Faster Existential FO Model Checking on Posets

Jakub Gajarský, Petr Hliněný, Jan Obdržálek(✉), and Sebastian Ordyniak

Faculty of Informatics, Masaryk University, Brno, Czech Republic
{gajarsky,hlineny,obdrzalek,ordyniak}@fi.muni.cz

Abstract. We prove that the model checking problem for the existential fragment of first order (FO) logic on partially ordered sets is fixed-parameter tractable (FPT) with respect to the formula and the width of a poset (the maximum size of an antichain). While there is a long line of research into FO model checking on graphs, the study of this problem on posets has been initiated just recently by Bova, Ganian and Szeider (LICS 2014), who proved that the existential fragment of FO has an FPT algorithm for a poset of fixed width. We improve upon their result in two ways: (1) the runtime of our algorithm is $O(f(|\phi|, w) \cdot n^2)$ on n-element posets of width w, compared to $O(g(|\phi|) \cdot n^{h(w)})$ of Bova et al., and (2) our proofs are simpler and easier to follow. We complement this result by showing that, under a certain complexity-theoretical assumption, the existential FO model checking problem does not have a polynomial kernel.

1 Introduction

The *model checking* problem, asking whether a logical formula holds true on a given input structure, is a fundamental problem of theoretical computer science with applications in many different areas, e.g. algorithm design or formal verification. One way to see why providing efficient algorithms for model checking is important is to note that such algorithms automatically establish efficient solvability of whole classes of problems. For the first-order (FO) logic, the model checking problem is known to be PSPACE-complete when the formula is part of the input, and polynomial time solvable when the formula is fixed in advance.

However, this does not tell the whole story. In the latter scenario we would like to identify the instances where we could do significantly better—in regard to running times—and quantify these gains. Stated in the parlance of parameterized complexity theory, we wish to identify classes of input structures on which we can evaluate every FO formula ϕ in polynomial time $f(|\phi|) \cdot n^c$, where c is a constant independent of the formula. If it is true, we say that FO model checking problem is *fixed-parameter tractable* (FPT) on this class of structures.

Research funded by the Czech Science Foundation under grant 14-03501S.
Sebastian Ordyniak: Research funded by Employment of Newly Graduated Doctors of Science for Scientific Excellence (CZ.1.07/2.3.00/30.0009).

H.-K. Ahn and C.-S. Shin (Eds.): ISAAC 2014, LNCS 8889, pp. 441–451, 2014.
DOI: 10.1007/978-3-319-13075-0_35

Over the past decade this line of research has been very active and led to several important results on (mainly) undirected graphs, which culminated in the recent result of Grohe, Kreutzer and Siebertz [10], stating that FO model checking is fixed-parameter tractable on all nowhere dense classes of graphs.

In contrast, almost nothing is known about the complexity of FO model checking on other finite algebraic structures. Very recently, Bova, Ganian and Szeider [4] initiated the study of the model checking problem for FO and partially ordered sets. Despite similarities between posets and graphs (e.g., in Hasse diagrams), the existing FO model checking results from graphs do not seem to transfer well to posets, perhaps due to lack of usable notions of "locality" and "sparsity" there. This feeling is supported by several negative results in [4], too.

The main result of Bova et al. [4] then is that the model checking problem for the existential fragment of FO (Poset ∃-FO-Model Checking) can be solved in time $f(|\phi|) \cdot n^{g(w)}$, where n is the size of a poset and w its *width*, i.e. the size of its largest antichain. In the language of parameterized complexity, this means that the problem is FPT in the size of the formula, but only XP with respect to the width of the poset. Note that this is not an easy result since, for instance, posets of fixed width can have unbounded clique-width [4].

The proof in [4] goes by first showing that the model checking problem for the existential fragment of FO is equivalent to the embedding problem for posets (which can be thought as analogous to the induced subgraph problem), and then reducing the embedding problem to a suitable family of instances of the homomorphism problem of certain semilattice structures.

While postponing further formal definitions till Section 2, we now state our main result which improves upon the aforementioned result of Bova et al.:

Theorem 1. Poset ∃-FO-Model Checking *is fixed-parameter tractable in the formula size and the width of an input poset; precisely, solvable in time $h(|\phi|, w) \cdot O(n^2)$ where n is the size of a poset and w its width.*

Our improvement is two-fold; (1) we show that the existential FO model checking problem is fixed-parameter tractable in *both* the size of the formula and the width of the poset, and (2) we give two simpler proofs of this result, one of them completely *self-contained*. Regarding improvement (2), we use the same reduction of existential FO model checking to the embedding problem from [4], but our subsequent solution to embedding is faster and at the same time much more straightforward and easier to follow.

As stated above, we give two different FPT algorithms solving the poset embedding problem (and thus also the existential FO model checking problem). The first algorithm (Section 3) is a natural, and easy to understand, polynomial-time reduction to a CSP (Constraint Satisfaction Problem) instance closed under min polymorphisms, giving us an $O(n^4)$ dependence of the running time on the size of the poset. The second algorithm (Section 4) has even better, quadratic, time complexity and works by reducing the embedding problem to a restricted variant of the multicoloured clique problem, which is then efficiently solved.

To complement the previous fixed-parameter tractability results, we also investigate possible kernelization of the embedding problem for posets (Section 5).

We show that the embedding problem does not have a polynomial kernel, unless co**NP** $\subseteq$ **NP**/poly, which is thought to be unlikely. This means the embedding problem (and therefore also the existential and full FO model checking problems) cannot be efficiently reduced to an equivalent instance of size polynomial in the parameter.

Full version of the paper, which includes all omitted proofs, is available at arxiv.org [9].

2 Preliminaries

2.1 Posets and Embedding

A *poset* $\mathcal{P}$ is a pair $(P, \leq^P)$ where P is a set and $\leq^P$ is a reflexive, antisymmetric, and transitive binary relation over P. The *size* of a poset $\mathcal{P} = (P, \leq^P)$ is $\|\mathcal{P}\| := |P|$. We say that p *covers* p' for $p, p' \in P$, denoted by $p' \lhd^P p$, if $p' \leq^P p$, $p \neq p'$, and for every p'' with $p' \leq^P p'' \leq^P p$ it holds that $p'' \in \{p, p'\}$. We say that p and p' are *incomparable* (in $\mathcal{P}$), denoted $p \parallel^P p'$ if neither $p \leq^P p'$ nor $p' \leq^P p$. A *chain* C of $\mathcal{P}$ is a subset of P such that $x \leq^P y$ or $y \leq^P x$ for every $x, y \in C$. An *anti-chain* A of $\mathcal{P}$ is a subset of P such that for all $x, y \in P$ it is true that $x \parallel^P y$. A *chain partition* of $\mathcal{P}$ is a tuple $(C_1, \ldots, C_k)$ such that $\{C_1, \ldots, C_k\}$ is a partition of P and for every i with $1 \leq i \leq k$ the poset induced by C_i is a chain of $\mathcal{P}$. The *width* of a poset $\mathcal{P}$, denoted by $\mathrm{width}(\mathcal{P})$ is the maximum cardinality of any anti-chain of $\mathcal{P}$.

Proposition 2.1 ([7, Theorem 1]). *Let $\mathcal{P}$ be a poset. Then in time $O(\mathrm{width}(\mathcal{P}) \cdot \|\mathcal{P}\|^2)$, it is possible to compute both $\mathrm{width}(\mathcal{P}) = w$ and a corresponding chain partition $(C_1, \ldots, C_w)$ of $\mathcal{P}$.*

Let $\mathcal{Q} = (Q, \leq^P)$ and $\mathcal{P} = (P, \leq^P)$ be two posets. An *embedding* from $\mathcal{Q}$ to $\mathcal{P}$ is an injective function $e : Q \to P$ such that $q \leq^Q q'$ if, and only if, $f(q) \leq^P f(q')$ for every $q, q' \in Q$. The *embedding problem* for posets is thus defined as:

EMBEDDING **Parameter:** $\mathrm{width}(\mathcal{P})$, $\|\mathcal{Q}\|$
Input: Two posets $\mathcal{Q} = (Q, \leq^Q)$ and $\mathcal{P} = (P, \leq^P)$.
Question: Is there an embedding from $\mathcal{Q}$ into $\mathcal{P}$?

2.2 Constraint Satisfaction Problems

A *constraint satisfaction problem* (CSP) I is a triple $\langle V, D, C\rangle$, where V is a finite set of variables over a finite set (domain) D, and C is a set of constraints. A *constraint* $c \in C$ consists of a *scope*, denoted by $V(c)$, which is an ordered subset of V, and a relation, denoted by $R(c)$, which is a $|V(c)|$-ary relation on D. For a CSP $I = \langle V, D, C\rangle$ we sometimes denote by $V(I)$, $D(I)$, and $C(I)$, its set of variables V, its domain D, and its set of constraints C, respectively. A *solution* to a CSP instance I is a mapping $\tau : V \to D$ such that $\langle \tau[v_1], \ldots, \tau[v_{|V(c)|}]\rangle \in R(c)$ for every $c \in C$ with $V(c) = \langle v_1, \ldots, v_{|V(c)|}\rangle$.

Given a k-ary relation R over some domain D and a function $\phi : D^n \rightarrow D$, we say that R is *closed under* ϕ, if for all collections of n tuples $t_1, \ldots, t_n$ from R, the tuple $\langle \phi(t_1[1], \ldots, t_n[1]), \ldots, \phi(t_1[k], \ldots, t_n[k]) \rangle$ belongs to R. The function ϕ is also said to be a *polymorphism of* R. We denote by $\text{Pol}(R)$ the set of all polymorphisms ϕ such that R is closed under ϕ.

Let $I = \langle V, D, C \rangle$ be a CSP instance and $c \in C$. We write $\text{Pol}(c)$ for the set $\text{Pol}(R(c))$ and we write $\text{Pol}(I)$ for the set $\bigcap_{c \in C} \text{Pol}(c)$. We say that I is closed under a polymorphism ϕ if $\phi \in \text{Pol}(I)$.

We will need the following type of polymorphism. A polymorphism $\phi : D^2 \rightarrow D$ is a *min* polymorphism if there is an ordering of the elements of D such that for every $d, d' \in D$, it holds that $\phi(d, d') = \phi(d', d) = \min\{d, d'\}$.

Proposition 2.2 ([11, Corollary 4.3]). *Any CSP instance I that is closed under a min polymorphism (that is provided with the input) can be solved in time $O((ct)^2)$, where $c = |C(I)|$ and t is the maximum cardinality of any constraint relation of I.*

2.3 Parameterized Complexity

Here we introduce the relevant concepts of parameterized complexity theory. For more details, we refer to text books on the topic [6,8,12]. An instance of a parameterized problem is a pair $\langle x, k \rangle$ where x is the input and k a parameter. A parameterized problem is *fixed-parameter tractable* if every instance $\langle x, k \rangle$ can be solved in time $f(k) \cdot |x|^c$, where f is a computable function, and c is a constant. **FPT** denotes the class of all fixed-parameter tractable problems.

A *kernelization* [1] for a parameterized problem $\mathcal{A}$ is a polynomial time algorithm that takes an instance $\langle x, k \rangle$ of $\mathcal{A}$ and maps it to an equivalent instance $\langle x', k' \rangle$ of $\mathcal{A}$ such that both $|x'|$ and k' are bounded by some function f of k. The output $\langle x', k' \rangle$ is called a *kernel*. We say that $\mathcal{A}$ has a *polynomial kernel* if f is a polynomial. Every fixed-parameter tractable problem admits a kernel, but not necessarily a polynomial kernel [5].

A *polynomial parameter reduction* from a parameterized problem $\mathcal{A}$ to a parameterized problem $\mathcal{B}$ is a polynomial time algorithm, which, given an instance $\langle x, k \rangle$ of $\mathcal{A}$ produces an instance $\langle x', k' \rangle$ of $\mathcal{B}$ such that $\langle x, k \rangle$ is a YES-instance of $\mathcal{A}$ if and only if $\langle x', k' \rangle$ is a YES-instance of $\mathcal{B}$ and k' is bounded by some polynomial of k. The following results show how polynomial parameter reductions can be employed to prove the non-existence of polynomial kernels.

Proposition 2.3 ([2, Theorem 8]). *Let $\mathcal{A}$ and $\mathcal{B}$ be two parameterized problems such that there is a polynomial parameter reduction from $\mathcal{A}$ to $\mathcal{B}$. If $\mathcal{B}$ has a polynomial kernel, then so has $\mathcal{A}$.*

An *OR-composition algorithm* for a parameterized problem $\mathcal{A}$ maps t instances $\langle x_1, k \rangle, \ldots, \langle x_t, k \rangle$ of $\mathcal{A}$ to one instance $\langle x', k' \rangle$ of $\mathcal{A}$ such that the algorithm runs in time polynomial in $\sum_{1 \leq i \leq t} |x_i| + k$, the parameter k' is bounded by a polynomial in the parameter k, and $\langle x', k' \rangle$ is a YES-instance if and only if there exists $1 \leq i \leq t$ such that $\langle x_i, k \rangle$ is a YES-instance.

Proposition 2.4 ([3, Lemmas 1 and 2]). *If a parameterized problem $\mathcal{A}$ has an OR-composition algorithm and its unparameterized version is* $\mathbf{NP}$*-complete, then $\mathcal{A}$ has no polynomial kernel, unless* co$\mathbf{NP} \subseteq \mathbf{NP}/\text{poly}$*.*

2.4 Existential First-Order Logic

In this paper we deal with the, well known, relational first-order (FO) logic. Formulas of this logic are built from (a finite set of) variables, relational symbols, logical connectives $(\wedge, \vee, \neg)$ and quantifiers $(\exists, \forall)$. A sentence is a formula with no free variables. We restrict ourselves to formulas that are in *prefix normal form*. (A first-order formula is in prefix normal form if all quantifiers occur in front of the formula and all negations occur in front of the atoms.) Furthermore an *existential* first-order formula is a first-order formula in prefix normal form that uses only existential quantifiers.

The problem we are interested in is so-called *model checking problem* for the existential FO formulas (and posets), which is formally defined as follows:

Poset $\exists$-FO-Model Checking **Parameter:** width($\mathcal{P}$), $|\phi|$
Input: An existential first-order sentence ϕ and a poset $\mathcal{P} = (P, \leq^P)$.
Question: Is it true $\mathcal{P} \models \phi$, i.e., is $\mathcal{P}$ a model of ϕ?

We remark here that all first-order formulas in this paper are evaluated over posets. In particular, the vocabulary of these formulas consists of only one binary relation $\leq^P$ and atoms of these formulas can be either equalities between variables $(x = y)$ or applications of the predicate $\leq^P$ $(x \leq^P y)$. For a more detailed treatment of the employed setting, we refer the reader to [4].

As shown in [4], the existential FO model checking problem is closely related to the aforementioned *embedding problem* for posets:

Proposition 2.5 ([4]). Poset $\exists$-FO-Model Checking *is fixed-parameter tractable if and only if so is* Embedding*. Moreover, there is a polynomial parameter reduction from* Embedding *to* Poset $\exists$-FO-Model Checking*.*

Proof. The first statement of the proposition follows immediately from [4, Proposition 1]. The second statement of the proposition follows from the proof of [4, Proposition 1] by observing that the obvious reduction from Embedding to Poset $\exists$-FO-Model Checking is polynomial parameter preserving. □

Remark 1. Even though [4] does not state the precise runtime and "instance blow-up" for Proposition 2.5, these can be alternatively bounded from above as follows. For an instance $(\mathcal{P}, \phi)$ where $\phi \equiv \exists x_1 \dots \exists x_q .\, \psi(x_1, \dots, x_q)$, we exhaustively enumerate all posets $\mathcal{Q}$ on $Q = \{x_1, \dots, x_q\}$ (modulo equality $=$ on Q) such that $\mathcal{Q} \models \psi$, and produce a separate instance of Embedding from this particular $\mathcal{Q}$ into the same $\mathcal{P}$. Then $\mathcal{P} \models \phi$ if and only if at least one of the constructed Embedding instances is Yes. The number of produced instances (of $\mathcal{Q}$) is trivially less than the number of all posets on q elements factorized by equality, $< 4^{q^2} = 2^{O(|\phi|^2)}$, and time spent per each one of them in the construction is $O(|\phi|^2)$.

3 Fixed-Parameter Tractability Proof

In this section we prove the first half of the main result of our paper (Theorem 1) that the existential FO model checking problem for posets is in FPT. By Proposition 2.5, it is enough to consider the embedding problem for that:

Theorem 3.1. *Let $\mathcal{Q} = (Q, \leq^Q)$ and $\mathcal{P} = (P, \leq^P)$ be two posets. Then the embedding problem from $\mathcal{Q}$ into $\mathcal{P}$ is fixed-parameter tractable, more precisely, it can be solved in time $O\big(\mathrm{width}(\mathcal{P})^{|Q|} \cdot |Q|^4 \cdot |P|^4\big)$.*

The remainder of this section is devoted to a proof of the above theorem. Let $w := \mathrm{width}(\mathcal{P})$ for the rest of this section. The algorithm starts by computing a chain partition $\mathcal{C} = (C_1, \dots, C_w)$ of $\mathcal{P}$. This can be done in time $O(\mathrm{width}(\mathcal{P}) \cdot |P|^2)$ by Proposition 2.1.

To make the proof clearer, we will, for an embedding, keep track into which chain each element of $\mathcal{Q}$ is mapped. We say that an embedding e from $\mathcal{Q}$ into $\mathcal{P}$ is *compatible* with a function f from Q to $\{1, \dots, w\}$ if $e(q) \in C_{f(q)}$ for every $q \in Q$. Observe that every embedding e is trivially compatible with the unique function f, where $f(q) = i$ if and only if $e(q) \in C_i$. Also note that there are at most $(\mathrm{width}(\mathcal{P})^{|Q|})$ such functions f.

Our algorithm now will do the following: We generate all possible functions f (as defined in the previous paragraph) and for each such f we test whether there is an embedding compatible with f. The following lemma, stating that we can perform such test efficiently, forms the core of our proof.

Lemma 3.2. *Let f be a function from Q to $\{1, \dots, w\}$ where $w = \mathrm{width}(\mathcal{P})$. Then one can decide in time $O\big(|Q|^4 \cdot |P|^4\big)$ whether there is an embedding e from $\mathcal{Q}$ to $\mathcal{P}$ that is compatible with f.*

Proof. We will prove the lemma by reducing the problem (of finding a compatible embedding) in polynomial time to a CSP instance that is closed under a certain min polymorphism and hence can be solved in polynomial time. We start by defining the CSP instance I for given $\mathcal{Q}$, $\mathcal{P}$, f, and $\mathcal{C}$ as above.

I has one variable x_q for every $q \in Q$ whose domain are the elements of $C_{f(q)}$. Furthermore, for every pair q, q' of distinct elements of Q, I contains one constraint $c_{q,q'}$ whose scope is $(x_q, x_{q'})$ and whose relation $R(c_{q,q'})$ contains all tuples (p, p') such that $p \in C_{f(q)}$, $p' \in C_{f(q')}$, and simultaneously

1. $p \leq^P p'$ iff $q \leq^Q q'$,
2. $p' \leq^P p$, iff $q' \leq^Q q$.

This completes the construction of I. Observe that a solution $\tau : V(I) \to D(I)$ of I gives rise to an embedding $e : Q \to P$ from $\mathcal{Q}$ to $\mathcal{P}$ that is compatible with f by setting $e(q) = \tau(x_q)$. Additionally, every embedding $e : Q \to P$ from $\mathcal{Q}$ to $\mathcal{P}$ that is compatible with f gives rise to a solution $\tau : V(I) \to D(I)$ of I by setting $\tau(x_q) = e(q)$. Hence, I has a solution if and only if there is an embedding from $\mathcal{Q}$ to $\mathcal{P}$ that is compatible with f and such an embedding can be easily obtained from a solution of I.

Concerning the runtime, I can be constructed in time $O((|Q| \cdot |P|)^2)$. Since there are less than $|Q|^2$ constraints and every constraint relation contains $O(|P|^2)$ pairs, Proposition 2.2 provides a solution to I in time $O((|Q|^2 \cdot |P|^2)^2)$. To finish it is enough to verify that I is closed under a certain min polymorphism—Lemma 3.3 below. □

Lemma 3.3. *For every $\mathcal{Q}$, $\mathcal{P}$, f, and $\mathcal{C}$ defined as above, the CSP instance I is closed under any min polymorphism that is compatible with the partial order $\leq^P$.*

Proof (of Theorem 3.1). We can generate the chain partition in time $O(\mathrm{width}(\mathcal{P}) \cdot |P|^2)$. Then, for each of the $(\mathrm{width}(\mathcal{P})^{|Q|})$ functions f we test the existence of an embedding compatible with f, which can be done in time $O\big(|Q|^4 \cdot |P|^4\big)$ by Lemma 3.2. This proves our theorem. □

4 Embedding and Multicoloured Clique

In the previous section we have proved that the embedding problem for posets $\mathcal{Q}$ and $\mathcal{P}$ is fixed-parameter tractable w.r.t. both $\mathrm{width}(\mathcal{P})$ and $\|\mathcal{Q}\|$, with the running time of $O\big(\mathrm{width}(\mathcal{P})^{\|\mathcal{Q}\|} \cdot \|\mathcal{Q}\|^4 \cdot \|\mathcal{P}\|^4\big)$. In this section we improve upon this result by giving an alternative self-contained algorithm for Embedding with running time $O\big(\mathrm{width}(\mathcal{P})^{\|\mathcal{Q}\|} \cdot \|\mathcal{Q}\|^3 \cdot \|\mathcal{P}\|^2\big)$. In combination with Proposition 2.5 (and Remark 1) we thus finish the proof of main Theorem 1.

This new algorithm achieves better efficiency by exploiting some special properties of the problem that are not fully utilized in the previous reduction to CSP. We pay for this improvement by having to work a little bit harder. The core idea is to show that the problem of finding a compatible embedding is reducible (in polynomial time) to a certain restricted variant of Multicoloured Clique.

The Multicoloured Clique problem takes as an input a graph G together with a proper k-colouring of the vertices of G. The question is whether there is a k-clique in G, i.e., a clique consisting of exactly one vertex of each colour.

Multicoloured Clique **Parameter:** k
Input: A graph G with a proper k-colouring of its vertices.
Question: Is there a clique (set of pairwise adjacent vertices) of size k in G?

Consider a chain partition $(C_1, \ldots, C_w)$ of $\mathcal{P} = (P, \leq^P)$ where $w = \mathrm{width}(\mathcal{P})$, and $\mathcal{Q} = (Q, \leq^Q)$ with $Q = \{1, \ldots, k\}$. Let $f : Q \to \{1, \ldots, w\}$. For an f-compatible poset embedding instance from $\mathcal{Q}$ into $\mathcal{P}$, we construct a k-coloured clique instance G simply as follows. The vertex set is a disjoint union $V(G) = V_1 \dot\cup \ldots \dot\cup V_k$ of k colour classes where V_i, $1 \leq i \leq k$, is a copy of $C_{f(i)}$. Let $p \in V_a, q \in V_b$ be copies of $p' \in C_{f(a)}, q' \in C_{f(b)}$. Then $pq \in E(G)$ if and only if $a \neq b$ and the following hold; $p' \leq^P q'$ iff $a \leq^Q b$, and $p' \geq^P q'$ iff $a \geq^Q b$. We associate each class V_i of G with a linear order $\leq^G$ naturally inherited from the corresponding chain of $\mathcal{P}$ (we are not going to compare between different classes).

Trivially, constructed G is a Yes-instance of k-coloured clique if, and only if, $\mathcal{Q}$ has an f-compatible embedding into $\mathcal{P}$.

Lemma 4.1. *Let G be a graph on $V(G) = V_1 \dot\cup \ldots \dot\cup V_k$ constructed from a compatible embedding instance as above. Suppose any $1 \leq a < b \leq k$.*

a) For any $p \in V_a$, $q_1, q_2, q_3 \in V_b$ such that $q_1 \leq^G q_2 \leq^G q_3$ it holds; if pq_1, $pq_3 \in E(G)$ then also $pq_2 \in E(G)$.
b) For any $p_1, p_2 \in V_a$, $q_1, q_2 \in V_b$ such that $p_1 \leq^G p_2$, $q_1 \leq^G q_2$ it holds; if p_1q_2, $p_2q_1 \in E(G)$ then also p_1q_1, $p_2q_2 \in E(G)$.

We call a Multicoloured Clique instance G *interval-monotone* if the colour classes of G can be given linear order(s) such that both claims a),b) of Lemma 4.1 are satisfied.

Corollary 4.2. *Let G be an interval-monotone (wrt. $\leq^G$) multicoloured clique instance with colour classes $V_1, \ldots, V_k$. Let $I \subseteq \{1, \ldots, k\}$. If $K_1, \ldots, K_\ell \subseteq \bigcup_{i \in I} V_i$ are cliques of size $|I|$, then also the set*

$$K = \left\{ \min_{\leq^G} \big((K_1 \cup \cdots \cup K_\ell) \cap V_i \big) \mid i \in I \right\},$$

called the minimum of $K_1, \ldots, K_\ell$ *wrt. $\leq^G$ and I, is a clique in G. The same holds for analogous* maximum of $K_1, \ldots, K_\ell$ *wrt. $\leq^G$ and I.*

For smooth explanation of our algorithm, we introduce the following shorthand notation. Let $[i, j] = \{i, i + 1, \ldots, j\}$. Let $V(G) = V_1 \dot\cup \ldots \dot\cup V_k$. Then $N_i(v)$ denotes the set of neighbours of $v \in V(G)$ in V_i, and moreover, $N_I(v) := \bigcup_{i \in I} N_i(v)$ and $N_I(X) := \bigcap_{v \in X} N_I(v)$. Provided that G is equipped with linear order(s) $\leq^G$ on each V_i, $N_i^{\uparrow}(v)$ denotes the set of all $w \in V_i$ such that there is $w' \in N_i(v)$ and $w' \leq^G w$ (all the vertices which are "above" some neighbour of v in V_i), and this is analogously extended to $N_I^{\uparrow}(v)$ and $N_I^{\uparrow}(X)$.

Algorithm 4.1. Input: An interval-monotone k-coloured clique instance G, the colours classes $V(G) = V_1 \dot\cup \ldots \dot\cup V_k$ and the order $\leq^G$ on them.

Output: Yes if G contains a clique of size k, and No otherwise.

Algorithm: Dynamically compute, for $i = 2, 3, \ldots, k$, sets $MinK^i(v)$ and $MaxK^i(v)$ where $v \in V_i$; such that $MinK^i(v)$ is the minimum of all the cliques of size i in G which are contained in $\{v\} \cup V_1 \cup \cdots \cup V_{i-1}$ (note, these cliques must contain v), or $\emptyset$ if nonexistent, and $MaxK^i(v)$ is described analogously.

The computation of $MaxK^i, MinK^i$ using values $MaxK^2, \ldots, MaxK^{i-1}$ and $MinK^2, \ldots, MinK^{i-1}$ is described in the pseudocode below. Note that we have to compute both $MinK^i$ and $MaxK^i$ because we compute $MinK^i$ from previously computed $MaxK^j$, $j < i$, and vice versa.

```
1. For every $v \in V_i$, set $X := \{v\}$ and repeat:
   a) For $j = i - 1, \ldots, 1$, and as long as $X \neq \emptyset$, do the following:
      find the minimum (wrt. $\leq^G$) element $x \in N_j(X)$ such that $j = 1$ or
      $\emptyset \neq MaxK^j(x) \subseteq N^{\uparrow}_{[1,j-1]}(X) \cup \{x\}$. If $x$ does not exist then $X := \emptyset$,
      and otherwise set $X := X \cup \{x\}$. Continue with next $j$.
   b) Set $MinK^i(v) := X$.
```

2. Analogously finish computation of $MaxK^i(v)$ using previous $MinK^j(x)$.
3. Output YES if there is $v \in V_k$ such that $MinK^k(v) \neq \emptyset$, and NO otherwise.

Theorem 4.4. *Algorithm 4.1 correctly solves any instance G of interval-monotone k-coloured* MULTICOLOURED CLIQUE *problem, in time $O(k \cdot |E(G)|)$.*

Proof. It is enough to prove that the value of each $MinK^i(v)$ and $MaxK^i(v)$ is computed correctly in the algorithm. Let $K_{i,v}$ be the minimum of all the cliques of size i in G which are contained in $\{v\} \cup V_1 \cup \cdots \cup V_{i-1}$ (well-defined by Corollary 4.2)—the correct value for $MinK^i(v)$. Assume that some $MinK^i(v) = K'_{i,v}$ value is computed wrong, i.e., $K_{i,v} \neq K'_{i,v}$, and that i is minimal among such wrong values. Clearly, $i > 2$.

If $K'_{i,v} = \emptyset$ then $K_{i,v} \neq \emptyset = K'_{i,v}$. Otherwise we observe that, by the choices $x \in N_j(X)$ in step 1.a), $K'_{i,v} \neq \emptyset$ is a clique of size i in G contained in $\{v\} \cup V_1 \cup \cdots \cup V_{i-1}$. Consequently, $K'_{i,v} \neq \emptyset$ implies $K_{i,v} \neq \emptyset$, too.

Let $K''_{i,v} = K'_{i,v}$ if $K'_{i,v} \neq \emptyset$, and otherwise let $K''_{i,v}$ be the last nonempty value of X in the course of computation of $MinK^i(v)$ in step 1.a) of the algorithm. Since the tests in step 1.a) of the algorithm always succeed for x being $K_{i,v} \cap V_j$ and $X = K_{i,v} \cap (V_{j+1} \cup \cdots \cup V_i)$, there exists $j < i$ (and we choose such j maximum) such that $\{x\} = K_{i,v} \cap V_j \neq K''_{i,v} \cap V_j = \{x'\}$. By the same argument, actually, $x >^G x'$.

Now, following iteration j of step 1.a) of the algorithm (which has "wrongly" chosen x' instead of x), let $K_0 = MaxK^j(x') \cup \big(K_{i,v} \cap (V_{j+1} \cup \cdots \cup V_i)\big)$. The minimum of $K_{i,v}$ and K_0 is also a clique of size i, by the interval-monotone property and Corollary 4.2, contradicting minimality of $K_{i,v}$ at x.

In any case, indeed $K_{i,v} = K'_{i,v}$.

It remains to analyse the running time. We consider separately every iteration of step 1, each $v \in V_i$, for $i = 2, \ldots, k$. Thanks to the interval-monotone property of G, we can preprocess the neighbours of v into subintervals of the classes $V_1, \ldots, V_{i-1}$ with respect to $\leq^G$. This is done in time $O\big(|N_{[1,i-1]}(v)|\big)$. After that, every iteration j of step 1.a) takes time $O\big(|N_j(v)| \cdot k\big)$, and so whole step 1 takes time $O\big(k \cdot |N_{[1,i-1]}(v)|\big)$. Summing this over v and i as in the algorithm we arrive right at the estimate $O(k \cdot |E(G)|)$. □

Corollary 4.5. EMBEDDING *can be solved in time $O\big(\mathrm{width}(\mathcal{P})^{|Q|} \cdot |Q|^3 \cdot |P|^2\big)$.*

Proof. The reduction from embedding to compatible embedding has been shown within Theorem 3.1. By the reduction here, $|V(G)| = O(|Q| \cdot |P|)$, $|E(G)| = |V(G)|^2$, and $k = |Q|$. The runtime bound thus follows as in Theorem 3.1. □

5 Kernelization Lower Bound

Having shown that the EMBEDDING problem is fixed-parameter tractable, it becomes natural to ask whether it also allows for a polynomial kernel. In this section we will show that this unfortunately is not the case, i.e., we show that

Embedding does not have a polynomial kernel unless co**NP** ⊆ **NP**/poly. Consequently, this also excludes a polynomial kernel for the Poset FO-Model Checking problem, of which Embedding is a special case. (Poset FO-Model Checking is an extension of Poset ∃-FO-Model Checking to the full FO logic.)

We will show our kernelization lower bound for Embedding using the OR-composition technique outlined by Proposition 2.4. Unfortunately, due to the generality of the Embedding problem it turns out to be very tricky to give an OR-composition algorithm directly for the Embedding problem. To overcome this problem, we introduce a restricted version of Embedding, which we call Independent Embedding, for which an OR-composition algorithm is much easier to find and whose unparameterized version is still **NP**-complete, as we prove below.

Let $\mathcal{I}_k = (I_k, \leq^{\mathcal{I}_k})$ be the poset that has k mutually incomparable chains consisting of three elements each. Then the Independent Embedding problem is defined as follows.

Independent Embedding **Parameter:** width($\mathcal{P}$), k
Input: A poset $\mathcal{P} = (P, \leq^P)$ and a natural number k.
Question: Is there an embedding from $\mathcal{I}_k$ to $\mathcal{P}$?

NP-completeness of Independent Embedding follows straightforwardly from **NP**-completeness of the ordinary independent set problem on graphs. As to an OR-composition algorithm for Independent Embedding, the other ingredient in Proposition 2.4, we do roughly as follows: we first align a given collection of instances to the same (maximum) value of the parameter k, and then we "stack" these instances on top of one another (all elements of a lower instance are "$\leq^P$" than all those of a higher instance), making a combined instance of Independent Embedding which is an OR-composition of all the input instances and whose width does not exceed the maximum of their widths.

Here we formulate these two claims without formal proofs, which can be found in the full version of the paper [9].

Lemma 5.1. Independent Embedding *is* **NP***-complete.*

Lemma 5.2. Independent Embedding *does not have a polynomial kernel unless* co**NP** ⊆ **NP**/poly*.*

We are now ready to summarize the main result of this section:

Theorem 5.3. Embedding*,* Poset ∃-FO-Model Checking *and* Poset FO-Model Checking *have no polynomial kernel unless* co**NP** ⊆ **NP**/poly*.*

6 Conclusions

Besides establishing tractability of existential FO model checking on posets of bounded width, the authors of [4] also considered several other poset invariants, giving (in-)tractability results for existential FO model checking for these variants. This makes, together with our simplification of proof of their main result, the parameterized complexity of the existential FO model checking on posets rather well understood.

The main direction for further research, suggested already in [4], is the parameterized complexity of model checking of full FO logic on restricted classes of posets, especially on posets of bounded width. This problem is challenging, because currently known techniques for establishing tractability of FO model checking are based on locality of FO and cannot be applied easily to posets—transitivity of $\leq$ causes that, typically, the whole poset is in a small neighbourhood of some element. On the other hand, attempts to evaluate an FO formula on a Hasse diagram (i.e., on the graph of the cover relation of a poset) fail precisely because of locality of FO.

References

1. Alon, N., Gutin, G., Kim, E., Szeider, S., Yeo, A.: Solving MAX-r-SAT above a tight lower bound. Algorithmica **61**(3), 638–655 (2011)
2. Bodlaender, H.L.: Kernelization: New Upper and Lower Bound Techniques. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 17–37. Springer, Heidelberg (2009)
3. Bodlaender, H.L., Downey, R., Fellows, M., Hermelin, D.: On problems without polynomial kernels. J. Comput. System Sci. **75**(8), 423–434 (2009)
4. Bova, S., Ganian, R., Szeider, S.: Model checking existential logic on partially ordered sets. In: CSL-LICS 2014 Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), Article No. 21. ACM, New York (2014), http://dx.doi.org/10.1145/2603088.2603110, ISBN: 978-1-4503-2886-9
5. Cai, L., Chen, J., Downey, R., Fellows, M.: Advice classes of parameterized tractability. Ann. Pure Appl. Logic **84**(1), 119–138 (1997)
6. Downey, R., Fellows, M.: Parameterized complexity. Monographs in Computer Science. Springer (1999)
7. Felsner, S., Raghavan, V., Spinrad, J.: Recognition algorithms for orders of small width and graphs of small dilworth number. Order **20**(4), 351–364 (2003)
8. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer (2006)
9. Gajarský, J., Hliněný, P., Obdržálek, J., Ordyniak, S.: Faster existential FO model checking on posets. http://arxiv.org/abs/1409.4433 (2014)
10. Grohe, M., Kreutzer, S., Siebertz, S.: Deciding first-order properties of nowhere dense graphs. In: STOC 2014, pp. 89–98. ACM (2014)
11. Jeavons, P., Cohen, D., Gyssens, M.: Closure properties of constraints. J. ACM **44**(4), 527–548 (1997)
12. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Ser. Math. Appl. OUP (2006)