon-hole principle, there is a submatrix of A indexed by two chains, T_i for the row indices and T_j for the column indices, which has a 3-mixed minor, realized by the $(3,3)$-division $(R_1, R_2, R_3), (C_1, C_2, C_3)$. The zone $R_2 \cap C_2$ is mixed, so it contains a -1 or a 1. If it is a -1, then by transitivity the zone $R_3 \cap C_1$ is entirely -1, a contradiction to its being mixed. A similar contradiction holds when there is a 1 entry in $R_2 \cap C_2$: zone $R_1 \cap C_3$ is entirely 1. See Figure 5 for an illustration. Hence, by Theorem 3, the twin-width of A (and the twin-width of P seen as a directed graph) is bounded by $4c_k \cdot 4^{4c_k+2} = 2^{2^{O(k)}}$.

Of course there was a bit of work to establish Theorem 3 inspired by the Guillemot-Marx framework, and supported by Marcus-Tardos theorem. There was even more work to prove that FO model checking is FPT on bounded twin-width (di)graphs. It is nevertheless noteworthy that once that theory is established, the proof that bounded twin-width captures the posets of bounded width is lightning fast. Indeed the known FPT algorithm on posets of bounded width [9] is a strong result, itself generalizing or implying the tractability of FO model checking on several geometric classes [11], [12]. We observe that posets of bounded twin-width constitute a strict superset of posets of bounded width. Arcless posets are trivial separating examples, which have unbounded maximum antichain and twin-width 0.

C. Proper minor-closed classes

A more intricate example is given by proper minor-closed classes. By definition, a proper minor-closed class does not contain some graph H as a minor. This implies in particular that it does not contain $K_{|V(H)|}$ as a minor. Thus we only need to show that K_t-minor free graphs have bounded twin-width.

If the K_t-minor free graph G admits a hamiltonian path, things become considerably simpler. We can enumerate the vertices of G according to this path and write the corresponding adjacency matrix A. The crucial observation is that a k-mixed minor yields a $K_{k/2,k/2}$-minor, hence a $K_{k/2}$-minor. So A cannot have a $2t$-mixed minor, and by Theorem 3, the twin-width of G bounded (by $4c_{2t}2^{4c_{2t}+2} = 2^{t^{O(t)}}$). Unfortunately, a hamiltonian path is not always granted in G. A depth-first search (DFS for short) tree may emulate the path, but any DFS will not necessarily work. Interestingly the main tool of the following theorem is a carefully chosen Lex-DFS.

Theorem 5: We set $g : t \mapsto 2(2^{4t+1}+1)^2$, $c_k := 8/3(k+1)^2 2^{4k}$, and $f : t \mapsto 4c_{g(t)}2^{4c_{g(t)}+2}$. Every K_t-minor free graph have twin-width at most $f(t) = 2^{2^{2^{O(t)}}}$.

Applied to planar graphs, which are K_5-minor free, the previous theorem gives us a constant bound on the twin-width, but that constant has billions of digits. We believe that the correct bound should have only one digit. It is natural to ask for a more reasonable bound in the case of planar graphs.

VI. FO MODEL CHECKING

In this section, we see that deciding first-order properties in d-collapsible graphs is fixed-parameter tractable in d and the size of the formula. We let E be a binary relation symbol. A graph G is seen as an $\{E\}$-structure with universe $V(G)$ and binary relation $E(G)$ (matching the arity of E). A *sentence* is a formula without free variables.

A *formula ϕ in prenex normal form*, or simply *prenex formula*, is any sentence written as a sequence of non-negated quantifiers followed by a quantifier-free formula:

$$\phi = Q_1 x_1 Q_2 x_2 \ldots Q_\ell x_\ell \phi^*$$

where for each $i \in [\ell]$, the variable x_i ranges over $V(G)$, $Q_i \in \{\forall, \exists\}$, while ϕ^* is a Boolean combination in atoms of the form $x_i = x_j$ and $E(x_i, x_j)$. The *length* of ϕ is ℓ. Note that this also corresponds to its quantifier depth. Every formula with quantifier depth k can be rewritten as a prenex formula of depth Tower$(k + \log^* k + 3)$ (see Theorem 2.2. and inequalities (32) in [19]).

Theorem 6: Given as input a prenex formula ϕ of length ℓ, an n-vertex graph G, and a d-sequence of G, one can decide $G \models \phi$ in time $f(\ell, d) \cdot n$.

Bounded twin-width is also preserved by FO interpretations and even FO transductions.

Theorem 7: Any (ϕ, γ, h)-transduction of a graph with twin-width at most d has twin-width bounded by a function of ϕ and d.

As a direct consequence, map graphs have bounded twin-width since they can be obtained by FO transductions of planar graphs (which have bounded twin-width).

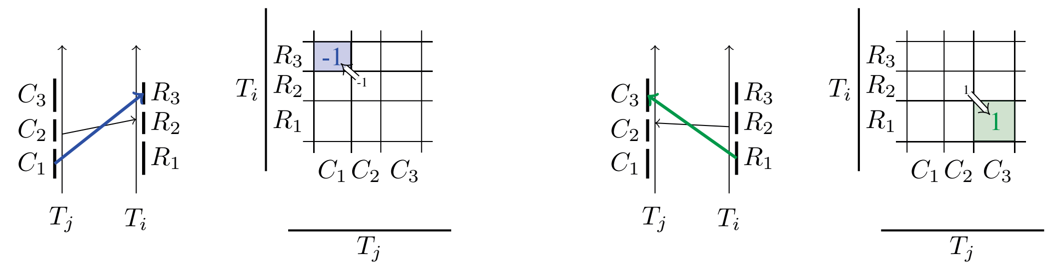

Figure 5. Left: If there is one arc from C_2 to R_2, then by transitivity there are all arcs from C_1 to R_3. On the matrix, this translates as: a -1 entry in $R_2 \cap C_2$ implies that all the entries of $R_3 \cap C_1$ are -1. Right: Similarly, a 1 entry in $R_2 \cap C_2$ implies that all the entries of $R_1 \cap C_3$ are 1. Hence at least one zone among $R_3 \cap C_1$, $R_2 \cap C_2$, $R_1 \cap C_3$ is constant, a contradiction to the $3k$-mixed minor.

References

[1] S. Guillemot and D. Marx, "Finding small patterns in permutations in linear time," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, 2014, pp. 82–101. [Online]. Available: https://doi.org/10.1137/1.9781611973402.7

[2] D. Seese, "Linear time computable problems and first-order descriptions," *Mathematical Structures in Computer Science*, vol. 6, no. 6, pp. 505–526, 1996.

[3] M. Frick and M. Grohe, "Deciding first-order properties of locally tree-decomposable structures," *J. ACM*, vol. 48, no. 6, pp. 1184–1206, 2001. [Online]. Available: https://doi.org/10.1145/504794.504798

[4] J. Flum and M. Grohe, "Fixed-parameter tractability, definability, and model-checking," *SIAM J. Comput.*, vol. 31, no. 1, pp. 113–145, 2001. [Online]. Available: https://doi.org/10.1137/S0097539799360768

[5] A. Dawar, M. Grohe, and S. Kreutzer, "Locally excluding a minor," in *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, 2007, pp. 270–279. [Online]. Available: https://doi.org/10.1109/LICS.2007.31

[6] Z. Dvorák, D. Král, and R. Thomas, "Testing first-order properties for subclasses of sparse graphs," *J. ACM*, vol. 60, no. 5, pp. 36:1–36:24, 2013. [Online]. Available: https://doi.org/10.1145/2499483

[7] M. Grohe, S. Kreutzer, and S. Siebertz, "Deciding first-order properties of nowhere dense graphs," *J. ACM*, vol. 64, no. 3, pp. 17:1–17:32, 2017. [Online]. Available: https://doi.org/10.1145/3051095

[8] S. Kreutzer and A. Dawar, "Parameterized complexity of first-order logic," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 16, p. 131, 2009. [Online]. Available: http://eccc.hpi-web.de/report/2009/131

[9] J. Gajarský, P. Hlinený, D. Lokshtanov, J. Obdrzálek, S. Ordyniak, M. S. Ramanujan, and S. Saurabh, "FO model checking on posets of bounded width," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 963–974. [Online]. Available: https://doi.org/10.1109/FOCS.2015.63

[10] B. Courcelle, J. A. Makowsky, and U. Rotics, "Linear time solvable optimization problems on graphs of bounded clique-width," *Theory Comput. Syst.*, vol. 33, no. 2, pp. 125–150, 2000. [Online]. Available: https://doi.org/10.1007/s002249910009

[11] R. Ganian, P. Hlinený, D. Král, J. Obdrzálek, J. Schwartz, and J. Teska, "FO model checking of interval graphs," *Logical Methods in Computer Science*, vol. 11, no. 4, 2015. [Online]. Available: https://doi.org/10.2168/LMCS-11(4:11)2015

[12] P. Hlinený, F. Pokrývka, and B. Roy, "FO model checking on geometric graphs," *Comput. Geom.*, vol. 78, pp. 1–19, 2019. [Online]. Available: https://doi.org/10.1016/j.comgeo.2018.10.001

[13] K. Eickmeyer and K. Kawarabayashi, "FO model checking on map graphs," in *Fundamentals of Computation Theory - 21st International Symposium, FCT 2017, Bordeaux, France, September 11-13, 2017, Proceedings*, 2017, pp. 204–216. [Online]. Available: https://doi.org/10.1007/978-3-662-55751-8_17

[14] Édouard Bonnet and Eun Jung Kim and Stéphan Thomassé and Rémi Watrigant, "Twin-width I: tractable FO model checking," *CoRR*, vol. abs/2004.14789, 2020. [Online]. Available: https://arxiv.org/abs/2004.14789

[15] A. Marcus and G. Tardos, "Excluded permutation matrices and the stanley-wilf conjecture," *J. Comb. Theory, Ser. A*, vol. 107, no. 1, pp. 153–160, 2004. [Online]. Available: https://doi.org/10.1016/j.jcta.2004.04.002

[16] A. Blumensath and B. Courcelle, "On the monadic second-order transduction hierarchy," *Logical Methods in Computer Science*, vol. 6, no. 2, 2010. [Online]. Available: https://doi.org/10.2168/LMCS-6(2:2)2010

[17] J. Fox, "Stanley-wilf limits are typically exponential," *CoRR*, vol. abs/1310.8378, 2013. [Online]. Available: http://arxiv.org/abs/1310.8378

[18] J. Cibulka and J. Kyncl, "Füredi-hajnal limits are typically subexponential," *CoRR*, vol. abs/1607.07491, 2016. [Online]. Available: http://arxiv.org/abs/1607.07491

[19] O. Pikhurko and O. Verbitsky, "Logical complexity of graphs: a survey," *Model theoretic methods in finite combinatorics*, vol. 558, pp. 129–179, 2011.