The Descriptive Complexity of Relation Modification Problems

Florian Chudigiewitsch
Institute for Theoretical Computer Science, Universität zu Lübeck, Germany

Marlene Gründel
Algorithms and Complexity Group, TU Wien, Austria

Christian Komusiewicz
Institute of Computer Science, Friedrich Schiller University Jena, Germany

Nils Morawietz
Institute of Computer Science, Friedrich Schiller University Jena, Germany
LaBRI, Université de Bordeaux, Talence, France

Till Tantau
Institute for Theoretical Computer Science, Universität zu Lübeck, Germany

Abstract

A relation modification problem gets a logical structure and a natural number k as input and asks whether k modifications of the structure suffice to make it satisfy a predefined property. We provide a complete classification of the classical and parameterized complexity of relation modification problems – the latter w. r. t. the modification budget k – based on the descriptive complexity of the respective target property. We consider different types of logical structures on which modifications are performed: Whereas monadic structures and undirected graphs without self-loops each yield their own complexity landscapes, we find that modifying undirected graphs with self-loops, directed graphs, or arbitrary logical structures is equally hard w. r. t. quantifier patterns.

Moreover, we observe that all classes of problems considered in this paper are subject to a strong dichotomy in the sense that they are either very easy to solve (that is, they lie in para-$\mathrm{AC}^{0\uparrow}$ or TC^0) or intractable (that is, they contain W[2]-hard or NP-hard problems).

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Theory of computation → Complexity theory and logic; Theory of computation → Circuit complexity; Theory of computation → Problems, reductions and completeness

Keywords and phrases graph problems, descriptive complexity, edge modification, parameterized complexity, circuit complexity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.93

Related Version *Full Version*: `https://arxiv.org/abs/2603.22043` [28]

Funding *Marlene Gründel*: Supported by the Austrian Science Fund (FWF, projects 10.55776/Y1329 and 10.55776/COE12).
Nils Morawietz: Partially supported by the French ANR, project ANR-22-CE48-0001 (TEMPOGRAL).

1 Introduction

Approaching computational problems by examining their *distance to triviality* [43] has greatly enhanced our theoretical understanding of their inherent complexities and inspired practical algorithm engineering in recent years. To accomplish the latter, one might ask how much (and at which costs) a given input structure needs to be modified so that the resulting structure has a specific – more or less "trivial" – property that can be algorithmically exploited. Different variants of distances between structures are conceivable, depending on the type of

51st International Symposium on Mathematical Foundations of Computer Science (MFCS 2026).
Editors: Michal Koucký and Daniela Petrişan; Article No. 93; pp. 93:1–93:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

modification permitted: For instance, if the input to a problem is given as a *logical structure* consisting of a universe and relations of arbitrary arity, we can allow to *add* tuples to relations (known as the *addition* or *completion problem*), to *delete* tuples (the *deletion problem*), or allow both (the *editing problem*).

Historically, the study of these meta-problems has been dominated by the special case where the input structure describes a graph and tuples correspond to *edges*. The interest in such *edge modification problems* [64, 65] stems from the fact that a variety of well-studied problems in theoretical computer science can be formulated as such. For instance, the problem of *cluster deletion* (*cluster editing*, respectively) [46, 49] asks whether we can delete (add or delete, resp.) at most k edges of an input graph to obtain a cluster graph, that is, a disjoint union of cliques. Similarly, the problem of *triangle deletion* [20, 60] tasks us to delete at most k edges to make a graph triangle-free. As another example, the *feedback arc set* problem [51] asks whether deleting at most k edges suffices to remove all cycles from a given directed graph. There are further applications of graph modification problems in related disciplines such as machine learning [12], network analysis [38, 63], sociology [53, 55], databases [39], computational biology [16, 15, 59], computational physics [13], and engineering [44, 26, 14, 48]. Analyzing the underlying meta-problem of graph modification towards a certain property instead of the aforementioned individual problems offers the obvious advantage of enabling solution procedures to be formulated as generally (and thus reusable) as possible.

While for the closely related meta-problem of *vertex modification*, polynomial-time solvability has been already precisely characterized for hereditary properties in 1980 by Lewis and Yannakakis [52], analyzing the computational complexity of edge modification problems proved to be more persistent and has been the subject of numerous publications in the last decades, many of which can be found in the surveys [21, 56, 58]. Since modification problems are equipped with a natural modification budget k, they are particularly suited to be studied through the lens of *parameterized complexity* [31, 30], where one analyzes the running time of algorithms not only in terms of the input size $|I|$, but also with respect to a numerical parameter k. From this perspective, too, the complexity of graph modification problems has already been extensively investigated. For an overview, we refer the reader to the recent survey of Crespelle et al. [29].

One way to systematically perform the computational complexity analysis of edge modification problems is to group problems according to the *descriptive complexity* [47, 42] of the targeted graph class. This approach is rooted in the hope that if a graph class is easy to describe, then the corresponding edge modification problem, which attempts to transform a given graph into a graph within that class, may also be easy to solve. A natural way to measure the descriptive complexity of a property is to analyze the *quantifier pattern* that is needed to express this property in a certain logic. Börger et al. initiated this line of research by providing a complete classification of decidable fragments of first-order logic based on their quantifier pattern [19]. Subsequent research pursued a complexity-theoretic approach and explored to which extent quantifier patterns of second-order problem formulations affect the classical [33, 40, 62] and parameterized complexity [5] of these problems.

Most pertinent to our work, Fomin et al. [36] investigated the parameterized complexity of vertex and edge modification problems based on the number of *quantifier alternations* in the first-order formalization of the targeted property. With respect to edge modification, they found that Σ_2-formulas (first-order formulas starting with existential quantifiers, followed by universal quantifiers) can only describe *fixed-parameter tractable* (*FPT*) problems – that is, problems that can be solved by algorithms that run in time $f(k) \cdot |I|^{\mathcal{O}(1)}$, where f is a computable function. On the other hand, Π_2-formulas (formulas starting with universal quantifiers, followed by existential quantifiers) can express problems that are W[2]-hard.

Our Setting. In this paper, we provide a complete and extensive classification of the classical and parameterized complexity of edge deletion, edge addition, and edge editing problems based on the first-order quantifier patterns of the targeted properties. We go beyond the problem setting proposed in [36] by considering three different types of input graphs: *basic graphs* (occasionally also called *simple graphs*) which are undirected and have no self-loops, *undirected graphs,* and *directed graphs.* Since we do observe differences in the complexity landscapes across these various input settings, we set out to examine the input-sensitivity of modification problems more rigorously by allowing arbitrary logical structures as inputs. We refer to these meta-problems as *relation modification problems.*

In Table 1, we provide an overview of our main results where the following notations are used: For a fixed first-order formula ϕ over a vocabulary $\tau = \{R_1, \dots, R_r\}$, the problem $\textsc{Relation-Modification}^{\mathrm{del}}_{\mathrm{arb}}(\phi)$ (abbreviated $\mathrm{RM}^{\mathrm{del}}_{\mathrm{arb}}(\phi)$) asks us to tell on input of a logical structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$ and a natural number k whether there is a tuple $D = (D_1, \dots, D_r)$ with $D_i \subseteq R_i^{\mathcal{A}}$ for all $i \in \{1, \dots, r\}$ and $\|D\| = \sum_{i=1}^{r} |D_i| \le k$, whose removal from $\mathcal{A}$ yields a structure $\mathcal{A}'$ for which $\mathcal{A}' \models \phi$ holds. We denote with $\mathrm{RM}^{\mathrm{add}}_{\mathrm{arb}}(\phi)$ the related problem that asks whether we can add up to k tuples such that the new structure satisfies ϕ. Finally, the problem $\mathrm{RM}^{\mathrm{edit}}_{\mathrm{arb}}(\phi)$ allows for the addition and deletion of tuples. For a modification operation $\otimes \in \{\mathrm{del}, \mathrm{add}, \mathrm{edit}\}$, we denote with $\mathrm{RM}^{\otimes}_{\mathrm{dir}}(\phi)$ ($\mathrm{RM}^{\otimes}_{\mathrm{undir}}(\phi)$, and $\mathrm{RM}^{\otimes}_{\mathrm{basic}}(\phi)$, respectively) the restriction of the modification problem to inputs and target structures that are directed graphs (undirected graphs, and basic graphs, respectively). Further, let $\mathrm{RM}^{\otimes}_{\mathrm{mon}}(\phi)$ be the problem restricted to input structures that contain only monadic relation symbols. For undirected and basic graphs $G = (V, E)$, where we simply have $D = (D_1)$ with $D_1 \subseteq E$ in the case of deletion (and $D_1 \subseteq V \times V$ in the cases of adding and editing), we use the more natural measure $\|D\| = |\{\{u, v\} \mid (u, v) \in D_1\}|$. We assume throughout that the given first-order formula ϕ is in *prenex normal form*, that is, all quantifiers that occur in ϕ are pushed to the beginning of the formula. The precise sequence of universal and existential quantifiers at the beginning of ϕ is captured by a *(first-order) quantifier pattern* $p \in \{a, e\}^*$. For instance, the formula $\phi = \forall u \forall v \exists y \big(u = v \lor u \sim v \lor (u \sim y \land y \sim v)\big)$ has the quantifier pattern $p = aae$. For all modification operations $\otimes \in \{\mathrm{del}, \mathrm{add}, \mathrm{edit}\}$ and all structure types $\mathrm{T} \in \{\mathrm{arb}, \mathrm{dir}, \mathrm{undir}, \mathrm{basic}, \mathrm{mon}\}$ we let $\mathrm{RM}^{\otimes}_{\mathrm{T}}(p)$ denote the class of all problems $\mathrm{RM}^{\otimes}_{\mathrm{T}}(\phi)$ where the fixed formula ϕ exhibits the quantifier pattern p. Finally, for all problem variants $\mathrm{RM}^{\otimes}_{\mathrm{T}}(\phi)$, we denote with $\text{p-}\mathrm{RM}^{\otimes}_{\mathrm{T}}(\phi)$ the parameterized version of this problem that treats the modification budget k as the parameter.

A complete complexity classification of tuple deletion, addition, and editing problems for first-order formulas depending on the quantifier pattern $p \in \{a, e\}^*$ (where $p \preceq q$ means that p is a subsequence of q) is given in Table 1. Note that $\text{para-AC}^{0\uparrow} \subseteq \text{para-P} = \mathrm{FPT} \subseteq \mathrm{W}[2]$ holds and that it is a standard assumption that both $\text{para-AC}^{0\uparrow} \subsetneq \mathrm{FPT}$ and $\mathrm{FPT} \subsetneq \mathrm{W}[2]$ are true as well. Below, we briefly summarize the main contributions of this paper.

Our Contributions. As a first set of contributions, we *refine* the dichotomy of the *parameterized complexity of edge modification problems* from [36] in multiple ways:

1. First, we resolve the following question, which is central to understanding the descriptive features that determine the computational complexity of edge modification problems: Is it the *number of alternations* of quantifiers that solely drives the transition from tractability to intractability, or is this boundary determined by *short patterns* that happen to exhibit a specific number of alternations? As we show, the latter is indeed the case. All intractability results already hold for very short and simple patterns. While it was previously known that an intractable problem can be defined by a formula in Π_2 (more specifically, one with the quantifier pattern *aeeee*), we find that already the simplest interesting pattern in Π_2, namely *ae*, suffices to establish intractability.

Table 1 Complete classification of the classical (above) as well as the parameterized (below) complexity of relation modification problems. For the latter, the modification budget is treated as the parameter. We abbreviate "$p_1 \preceq p$ or $p_2 \preceq p$" as "$p_1, p_2 \preceq p$".

$\mathrm{RM}^{\otimes}_{\mathrm{undir}}(p)$, $\mathrm{RM}^{\otimes}_{\mathrm{dir}}(p)$, and $\mathrm{RM}^{\otimes}_{\mathrm{arb}}(p)$	$\subseteq \mathrm{AC}^0$, when	$p \preceq e^*$.
	$\not\subseteq \mathrm{AC}^0$ but $\subseteq \mathrm{TC}^0$, when	$a \preceq p \preceq e^*a$.
	$\cap$ NP-hard $\neq \emptyset$, when	$aa,\ ae \preceq p$.
$\mathrm{RM}^{\otimes}_{\mathrm{basic}}(p)$	$\subseteq \mathrm{AC}^0$, when	$p \preceq e^*,\ a$.
	$\not\subseteq \mathrm{AC}^0$ but $\subseteq \mathrm{TC}^0$, when	$ea,\ ae,\ aa \preceq p \preceq e^*a,\ aa,\ ae$.
	$\cap$ NP-hard $\neq \emptyset$, when	$aea,\ aee,\ aae,\ eae,\ eaa,\ aaa \preceq p$.
$\mathrm{RM}^{\otimes}_{\mathrm{mon}}(p)$	$\subseteq \mathrm{AC}^0$, when	$p \preceq e^*$.
	$\not\subseteq \mathrm{AC}^0$ but $\subseteq \mathrm{TC}^0$, when	$a \preceq p$.
p-$\mathrm{RM}^{\otimes}_{\mathrm{undir}}(p)$, p-$\mathrm{RM}^{\otimes}_{\mathrm{dir}}(p)$, and p-$\mathrm{RM}^{\otimes}_{\mathrm{arb}}(p)$	$\subseteq$ para-$\mathrm{AC}^{0\uparrow}$, when	$p \preceq e^*a^*$.
	$\cap$ W[2]-hard $\neq \emptyset$, when	$ae \preceq p$.
p-$\mathrm{RM}^{\otimes}_{\mathrm{basic}}(p)$	$\subseteq$ para-$\mathrm{AC}^{0\uparrow}$, when	$p \preceq e^*a^*,\ ae$.
	$\cap$ W[2]-hard $\neq \emptyset$, when	$aea,\ aee,\ aae,\ eae \preceq p$.
p-$\mathrm{RM}^{\otimes}_{\mathrm{mon}}(p)$	$\subseteq$ para-AC^0.	

2. We *improve existing upper bounds* by showing that all classes of modification problems that are (fixed-parameter) tractable are even in para-$\mathrm{AC}^{0\uparrow}$. From an algorithmic perspective, this implies that these problems admit efficient *parallel fixed-parameter algorithms.* From a logical perspective, this insight establishes an interesting dichotomy in which all considered problem classes are either very easy to solve or contain intractable problems.
3. We identify *new intractable problems* that are of independent interest, such as p-Edge-Addition To Radius r and p-Edge-Editing To Radius r for all $r \geq 2$.
4. We contrast the parameterized study of edge modification towards first-order properties by a systematic investigation of their *classical complexities.* Here again, we gain valuable insights by pursuing our fine-grained pattern-based approach: For instance, with respect to basic graphs, we find that both Σ_2 and Π_2 contain short tractable (and non-trivial) fragments, whereas a purely alternation-based study would label the whole class of formulas in Π_1 as intractable.

Our results complement the complexity landscape of *vertex deletion problems* obtained by Bannach et al. [6]. In particular, we find that the parameterized complexity of edge modification problems is considerably higher: While for directed graphs, vertex deletion towards all formulas that exhibit the pattern $e^*a^*e^*$ is tractable, we show that with respect to edge deletion, there are already intractable problems describable by formulas with pattern ae.

Going beyond edge modification problems, this paper is the first to observe that the complexity of *relation modification problems* is *sensitive towards the type of logical structures* that are allowed as input and targeted property. Both in the parameterized and the classical setting, we observe a complexity jump when transitioning from basic graphs to the more general undirected graphs. But surprisingly, all further generalizations, that is, allowing directed graphs or even arbitrary logical structures do not increase the complexity of the respective problems further. To the other end, if the vocabulary contains only monadic symbols, the problem is always tractable.

Lastly and contrary to what one might expect, we show that the *complexities of tuple deletion, tuple addition, and tuple editing coincide* for all considered fragments.

Further Related Work. Besides the lines of work presented above, the interplay of descriptive and parameterized complexity has attracted further recent interest. Fomin et al. [37], for example, investigated the parameterized complexity of graph modification to first-order formulas, with respect to modifications within a certain *elimination distance.* Moreover, the complexities of a large family of graph modification operations containing, for example, vertex and edge deletion, edge contraction, and independent set deletion, were recently considered by Morelle et al. [57] for target classes that are minor-closed.

The problems we address in this paper have also been studied in the context of *hereditary model checking in first-order logic.* Bodirsky et al. recently characterized the prefix classes for which hereditary first-order model checking is tractable [17]. The complexity of this problem is in turn connected to the complexity of extensional existential second-order logic [18].

Another related field of research studies *database repairs* [2], where both deleting [27] and adding tuples [22] have been addressed. Although these works also investigate the complexity for selected problems, they do not rigorously examine the modification of tuples in a broader sense. *Consistent query answering (CQA)* is a related problem that aims to ensure that retrieved query answers are consistent with some specifications. The classical complexity of this problem was investigated in [61] and its parameterized complexity in [54]. Recently, a complexity tetrachotomy of CQA with certain constraints was established in [50].

Parallel Parameterized Algorithms were first investigated in [23]. Elberfeld et al. [34] subsequently provided a general framework for parallel parameterized complexity classes. As a major result, Chen and Flum showed by a parameterized circuit lower bound that CLIQUE parameterized by the solution size is not in para-AC^0 [25]. Subsequent work [8, 9, 10, 11, 7, 5] pursued this line of research further, with one of the most surprising results being that the problem HITTING SET on hypergraphs of constant edge cardinality is in para-AC^0 when parameterized by the solution size. The uniformity of parameterized circuit classes was thoroughly studied by Hegeman et al. [45] and practical implementations of parallel parameterized algorithms on parallel random access machines were designed in [1, 24].

Organization of this Paper. We review basic concepts and terminology from Finite Model Theory and Computational Complexity in Section 2, and then present the complexity-theoretic classification of relation modification problems for all structure variants in Section 3 for the parameterized, and in Section 4 for the classical setting. Proofs and paragraphs that only appear in the full version [28] due to space constraints are replaced by (▼) in the main text.

2 Descriptive, Parameterized, and Circuit Complexity

Terminology from Finite Model Theory. We use standard terminology from finite model theory. For a more thorough introduction, see [32]. A *relational vocabulary* τ (also known as a *signature*) is a set of *relation symbols* to each of which we assign a positive *arity*, denoted by a superscript. For example, $\tau = \{P^1, E^2\}$ is a relational vocabulary with a unary relation symbol P and a binary relation symbol E. In a *monadic* vocabulary, each relation symbol has arity one. A *finite τ-structure* $\mathcal{A}$ consists of a *finite universe* A and for each of the r relation symbols $R_i \in \tau, i \in \{1, \dots, r\}$ with arity a_i of a relation $R_i^{\mathcal{A}} \subseteq A^{a_i}$. We denote the set of *finite τ-structures* as $\mathrm{STRUC}[\tau]$. We assume the structures are encoded in a reasonable way, see for example [41, page 136]. We further assume all structures to be ordered.

First-order τ-formulas are defined in the usual way. For a τ-formula ϕ with free variables $x_1, \dots, x_\ell$, a structure $\mathcal{S} \in \text{STRUC}[\tau]$ with universe S, and $s_1, \dots, s_\ell \in S$ we write $\mathcal{S} \models \phi[s_1, \dots, s_\ell]$ to denote that $\mathcal{S}$ is a model of ϕ when the free variables are interpreted as the s_i. If ϕ has no free variables, we write $\text{MODELS}(\phi)$ for the class of finite models of ϕ. A *decision problem* P is a subset of $\text{STRUC}[\tau]$ that is closed under isomorphisms. A formula ϕ *describes* P if $\text{MODELS}(\phi) = P$.

For τ-structures $\mathcal{A}$ and $\mathcal{B}$ with universes A and B, respectively, we say that $\mathcal{A}$ is an *induced substructure* of $\mathcal{B}$ if $A \subseteq B$ and for all r-ary $R \in \tau$, we have $R^{\mathcal{A}} = R^{\mathcal{B}} \cap A^r$. For a set $S \subseteq B$, we denote by $\mathcal{B} \setminus S$ the substructure induced by $B \setminus S$.

We regard *directed* graphs $G = (V, E)$ (which are pairs of a nonempty vertex set V and an edge relation $E \subseteq V \times V$) as logical structures $\mathcal{G}$ over the vocabulary $\tau_{\text{digraph}} = \{\sim^2\}$ where V is the universe and $\sim^{\mathcal{G}} = E$. An *undirected* graph is a directed graph that additionally satisfies $\phi_{\text{undirected}} := \forall x \forall y (x \sim y \to y \sim x)$, while a *basic* graph satisfies $\phi_{\text{basic}} := \forall x \forall y \big(x \sim y \to (y \sim x \land x \neq y)\big)$. We will use $u \multimap\!\!\!\circ\, v$ as a shorthand for $\{(u, v), (v, u)\} \subseteq E$.

For a first-order formula in prenex normal form (meaning all quantifiers are at the front), we can associate a *quantifier prefix pattern* (or *pattern* for short), which are words over the alphabet $\{e, a\}$. For example, the formula ϕ_{basic} has the pattern aa, while the formula $\phi_{\text{degree}\geq 2} := \forall x \exists y_1 \exists y_2 \big((x \sim y_1) \land (x \sim y_2) \land (y_1 \neq y_2)\big)$ has the pattern aee. As another example, the formulas in the class Π_2 (which start with a universal quantifier and have one alternation) are exactly the formulas with a pattern $p \in \{a\}^* \circ \{e\}^*$, which we write briefly as $p \in a^* e^*$. We write $p \preceq q$ if p is a subsequence of q. For instance, $aaee \preceq aeaeae$.

When we consider budgets for the modification of logical structures $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$, we have several degrees of freedom: First, we can specify that modifications are only allowed with respect to a single relation, or we can allow to modify all relations of the input structure. We call the first variant *uniform modification* and the second variant *unrestricted modification*. In this paper, we consider unrestricted modification, but address uniform modification in the conclusion. Second, we can choose whether we count each tuple we add or delete (which is the more natural measure for directed graphs and arbitrary structures), or only the number of different sets of elements on which we modify relations (this is the more natural measure on basic and undirected graphs). For directed graphs and arbitrary structures, we will use the former measure, whereas for basic and undirected graphs, we will use the latter. However, we will discuss in the conclusion that applying the respective opposite counting version yields the same complexity landscape.

Let $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$ be a logical structure where R_i has arity a_i for all $i \in \{1, \dots, r\}$. A *modulator* $S = (S_1, \dots, S_r)$ contains sets $S_i \subseteq A^{a_i}$. If we have $S_i \subseteq R_i^{\mathcal{A}}$ for all $i \in \{1, \dots, r\}$, we call S a *deletion modulator*, and if we have $S_i \cap R_i^{\mathcal{A}} = \emptyset$ for all $i \in \{1, \dots, r\}$, we call S an *addition modulator*. We write

$$\|S\| := \begin{cases} \sum_{i=1}^{r} \big|\{\{u, v\} \mid (u, v) \in S_i\}\big| & \mathcal{A} \text{ is an undirected or basic graph,} \\ \sum_{i=1}^{r} |S_i| & \text{otherwise.} \end{cases}$$

Now, for a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$ and a modulator $S = (S_1, \dots, S_r)$ we define $\mathcal{A} \triangleright S := (A, R_1^{\mathcal{A}} \triangle S_1, \dots, R_r^{\mathcal{A}} \triangle S_r)$, where $A \triangle B$ denotes the symmetric difference between sets A and B. To illustrate these notions, let us consider the special case of graphs. For a basic or undirected graph $G = (V, E)$, the solution for the deletion problem is simply a set $S \subseteq E$ and for the addition or editing problem it is a set $S \subseteq V \times V$.

Relation modification problems. We study various relation modification problems. As an introductory example, consider the following problem where ϕ is a first-order τ-formula.

▶ **Problem 2.1** ($\mathrm{RM}^{\mathrm{edit}}_{\mathrm{arb}}(\phi)$).
Instance: *A logical structure* $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$ *and a number* $k \in \mathbb{N}$.
Question: *Is there a modulator* S *with* $\|S\| \leq k$ *such that* $\mathcal{A} \triangleright S \models \phi$*?*

Analogously, we define $\mathrm{RM}^{\mathrm{add}}_{\mathrm{arb}}(\phi)$ ($\mathrm{RM}^{\mathrm{del}}_{\mathrm{arb}}(\phi)$, respectively) to be the question of whether there is an addition (a deletion, resp.) modulator S with $\|S\| \leq k$ such that $\mathcal{A} \triangleright S \models \phi$.

We define restrictions on the allowed input and target structure by considering problems $\mathrm{RM}^{\otimes}_{\mathrm{dir}}(\phi)$, where $\otimes \in \{\mathrm{add}, \mathrm{del}, \mathrm{edit}\}$ and the vocabulary of ϕ consists only of a single binary relation E, thus forcing input and target structures to be directed graphs. Similarly, we denote with $\mathrm{RM}^{\otimes}_{\mathrm{basic}}(\phi) = \mathrm{RM}^{\otimes}_{\mathrm{dir}}(\phi \wedge \phi_{\mathrm{basic}}) \cap \textsc{models}(\phi_{\mathrm{basic}})$ the meta-problem for which the input and target structures are basic graphs, and $\mathrm{RM}^{\otimes}_{\mathrm{undir}}(\phi) = \mathrm{RM}^{\otimes}_{\mathrm{dir}}(\phi \wedge \phi_{\mathrm{undir}}) \cap \textsc{models}(\phi_{\mathrm{undir}})$ forces input and target structures to be undirected graphs. For a pattern $p \in \{a, e\}^*$, the class $\mathrm{RM}^{\otimes}_{\mathrm{arb}}(p)$ contains all problems $\mathrm{RM}^{\otimes}_{\mathrm{arb}}(\phi)$ such that ϕ has pattern p. The classes with the subscripts "basic", "undir" and "dir" are defined similarly.

Fomin et al. [36] observed that for each class of formulas with the same quantifier alternation, the complexity of edge deletion and edge addition coincides on basic graphs. Note that the same holds for tuples on arbitrary logical structures by the following observation: For a structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$ let $\overline{\mathcal{A}} = (A, A^{a_1} \setminus R_1^{\mathcal{A}}, \dots, A^{a_r} \setminus R_r^{\mathcal{A}})$ denote its complement structure, and for a formula ϕ, let $\overline{\phi}$ denote the formula obtained from ϕ by replacing every occurrence of $R_i(x_1, \dots, x_{a_i})$ by $\neg R_i(x_1, \dots, x_{a_i})$ for arbitrary structures, directed and undirected graphs, and by replacing every occurrence of $(x \sim y)$ by $\neg(x = y) \wedge \neg(x \sim y)$ when we restrict the inputs to basic graphs. Then it is not hard to verify the following.

▶ **Observation 2.2.** *For every first-order formula* ϕ *and every type* $\mathrm{T} \in \{\mathrm{arb}, \mathrm{dir}, \mathrm{undir}, \mathrm{basic}\}$*,* $(\mathcal{A}, k)$ *is a yes-instance of* $\mathrm{RM}^{\mathrm{add}}_{\mathrm{T}}(\phi)$ *if and only if* $(\overline{\mathcal{A}}, k)$ *is a yes-instance of* $\mathrm{RM}^{\mathrm{del}}_{\mathrm{T}}(\overline{\phi})$.

We use this observation in subsequent sections by interchangeably employing relation deletions or additions, depending on which version is best suited for a concise presentation.

Terminology from (Parameterized) Complexity. In order to establish tractability in the classical setting, we show that the respective problems are contained in the class AC^0, which is the class of problems that can be defined by a family of unbounded fan-in circuits $(C_n)_{n \in \mathbb{N}}$ with constant depth, size $n^{O(1)}$, and and-, or-, and not-gates, or in the class TC^0, which is defined analogously, but where circuits can additionally use threshold-gates. Although questions of uniformity will not be addressed further, we remark that DLOGTIME uniformity can be achieved in all relevant proofs. Moreover, recall that $\mathrm{TC}^0 \subseteq \mathrm{L} \subseteq \mathrm{P}$ holds.

We use standard definitions from parameterized complexity, see for instance [30, 31, 35]. A *parameterized problem* is a set $Q \subseteq \Sigma^* \times \mathbb{N}$ for an alphabet Σ. Given an *instance* $(x, k) \in \Sigma^* \times \mathbb{N}$, we call x the *input* and k the *parameter*. We denote the parameterized version of a specific problem by adding the prefix p-, so, for example, $\mathrm{RM}^{\mathrm{del}}_{\mathrm{dir}}(\phi)$ becomes p-$\mathrm{RM}^{\mathrm{del}}_{\mathrm{dir}}(\phi)$. Similarly to the classical setting, we will establish the tractability of a parameterized problem by proving its containment in a certain parameterized circuit class, known as para-$\mathrm{AC}^{0\uparrow}$: Formally, para-$\mathrm{AC}^{0\uparrow}$ is defined as the class of all parameterized problems that can be decided by a family of unbounded fan-in circuits $(C_{n,k})_{n,k \in \mathbb{N}}$ that may use and-, or-, and not-gates, and have depth $f(k)$ and size $f(k) \cdot n^{O(1)}$ for some computable function f. The class para-AC^0 is defined analogously with the additional restriction that the depth of the circuit must be in $O(1)$. Note that para-AC^0-circuits can simulate threshold-gates when the threshold is the parameter [8, Lemma 3.3]. It holds that para-$\mathrm{AC}^0 \subseteq$ para-$\mathrm{AC}^{0\uparrow} \subseteq$ para-$\mathrm{P} = \mathrm{FPT}$, for a suitable notion of uniformity, see [8]. We use AC^0-many-one reductions in the classical setting, and para-AC^0-many-one reductions in the parameterized setting. Both are very weak notions of reducibility, which has the benefit of yielding strong hardness results.

3 Parameterized Complexity of Relation Modification Problems

In this section, we will address the parameterized complexity of relation modification problems based on the first-order quantifier patterns of the targeted properties.

3.1 Undirected Graphs, Directed Graphs, and Arbitrary Structures

We first establish the tractability landscapes of relation modification problems with respect to undirected and directed graphs and arbitrary structures. More specifically, we will show that these landscapes coincide and exhibit the following complexities.

▶ **Theorem 3.1.** *For each modification operation* $\otimes \in \{\text{del}, \text{add}, \text{edit}\}$*, we have*

1. $\text{p-RM}^{\otimes}_{\text{arb}}(p) \subseteq \text{para-AC}^{0\uparrow}$*, if* $p \preceq e^*a^*$ *and*
2. $\text{p-RM}^{\otimes}_{\text{undir}}(p)$ *and* $\text{p-RM}^{\otimes}_{\text{dir}}(p)$ *both contain a* W[2]*-hard problem, if* $ae \preceq p$*.*

We prove this theorem by individually establishing the respective upper and lower bounds. Note that the upper bound on arbitrary logical structures naturally carries over to the more restricted structures of basic, undirected, and directed graphs. Conversely, the hardness results for undirected and directed graphs imply hardness of more general structures.

Upper Bound. We establish the tractability of the fragment e^*a^* for all parameterized problem versions by proving the following lemma.

▶ **Lemma 3.2.** *For* $\otimes \in \{\text{del}, \text{add}, \text{edit}\}$*, we have* $\text{p-RM}^{\otimes}_{\text{arb}}(e^*a^*) \subseteq \text{para-AC}^{0\uparrow}$*.*

Proof. We generalize a construction given by Fomin et al. [36] such that it (a) handles arbitrary logical structures, and (b) is parallelized, thus showing the containment in para-AC$^{0\uparrow}$.

For $\text{p-RM}^{\text{edit}}_{\text{arb}}(\phi) \in \text{p-RM}^{\text{edit}}_{\text{arb}}(e^*a^*)$, let ϕ be a first-order τ-formula over a vocabulary $\tau = \{R_1, \dots, R_r\}$, where relation R_i has arity a_i and let furthermore ϕ be of the form $\exists x_1 \cdots \exists x_c \forall y_1 \cdots \forall y_d(\psi)$, where ψ is a quantifier-free formula and $c, d \in \mathbb{N}$. We aim for an algorithm that, given a logical structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$ and $k \in \mathbb{N}$, decides in para-AC$^{0\uparrow}$ whether there is a modulator S with $\|S\| \leq k$ such that $\mathcal{A} \triangleright S \models \phi$. We start by observing the following:

Let $v_1, \dots, v_c$ be a sequence of elements from the universe A. Then, we either have $\mathcal{A} \models (\forall y_1 \cdots \forall y_d(\psi))[v_1, \dots, v_c]$ (in which case we accept the instance) or we have $\mathcal{A} \not\models (\forall y_1 \cdots \forall y_d(\psi))[v_1, \dots, v_c]$. In the latter case, there must be a sequence $u_1, \dots, u_d$ of elements from the universe A such that the structure induced by the union of $v_1, \dots, v_c$ and $u_1, \dots, u_d$ violates the assertions made by ψ. If we can edit at most k tuples to resolve all violations on the structure induced by $v_1, \dots, v_c$ and every violating assignment to the universal variables, then $\mathcal{A} \triangleright S \models \phi$. Otherwise, we can rule out $v_1, \dots, v_c$ as an assignment to $x_1, \dots, x_c$.

Based on this observation, we construct the following algorithm: Consider all of the $|A|^c$ possible assignments $(v_1, \dots, v_c)$ to the variables $x_1, \dots, x_c$ in parallel. For each, we in turn consider all of the $2^{\sum_{i=1}^{r} c^{a_i}}$ possible relations on the structure induced by $\{v_1, \dots, v_c\}$ in parallel. For each such set of relations, there is a modulator $S^{(0)}$ such that $\mathcal{A} \triangleright S^{(0)}$ yields precisely this set of relations. By definition, $S^{(0)}$ is a tuple in which each entry is a set of tuples over (subsets of) $\{v_1, \dots, v_c\}$, hence, its size is constantly bounded. Now, fix a certain set of relations on $\{v_1, \dots, v_c\}$ and a respective modulator $S^{(0)}$. We use k layers of a Boolean circuit to find and resolve all violations that arise from assignments to the universally quantified variables. We let $S^{(0)}$ be associated with the first layer, and proceed as follows in all subsequent layers $i \in \{1, \dots, k\}$: We check whether an assignment

$(w_1, \dots, w_d)$ to the variables $y_1, \dots, y_d$ (for example, the lexicographically first) exists such that $\mathcal{A} \triangleright S^{(i)} \not\models \psi[v_1, \dots, v_c, w_1, \dots, w_d]$. If we cannot find such an assignment, we can accept the instance, since we have $\mathcal{A} \triangleright S^{(i)} \models \phi$. Otherwise, we have to edit a relation in the structure induced by the current assignment. We branch over all possible edits and associate each branch with a new modulator $S^{(i+1)}$ that extends $S^{(i)}$ by the relations we marked to be edited. If at layer k, we still find an assignment $(w_1, \dots, w_d)$ to the variables $y_1, \dots, y_d$ such that $\mathcal{A} \triangleright S^{(k)} \not\models \psi[v_1, \dots, v_c, w_1, \dots, w_d]$, then we reject the instance.

Since d, c and all the arities a_i are constants, the number of branches created at each level of the search tree is constant as well. Therefore, the total size of the search tree is at most $2^{\sum_{i=1}^{r} c^{a_i}} + g(r, c, d, a_1, \dots, a_r)^k$, where g is the constant function that describes the number of branches. The depth of the search tree is bounded by the number of tuples we can edit, which is our parameter. Hence, we get a para-$\mathrm{AC}^{0\uparrow}$-circuit. ◀

Lower Bounds. To show hardness of the fragment *ae* for various problem versions, we perform a reduction from p-Set-Cover, a well-known W[2]-hard problem [31, page 464], which we here formulate as a graph problem on the incidence graph of the input set family.[1] Recall that we write $u \multimapboth s$ to denote $\{(u, v), (v, u)\} \subseteq E$.

▶ **Problem 3.3** (p-Set-Cover).

Instance: *An undirected bipartite graph $G = (S \mathbin{\dot\cup} U, E)$ with shores S and U and a number k.*

Parameter: k

Question: *Is there a cover $C \subseteq S$ with $|C| \le k$ of U, meaning that for each $u \in U$ there is an $s \in C$ with $u \multimapboth s$?*

Our hardness results that are based on set cover reductions consist of 6 steps.

1. We first establish the hardness for the problem versions that restrict the type of structure to undirected graphs. In order to do this, we start by constructing a formula ϕ with the quantifier pattern p for which we want to establish hardness.
2. We describe a construction that transforms an arbitrary input (G, k) of p-Set-Cover into an instance (G', k') of $\text{p-RM}^{\text{del}}_{\text{undir}}(\phi)$ (or $\text{p-RM}^{\text{add}}_{\text{undir}}(\phi)$, respectively), typically by adding new vertices and edges.
3. We show that $(G, k) \in$ p-Set-Cover implies $(G', k') \in \text{p-RM}^{\text{del}}_{\text{undir}}(\phi)$ (or $(G', k') \in \text{p-RM}^{\text{add}}_{\text{undir}}(\phi)$, resp.). We call this part of the proof the *forward direction.*
4. Next, we show that $(G', k') \in \text{p-RM}^{\text{del}}_{\text{undir}}(\phi)$ (or $(G', k') \in \text{p-RM}^{\text{add}}_{\text{undir}}(\phi)$, resp.) implies $(G, k) \in$ p-Set-Cover. We call this part of the proof the *backward direction.*
5. We show that the reduction can be lifted to the problem $\text{p-RM}^{\text{edit}}_{\text{undir}}(\phi)$.
6. Finally, we show how each reduction can be adapted to provide a lower bound for $\text{p-RM}^{\otimes}_{\text{dir}}(\phi)$ with $\otimes \in \{\text{del}, \text{add}, \text{edit}\}$, and thus also for $\text{p-RM}^{\otimes}_{\text{arb}}(\phi)$.

Since the complexities of edge deletion and edge addition coincide (see Observation 2.2), it suffices to analyze one of these problem versions in our proofs. We will usually choose the one that eases the presentation and state our choice in each *reduction* paragraph.

▶ **Lemma 3.4.** *For $\otimes \in \{\text{del}, \text{add}, \text{edit}\}$, $\text{p-RM}^{\otimes}_{\text{undir}}(ae)$ contains a W[2]-hard problem.*

Proof.

The formula. Consider the formula $\phi_{3.4} = \forall x \exists y \big(\neg x \sim x \to (x \sim y \land \neg y \sim y)\big)$, which expresses the property that every vertex without a self-loop has a neighbor without a self-loop.

[1] This variant is also known as p-Red-Blue Dominating Set in Parameterized Complexity.

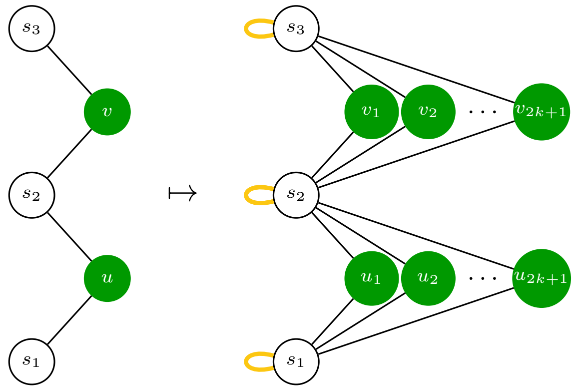

Figure 1 Visualization of the reduction in Lemma 3.4 for the graph $G = (\{s_1, s_2, s_3\} \mathbin{\dot\cup} \{u, v\}, E)$. For each element of $U = \{u, v\}$, exactly $2k+1$ vertices are added to the new graph. Each undirected edge $u \mathrel{\circ\!\!-\!\!\circ} s$ ($v \mathrel{\circ\!\!-\!\!\circ} s$, respectively) gets replaced by $2k+1$ undirected edges, namely $u_i \mathrel{\circ\!\!-\!\!\circ'} s$ ($v_i \mathrel{\circ\!\!-\!\!\circ'} s$, resp.) for all copies u_i of u (all copies v_i of v, resp.). The indicated colors, numbers, and labels are not part of the input or output. To satisfy $\phi_{3.4}$, some of the yellow edges need to be deleted. The size-1 set cover $C = \{s_2\}$ corresponds to the fact that deleting $s_2 \mathrel{\circ\!\!-\!\!\circ'} s_2$ from G' yields a graph in which each vertex without a self-loop has a neighbor without a self-loop. The same is true for the size-2 set cover $C = \{s_1, s_3\}$. In contrast, $C = \{s_1\}$ is not a set cover as v is not covered and, indeed, all neighbors of v_1 (namely s_2 and s_3) have a self-loop, if we delete neither $s_2 \mathrel{\circ\!\!-\!\!\circ'} s_2$ nor $s_3 \mathrel{\circ\!\!-\!\!\circ'} s_3$.

The reduction. We reduce p-SET-COVER to p-$\mathrm{RM}^{\mathrm{del}}_{\mathrm{undir}}(\phi_{3.4})$. Let $(S \mathbin{\dot\cup} U, E, k)$ be given. The reduction outputs $k' = k$ and the undirected graph $G' = (V', E')$ constructed as follows:

- For each $s \in S$, add s to V'. Call this set of vertices S'.
- For each $u \in U$, add $2k+1$ copies $u_1, \ldots, u_{2k+1}$ of u to V'. Call this set of vertices U'.
- Whenever $u \mathrel{\circ\!\!-\!\!\circ} s$ holds for $u \in U, s \in S$, let all u_i be adjacent to s in the new graph, that is, for $i \in \{1, \ldots, 2k+1\}$, let $u_i \mathrel{\circ\!\!-\!\!\circ'} s$.
- For every $s \in S'$ in the new graph, add the self-loop $s \mathrel{\circ\!\!-\!\!\circ'} s$.

An exemplary reduction is shown in Figure 1. Note that our construction is slightly more complex than the deletion case demands, to facilitate lifting it to the editing case later on.

Forward direction. Let $(S \mathbin{\dot\cup} U, E, k) \in$ p-SET-COVER be given. We need to show that $(G', k') \in$ p-$\mathrm{RM}^{\mathrm{del}}_{\mathrm{undir}}(\phi_{3.4})$ holds. Let $C \subseteq S$ with $|C| \leq k$ be a cover of U. Without loss of generality, assume that every vertex in C has at least one neighbor in U. Let $C' = \{(s, s) \mid s \in C\}$. We claim that $(V', E' \setminus C') \models \phi_{3.4}$, that is, by removing all self-loops from vertices $s \in C$, every vertex without a self-loop has a neighbor without a self-loop and thus the resulting graph satisfies $\phi_{3.4}$. To see this, call a vertex that has a self-loop or a neighbor without a self-loop *happy*. Naturally, if every vertex of the graph $(V', E' \setminus C')$ is happy, we have $(V', E' \setminus C') \models \phi_{3.4}$. Observe that all vertices in S' are happy, since they have at least one neighbor in U' and the vertices in U' do not have self-loops. Moreover, the vertices in U' each have at least one neighbor in S' that does not have a self-loop since we assumed C to be a cover of U. Clearly, $\|C'\| = |C| \leq k$.

Backward direction. Conversely, suppose that for $G' = (V', E')$, we are given a set $D \subseteq E'$ with $\|D\| \leq k$ such that $(V', E' \setminus D) \models \phi_{3.4}$. Observe that each $u \in U$ is only connected to vertices in S that have a self-loop. Hence, from $(V', E' \setminus D) \models \phi_{3.4}$, we conclude that for each $u_i \in U'$, there is an $s \in S'$ with $s \mathrel{\circ\!\!-\!\!\circ'} u_i$ such that $(s, s) \in D$. This means that $C = \{s \in S' \mid (s, s) \in D\}$ is a set cover of $(S \mathbin{\dot\cup} U, E)$ and $|C| \leq \|D\| \leq k$.

Edge editing. To see that the reduction applies to edge editing as well, observe that there are only two other modifications that will make vertices in U' happy: (1) Adding a self-loop to such vertices, or (2) making vertices in U' adjacent to each other, where edges may be added both between vertices that stem from the same vertex in U, and between vertices that are copies of two distinct vertices in U. However, since introducing a self-loop can only make one more vertex happy, and introducing an edge between two distinct vertices in U' can only make up to two vertices happy, these operations do not suffice to ensure the happiness of all $2k+1$ copies of u. Thus, we still have to delete the self-loop of at least one neighbor of u.

Adaptation to the directed setting. We let $\phi_{3.4,\text{dir}} = \forall x \exists y(\neg x \sim x \to (x \sim y \land \neg y \sim y))$ and employ the same reduction, except that we direct each added edge $u_i \multimap\!\circ' s$ from u_i to s. ◀

3.2 Basic Graphs

In the last section, we proved that the complexity landscapes of relation modification problems coincide on undirected graphs, directed graphs, and arbitrary structures. We now consider the restriction to basic graphs, that is, undirected graphs that have no self-loops, which, interestingly, yields a different tractability frontier than the previous problem versions.

▶ **Theorem 3.5** (▼). *For each modification operation* $\otimes \in \{\text{del}, \text{add}, \text{edit}\}$*, we have*
1. $\text{p-RM}^{\otimes}_{\text{basic}}(p) \subseteq \text{para-AC}^{0\uparrow}$*, if* $p \preceq e^*a^*$ *or* $p \preceq ae$ *and*
2. $\text{p-RM}^{\otimes}_{\text{basic}}(p)$ *contains a* $\mathrm{W}[2]$*-hard problem, if* $aea \preceq p$*,* $eae \preceq p$*,* $aee \preceq p$*, or* $aae \preceq p$*.*

3.3 Monadic Structures

To complete the complexity classification for the parameterized case, we finally turn to structures over monadic vocabularies.

▶ **Theorem 3.6.** *For each* $\otimes \in \{\text{del}, \text{add}, \text{edit}\}$*, we have* $\text{p-RM}^{\otimes}_{\text{mon}}(p) \subseteq \text{para-AC}^0$.

Proof. For a fixed monadic vocabulary $\tau = \{R_1^1, \dots, R_r^1\}$ and a given structure $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}})$, let the *type* of an element $a \in A$ be defined as $\text{type}(a) := \{R_i \in \tau \mid a \in R_i\}$. Let T_τ denote the set of all possible types of elements in $\mathcal{A}$ and observe that $|T_\tau| = 2^r$. We can order T_τ by letting $t_1 <_{\text{lex}} t_2$ if $\sum_{R_i \in t_1} 2^{i-1} < \sum_{R_i \in t_2} 2^{i-1}$ for $t_1, t_2 \in T_\tau$. The *type histogram* of the structure $\mathcal{A}$ is then defined as $\text{hist}(\mathcal{A}) := (e_t)_{t \in T_\tau}$, where e_t is the number of elements in $\mathcal{A}$ that have type t.

A *type modulator* is a set S_T consisting of triples (t_1, t_2, a), which states that we modify element a from having type t_1 to having type t_2. A type modulator is *admissible for a structure* $\mathcal{A}$, if for every tuple (t_1, t_2, a), the element a has type t_1 in $\mathcal{A}$ and every element occurs in at most one triple of the type modulator. Note that each admissible type modulator S_T gives rise to a modulator S and vice versa. Thus, we let $\|S_T\| := \|S\|$ and write $\mathcal{A} \triangleright S_T$ for the structure obtained by applying the modulator S that corresponds to the type modulator S_T to $\mathcal{A}$. We can define a lexicographic order on type modulators by letting $(t_1, t_2, a) <_{\text{lex}} (t_1', t_2', a')$ if (1) $t_1 <_{\text{lex}} t_1'$ or if (2) $t_1 = t_1'$ and $t_2 <_{\text{lex}} t_2'$ or if (3) $t_1 = t_1'$, $t_2 = t_2'$, and $a < a'$, where $a < a'$ is the order we assume on the universe. Two type modulators S_T, S_T' are *congruent*, if for every $(t_1, t_2, a) \in S_T$ and $(t_1', t_2', a') \in S_T'$ at the same position in the order, we have that $t_1 = t_1'$ and $t_2 = t_2'$. Now, for two congruent type modulators S_T and S_T', we have $S_T <_{\text{lex}} S_T'$ if $a < a'$ for all pairs of tuples $(t_1, t_2, a) \in S_T$ and $(t_1', t_2', a') \in S_T'$ at the same position in the order.

Note that two monadic structures are isomorphic if they have the same type histogram. This in turn means that two congruent type modulators yield the same structure up to isomorphism, so it suffices to consider lexicographically minimal type modulators. There are only $k^{|T_\tau|^2} = k^{2^{2r}}$ lexicographically minimal type modulators of size k, so the number of candidate modulators that need to be considered is polynomial in k.

Now, with a para-AC^0-circuit, we can check in parallel, if for any of these candidates, it holds that $\mathcal{A} \triangleright S_T \models \phi$. If this is the case, we accept, otherwise, we reject. ◀

4 Classical Complexity of Relation Modification Problems

In this section, we establish the following dichotomy: the classes of relation modification problems considered in this paper are either solvable in TC^0 or contain intractable problems.

4.1 Undirected Graphs, Directed Graphs, and Arbitrary Structures

As in the parameterized setting, we find that the complexity landscapes of relation modification problems coincide for undirected and directed graphs and arbitrary structures.

▶ **Theorem 4.1** (▼). *For each modification operation $\otimes \in \{\mathrm{del}, \mathrm{add}, \mathrm{edit}\}$, we have*

1. $\mathrm{RM}^{\otimes}_{\mathrm{arb}}(p) \subseteq \mathrm{AC}^0$*, if* $p \preceq e^*$.
2. $\mathrm{RM}^{\otimes}_{\mathrm{basic}}(p) \not\subseteq \mathrm{AC}^0$*, but* $\mathrm{RM}^{\otimes}_{\mathrm{arb}}(p) \subseteq \mathrm{TC}^0$*, if* $a \preceq p \preceq e^*a$.
3. $\mathrm{RM}^{\otimes}_{\mathrm{undir}}(p)$ *and* $\mathrm{RM}^{\otimes}_{\mathrm{dir}}(p)$ *both contain an* NP*-hard problem, if* $ae \preceq p$ *or* $aa \preceq p$.

4.2 Basic Graphs

Finally, we consider the restriction of input and target structures to basic graphs and prove the following tractability frontier.

▶ **Theorem 4.2** (▼). *For each modification operation $\otimes \in \{\mathrm{del}, \mathrm{add}, \mathrm{edit}\}$, we have*

1. $\mathrm{RM}^{\otimes}_{\mathrm{basic}}(p) \subseteq \mathrm{AC}^0$*, if* $p \preceq e^*$ *or* $p \preceq a$.
2. $\mathrm{RM}^{\otimes}_{\mathrm{basic}}(p) \not\subseteq \mathrm{AC}^0$*, but* $\mathrm{RM}^{\otimes}_{\mathrm{basic}}(p) \subseteq \mathrm{TC}^0$*, if* $aa \preceq p$ *or* $ae \preceq p$ *or* $ea \preceq p$ *and* $p \preceq e^*a$ *or* $p \preceq aa$ *or* $p \preceq ae$.
3. $\mathrm{RM}^{\otimes}_{\mathrm{basic}}(p)$ *and* $\mathrm{RM}^{\otimes}_{\mathrm{dir}}(p)$ *each contain an* NP*-hard problem, if* $aaa \preceq p$ *or* $aea \preceq p$ *or* $aee \preceq p$ *or* $aae \preceq p$ *or* $eae \preceq p$ *or* $eaa \preceq p$.

4.3 Monadic Structures

We conclude by considering structures over monadic vocabularies.

▶ **Theorem 4.3** (▼). *For each modification operation $\otimes \in \{\mathrm{del}, \mathrm{add}, \mathrm{edit}\}$, we have*

1. $\mathrm{RM}^{\otimes}_{\mathrm{mon}}(p) \subseteq \mathrm{AC}^0$*, if* $p \preceq e^*$.
2. $\mathrm{RM}^{\otimes}_{\mathrm{mon}}(p) \not\subseteq \mathrm{AC}^0$*, but* $\mathrm{RM}^{\otimes}_{\mathrm{mon}}(p) \subseteq \mathrm{TC}^0$*, if* $a \preceq p$.

5 Conclusion and Discussion

We provided a full classification of the parameterized and classical complexity of relation modification problems to target properties that are expressible in fragments of first-order logic. In our analysis, we considered (1) different modification operations, namely deleting relations, adding relations, and editing relations and (2) different input structures, namely monadic structures, basic graphs, undirected graphs, directed graphs, and arbitrary structures.

While the tractability frontier coincides for all modification operations, it differs with respect to the type of structure that is allowed as input. More specifically, we obtain a different complexity landscape on monadic structures and basic graphs each, whereas modifying all other input structures is equally hard.

The classifications we established are "strong" separations in the sense that all fragments either yield relation modification problems that are quite easy to solve (that is, they lie in para-$\mathrm{AC}^{0\uparrow}$ or TC^0) or that are intractable (that is, they contain W[2]-hard or NP-hard problems). See Table 1 for a full summary of our results.

Upon closer examination of our proofs, we find that considering different restrictions and counting measures for our modification budget yields identical complexity landscapes. More specifically, we observe the following:

▶ **Observation 5.1.** *For each* $\otimes \in \{\mathrm{del}, \mathrm{add}, \mathrm{edit}\}$ *and* $\mathrm{T} \in \{\mathrm{arb}, \mathrm{dir}, \mathrm{undir}, \mathrm{basic}, \mathrm{mon}\}$*,* $\mathrm{RM}^{\otimes}_{\mathrm{T}}$ *(*$\text{p-RM}^{\otimes}_{\mathrm{T}}$*, resp.) exhibits the complexity landscape as outlined in Table 1, (1) when up to k tuples (across all relations) can be modified, (2) when up to k tuples of a single relation can be modified, and (3) when up to k tuples can be modified in every relation. Moreover, we find that for* $\mathrm{T} \in \{\mathrm{arb}, \mathrm{dir}\}$ *and modification budget k, the complexity of the problem* $\mathrm{RM}^{\otimes}_{\mathrm{T}}$ *(or* $\text{p-RM}^{\otimes}_{\mathrm{T}}$*, resp.) where we count the number of modified* ordered *tuples and compare it to k, coincides with the variant of counting the number of* unordered *tuples.*

The latter raises the following question: Are there other natural counting versions that are of interest – for instance in database repair – under which the complexity of $\mathrm{RM}^{\otimes}_{\mathrm{arb}}$ and $\mathrm{RM}^{\otimes}_{\mathrm{dir}}$ (of $\text{p-RM}^{\otimes}_{\mathrm{arb}}$ and $\text{p-RM}^{\otimes}_{\mathrm{dir}}$, resp.) diverges?

The relation modification operations we considered in this paper are not exhaustive: Graph theory studies other edge modification operations such as *edge contraction*, *edge splitting*, or *edge flipping*. Edge flipping may be of particular interest as this operation can be used to define a variant of the feedback arc set problem, where instead of making a graph acyclic by deleting edges, we have to flip them. This problem was recently used to solve the One-Sided-Crossing-Minimization problem efficiently in a practical setting [4].

Another interesting direction of research is to extend our results to higher-order logics, such as monadic second-order logic or existential second-order logic. While these logics allow to express more properties of interest, such as acyclicity, which allows us to model the feedback arc set problem as an edge deletion or edge flipping problem, even model checking can become intractable there. In that regard, standard extensions of first-order logic could also be of interest, as they may capture more properties, such as transitive closures, fixed-point operators, or counting.

Finally, a closer inspection of the fragments tractable in the parameterized setting is of interest: It immediately follows from our classical complexity results that the fragment e^*a is in para-AC^0 for all graph classes and modification operations. With respect to the fragment a^* – which for instance can be used to describe the interesting meta-problem H-free Deletion for fixed graphs H – we know that the respective relation modification problems lie in para-$\mathrm{AC}^{0\uparrow}$, but the question of whether they can be placed in para-AC^0 as well is open (see also [3, page 52]).

References

1 Faisal N. Abu-Khzam, Michael A. Langston, Pushkar Shanbhag, and Christopher T. Symons. Scalable parallel algorithms for FPT problems. *Algorithmica*, 45(3):269–284, 2006. doi:10.1007/S00453-006-1214-1.

2 Marcelo Arenas, Leopoldo Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the Eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '99, pages 68–79, New York, NY, USA, 1999. Association for Computing Machinery. doi:10.1145/303976.303983.

3 Max Bannach. *Parallel Parameterized Algorithms.* PhD thesis, University of Lübeck, Germany, 2020.

4 Max Bannach, Florian Chudigiewitsch, Kim-Manuel Klein, and Marcel Wienöbst. PACE solver description: Uzl exact solver for one-sided crossing minimization. In Édouard Bonnet and Pawel Rzazewski, editors, *19th International Symposium on Parameterized and Exact Computation, IPEC 2024*, volume 321 of *LIPIcs*, pages 28:1–28:4. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.IPEC.2024.28.

5 Max Bannach, Florian Chudigiewitsch, and Till Tantau. Existential second-order logic over graphs: Parameterized complexity. In Neeldhara Misra and Magnus Wahlström, editors, *18th International Symposium on Parameterized and Exact Computation, IPEC 2023*, volume 285 of *LIPIcs*, pages 3:1–3:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.IPEC.2023.3.

6 Max Bannach, Florian Chudigiewitsch, and Till Tantau. On the descriptive complexity of vertex deletion problems. In Rastislav Královic and Antonín Kucera, editors, *49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024*, volume 306 of *LIPIcs*, pages 17:1–17:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.MFCS.2024.17.

7 Max Bannach, Malte Skambath, and Till Tantau. On the parallel parameterized complexity of maxsat variants. *Journal of Artificial Intelligence Research*, 78, 2023. doi:10.1613/JAIR.1.14748.

8 Max Bannach, Christoph Stockhusen, and Till Tantau. Fast parallel fixed-parameter algorithms via color coding. In Thore Husfeldt and Iyad A. Kanj, editors, *10th International Symposium on Parameterized and Exact Computation, IPEC 2015*, volume 43 of *LIPIcs*, pages 224–235. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.IPEC.2015.224.

9 Max Bannach and Till Tantau. Parallel multivariate meta-theorems. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016*, volume 63 of *LIPIcs*, pages 4:1–4:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.4.

10 Max Bannach and Till Tantau. Computing kernels in parallel: Lower and upper bounds. In *13th International Symposium on Parameterized and Exact Computation, IPEC 2018*, pages 13:1–13:14, 2018. doi:10.4230/LIPIcs.IPEC.2018.13.

11 Max Bannach and Till Tantau. Computing hitting set kernels by AC^0-circuits. *Theory of Computing Systems*, 64(3):374–399, 2020. doi:10.1007/s00224-019-09941-z.

12 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. doi:10.1023/B:MACH.0000033116.57574.95.

13 Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 4502–4510, 2016. URL: https://proceedings.neurips.cc/paper/2016/hash/3147da8ab4a0437c15ef51a5cc7f2dc4-Abstract.html.

14 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Journal of Computational Geometry*, 4:235–282, 1994. doi:10.1016/0925-7721(94)00014-X.

15 Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Jornal of Computational Biology*, 6(3/4):281–297, 1999. doi:10.1089/106652799318274.

16 Seymour Benzer. On the topology of the genetic fine structure. *Proceedings of the National Academy of Sciences of the United States of America*, 45(11):1607–1620, 1959.

17 Manuel Bodirsky and Santiago Guzmán-Pro. Hereditary first-order logic: the tractable quantifier prefix classes, 2025. `arXiv:2411.10860`.

18 Manuel Bodirsky and Santiago Guzmán Pro. On the computational power of extensional eso, 2025. `doi:10.48550/arXiv.2511.08515`.

19 Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

20 Daniel Brügmann, Christian Komusiewicz, and Hannes Moser. On generating triangle-free graphs. *Electronic Notes in Discrete Mathematics*, 32:51–58, 2009. `doi:10.1016/J.ENDM.2009.02.008`.

21 Pablo Burzyn, Flavia Bonomo, and Guillermo Durán. NP-completeness results for edge modification problems. *Discrete Applied Mathematics*, 154(13):1824–1844, 2006. `doi:10.1016/J.DAM.2006.03.031`.

22 Andrea Calì, Domenico Lembo, and Riccardo Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 260–271, New York, NY, USA, 2003. Association for Computing Machinery. `doi:10.1145/773153.773179`.

23 Marco Cesati and Miriam Di Ianni. Parameterized parallel complexity. In David J. Pritchard and Jeff Reeve, editors, *Euro-Par '98 Parallel Processing, 4th International Euro-Par Conference, Southampton, UK, September 1–4, 1998, Proceedings*, volume 1470 of *Lecture Notes in Computer Science*, pages 892–896. Springer, 1998. `doi:10.1007/BFB0057945`.

24 James Cheetham, Frank K. H. A. Dehne, Andrew Rau-Chaplin, Ulrike Stege, and Peter J. Taillon. Solving large FPT problems on coarse-grained parallel machines. *Journal of Computing and System Sciences*, 67(4):691–706, 2003. `doi:10.1016/S0022-0000(03)00075-8`.

25 Yijia Chen and Jörg Flum. Some lower bounds in parameterized ac^0. *Information and Computation*, 267:116–134, 2019. `doi:10.1016/J.IC.2019.03.008`.

26 Hyeong-Ah Choi, Kazuo Nakajima, and Chong S. Rim. Graph bipartization and via minimization. *SIAM Journal on Discrete Mathematics*, 2(1):38–47, 1989. `doi:10.1137/0402004`.

27 Jan Chomicki and Jerzy Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Information and Computation*, 197(1):90–121, 2005. `doi:10.1016/j.ic.2004.04.007`.

28 Florian Chudigiewitsch, Marlene Gründel, Christian Komusiewicz, Nils Morawietz, and Till Tantau. The descriptive complexity of relation modification problems, 2026. `doi:10.48550/arXiv.2603.22043`.

29 Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Computer Science Review*, 48:100556, 2023. `doi:10.1016/j.cosrev.2023.100556`.

30 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. `doi:10.1007/978-3-319-21275-3`.

31 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. `doi:10.1007/978-1-4612-0515-9`.

32 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer, 2nd edition, 2005. `doi:10.1007/3-540-28788-4`.

33 Thomas Eiter, Yuri Gurevich, and Georg Gottlob. Existential second-order logic over strings. *Journal of the ACM*, 47(1):77–131, 2000. `doi:10.1145/331605.331609`.

34 Michael Elberfeld, Christoph Stockhusen, and Till Tantau. On the space and circuit complexity of parameterized problems: Classes and completeness. *Algorithmica*, 71(3):661–701, 2015. `doi:10.1007/S00453-014-9944-Y`.

35 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006. `doi:10.1007/3-540-29953-X`.

36 Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. On the parameterized complexity of graph modification to first-order logic properties. *Theory of Computing Systems*, 64(2):251–271, 2020. doi:10.1007/s00224-019-09938-8.

37 Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Parameterized complexity of elimination distance to first-order logic properties. In *ACM Transactions on Computational Logic*, volume 23(3), pages 17:1–17:35, 2022. doi:10.1145/3517129.

38 Fedor V. Fomin, Saket Saurabh, and Neeldhara Misra. Graph modification problems: A modern perspective. In Jianxin Wang and Chee-Keng Yap, editors, *Frontiers in Algorithmics - 9th International Workshop, FAW 2015*, volume 9130 of *Lecture Notes in Computer Science*, pages 3–6. Springer, 2015. doi:10.1007/978-3-319-19647-3_1.

39 Georges Gardarin and Stefano Spaccapietra. Integrity of data bases: A general lockout algorithm with deadlock avoidance. In G. M. Nijssen, editor, *Modelling in Data Base Management Systems, Proceeding of the IFIP Working Conference on Modelling in Data Base Management Systems, Freudenstadt, Germany*, pages 395–412. North-Holland, 1976.

40 Georg Gottlob, Phokion G. Kolaitis, and Thomas Schwentick. Existential second-order logic over graphs: Charting the tractability frontier. *Journal of the ACM*, 51(2):312–362, 2004. doi:10.1145/972639.972646.

41 Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2007. doi:10.1007/3-540-68804-8.

42 Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017. doi:10.1017/9781139028868.

43 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In Rodney G. Downey, Michael R. Fellows, and Frank K. H. A. Dehne, editors, *Parameterized and Exact Computation, First International Workshop, IWPEC 2004*, volume 3162 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 2004. doi:10.1007/978-3-540-28639-4_15.

44 Mohsen MD Hassan and Gary L Hogg. A review of graph theory application to the facilities layout problem. *Omega*, 15(4):291–300, 1987. doi:10.1016/0305-0483(87)90017-X.

45 Steef Hegeman, Jan Martens, and Alfons Laarman. Uniformity within parameterized circuit classes. In Akanksha Agrawal and Erik Jan van Leeuwen, editors, *20th International Symposium on Parameterized and Exact Computation, IPEC 2025*, volume 358 of *LIPIcs*, pages 27:1–27:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.IPEC.2025.27.

46 Falk Hüffner, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. *Theory of Computing Systems*, 47(1):196–217, 2010. doi:10.1007/S00224-008-9150-X.

47 Neil Immerman. *Descriptive Complexity*. Springer Verlag, 1998.

48 Michael Jünger and Petra Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16(1):33–59, 1996. doi:10.1007/BF02086607.

49 Athanasios L. Konstantinidis and Charis Papadopoulos. Cluster deletion on interval graphs and split related graphs. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019*, volume 138 of *LIPIcs*, pages 12:1–12:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.MFCS.2019.12.

50 Paraschos Koutris, Xiating Ouyang, and Jef Wijsen. Consistent query answering for primary keys on path queries. In *Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS'21, pages 215–232, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3452021.3458334.

51 Robert Kudelic. *Feedback Arc Set – A History of the Problem and Algorithms*. Springer Briefs in Computer Science. Springer, 2022. doi:10.1007/978-3-031-10515-9.

52 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

53 Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 93–106, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1376616.1376629.

54 Andrei Lopatenko and Leopoldo Bertossi. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. In Thomas Schwentick and Dan Suciu, editors, *Database Theory – ICDT 2007*, pages 179–193, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

55 Xuesong Lu, Yi Song, and Stéphane Bressan. Fast identity anonymization on graphs. In Stephen W. Liddle, Klaus-Dieter Schewe, A Min Tjoa, and Xiaofang Zhou, editors, *Database and Expert Systems Applications - 23rd International Conference, DEXA 2012*, volume 7446 of *Lecture Notes in Computer Science*, pages 281–295. Springer, 2012. doi:10.1007/978-3-642-32600-4_21.

56 Frederico Mancini. *Graph modification problems related to graph classes*. PhD thesis, University of Bergen, Norway, May 2008.

57 Laure Morelle, Ignasi Sau, and Dimitrios M. Thilikos. Graph modification of bounded size to minor-closed classes as fast as vertex deletion. In Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman, editors, *33rd Annual European Symposium on Algorithms, ESA 2025*, volume 351 of *LIPIcs*, pages 7:1–7:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ESA.2025.7.

58 Assaf Natanzon, Ron Shamir, and Roded Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1):109–128, 2001. doi:10.1016/S0166-218X(00)00391-7.

59 Romeo Rizzi, Vineet Bafna, Sorin Istrail, and Giuseppe Lancia. Practical algorithms and fixed-parameter tractability for the single individual snp haplotyping problem. In *Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, WABI '02, pages 29–43, Berlin, Heidelberg, 2002. Springer-Verlag. doi:10.1007/3-540-45784-4_3.

60 Zimo Sheng and Mingyu Xiao. A discharging method: Improved kernels for edge triangle packing and covering. In Weili Wu and Guangmo Tong, editors, *Computing and Combinatorics – 29th International Conference, COCOON 2023*, volume 14423 of *Lecture Notes in Computer Science*, pages 171–183. Springer, 2023. doi:10.1007/978-3-031-49193-1_13.

61 Sławomir Staworko and Jan Chomicki. Consistent query answers in the presence of universal constraints. *Information Systems*, 35(1):1–22, 2010. doi:10.1016/j.is.2009.03.004.

62 Till Tantau. Existential second-order logic over graphs: A complete complexity-theoretic classification. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015*, pages 703–715, 2015. doi:10.4230/LIPIcs.STACS.2015.703.

63 Shaobin Xu, Minghui Sun, and Jun Qin. A high-scalability graph modification system for large-scale networks. *Computer Graphics Forum*, 43(6), 2024. doi:10.1111/CGF.15191.

64 Mihalis Yannakakis. Node-and edge-deletion NP-complete problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 253–264, New York, NY, USA, 1978. Association for Computing Machinery. doi:10.1145/800133.804355.

65 Mihalis Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981. doi:10.1137/0210021.