

On Existential MSO and its Relation to ETH

Robert Ganian¹, Ronald de Haan², Iyad Kanj³, and Stefan Szeider⁴

1 Algorithms and Complexity Group, TU Wien, Vienna, Austria
ganian@ac.tuwien.ac.at

2 Algorithms and Complexity Group, TU Wien, Vienna, Austria
dehaan@ac.tuwien.ac.at

3 School of Computing, DePaul University, Chicago, IL, USA
ikanj@cs.depaul.edu

4 Algorithms and Complexity Group, TU Wien, Vienna, Austria
sz@ac.tuwien.ac.at

Abstract

Impagliazzo *et al.* proposed a framework, based on the logic fragment defining the complexity class SNP, to identify problems that are equivalent to k -CNF-SAT modulo subexponential-time reducibility (serf-reducibility). The subexponential-time solvability of any of these problems implies the failure of the Exponential Time Hypothesis (ETH). In this paper, we extend the framework of Impagliazzo *et al.*, and identify a larger set of problems that are equivalent to k -CNF-SAT modulo serf-reducibility. We propose a complexity class, referred to as Linear Monadic NP, that consists of all problems expressible in existential monadic second order logic whose expressions have a linear *measure* in terms of a complexity parameter, which is usually the universe size of the problem.

This research direction can be traced back to Fagin's celebrated theorem stating that NP coincides with the class of problems expressible in existential second order logic. Monadic NP, a well-studied class in the literature, is the restriction of the aforementioned logic fragment to existential *monadic* second order logic. The proposed class Linear Monadic NP is then the restriction of Monadic NP to problems whose expressions have linear measure in the complexity parameter.

We show that Linear Monadic NP includes many natural complete problems such as the satisfiability of linear-size circuits, dominating set, independent dominating set, and perfect code. Therefore, for any of these problems, its subexponential-time solvability is equivalent to the failure of ETH. We prove, using logic games, that the aforementioned problems are inexpressible in the monadic fragment of SNP, and hence, are not captured by the framework of Impagliazzo *et al.* Finally, we show that FEEDBACK VERTEX SET is inexpressible in existential monadic second order logic, and hence is not in Linear Monadic NP, and investigate the existence of certain reductions between FEEDBACK VERTEX SET (and variants of it) and 3-CNF-SAT.

1998 ACM Subject Classification F.2.0 Analysis of Algorithms and Problem Complexity, F.1.3 Complexity Measures and Classes, F.4.1 Mathematical Logic

Keywords and phrases Exponential Time Hypothesis (ETH), Monadic Second Order Logic, Subexponential Time Complexity, Serf-Reducibility, Logic Games

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.42



© Robert Ganian, Ronald de Haan, Iyad Kanj, and Stefan Szeider;
licensed under Creative Commons License CC-BY

41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016).

Editors: Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier; Article No. 42; pp. 42:1–42:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Motivation and related work

The area of exact algorithms seeks moderately exponential-time algorithms for NP-hard problems that improve on the trivial brute-force algorithms. This area of research has been gaining a lot of traction in the last two decades. Nowadays, there is a large set of important NP-complete problems such that for each problem there is a long list of exact algorithms, each improving slightly on the running time of the preceding one; we refer the interested reader to [19] for exposure to the area of exact algorithms. Perhaps the most well-known—and important—NP-hard problem is the satisfiability of Boolean formulas in conjunctive normal form, abbreviated CNF-SAT. Its importance stems from its practical relevance as well as from the canonical role it plays for the complexity class NP, being the first problem shown to be complete for this class. Despite all the exercised efforts over the last several decades, the best algorithms remain equivalent (in terms of the upper bound on their running time) to the trivial brute-force algorithm that solves CNF-SAT in time $\mathcal{O}^*(2^n)$,¹ where n is the number of Boolean variables in the input formula. It has become a common belief among a significant number of researchers in the area of exact algorithms that no $\mathcal{O}^*(2^{cn})$ -time algorithm, for any constant $c < 1$, exists for CNF-SAT. If we restrict the clause-width to be at most k (where $k \geq 3$) in the instances of CNF-SAT, we obtain the k -CNF-SAT problem, for which there is a long list of moderately exponential-time algorithms culminating in the currently-best deterministic algorithm (for a general value of k) running in time $\mathcal{O}((2 - 2/(k + 1))^n)$ [12].

The sequence of improvements in the running time of moderately exponential-time algorithms for k -CNF-SAT, as well as for other NP-complete problems, led researchers to ask whether we can “indefinitely” keep improving the running time of exact algorithms for these problems. More formally, it triggered the question of whether an algorithm solving k -CNF-SAT in time $\mathcal{O}(2^{o(n)})$, or equivalently in time $\mathcal{O}(2^{\varepsilon n})$ for every $\varepsilon > 0$, exists; an algorithm with such running time is referred to as a *subexponential-time algorithm*. In their seminal paper, Impagliazzo *et al.* [22] investigated this question. They proved that the existence of a subexponential-time algorithm for k -CNF-SAT, for any integer-constant $k \geq 3$, is related to its membership in the class SNP, introduced in [30]. The class SNP consists of all search problems expressible by second-order existential formulas whose first-order part is universal; we shall refer to the aforementioned logic fragment as *SNP logic*. For a problem in SNP, they defined a complexity parameter based on its SNP logic expression. They introduced the notion of completeness for the class SNP under serf-reductions, which are subexponential-time complexity preserving reductions with respect to the aforementioned complexity parameter, and identified a class of problems that are complete for SNP under serf-reducibility; the subexponential-time solvability of any of these problems with respect to its corresponding complexity parameter implies the subexponential-time solvability of all problems in SNP. They proved that many well-known NP-hard problems are SNP-complete under serf-reducibility with respect to a complexity parameter that is linear in the natural universe size of the problem, including k -CNF-SAT (universe is the Boolean variables in the formula), VERTEX COVER, INDEPENDENT SET, and c -COLORABILITY ($c \geq 3$) (universe is the vertex set of the graph), for which extensive efforts to develop subexponential-time

Kanj was partially supported by DFG, project NI 369/12. Ganian, de Haan and Szeider were partially supported by FWF, projects P 26200 and P 26696. Ganian is also affiliated with FI MU, Brno, Czech Republic.

¹ The \mathcal{O}^* notation suppresses a polynomial factor in the input length.

algorithms have been made in the last three decades with no success. This led them to formulate the Exponential Time Hypothesis (ETH), stating that k -CNF-SAT is not solvable in subexponential-time, which is equivalent to the statement that not all SNP problems are solvable in subexponential time. ETH has become a standard hypothesis in complexity theory for proving hardness results that is closely related to the computational intractability of a large class of well-known NP-hard problems, measured from a number of different angles, such as subexponential-time complexity, fixed-parameter tractability, and approximation [6, 7, 10, 28, 29].

Our results

The SNP logic framework developed by Impagliazzo *et al.* [22] captures some well-known NP-complete problems that are serf-reducible to k -CNF-SAT with respect to their universe size, but fails to capture many others. Using logic games, we prove in Section 3 that there are many natural NP-complete problems that are not captured by this SNP logic framework, and yet are serf-reducible to k -CNF-SAT. Examples of such problems include the satisfiability problem for linear-size circuits (LINEAR CIRCUIT-SAT), DOMINATING SET, INDEPENDENT DOMINATING SET, and PERFECT CODE. The restrictedness of the SNP-based framework, due to the restrictedness of SNP logic to allowing only universal quantifiers in the first order part, prevents the expressibility of problems such as the aforementioned ones, as we show in this paper that none of these problems can be expressed in SNP logic with a complexity parameter that is linear in the natural universe size of the problem.

We propose a complexity class, referred to as Linear Monadic NP, that captures more problems than the SNP-based framework of Impagliazzo *et al.* [22]. Fagin's celebrated theorem [15] states that NP coincides with the class of properties expressible in existential second-order logic. The class monadic NP, introduced by Fagin *et al.* [16], is the restriction of NP to problems expressible in monadic second-order logic in which the second-order part is existential, that is, all second-order variables have arity at most one (*i.e.*, set variables), and no universal quantification is allowed over these relations. The class monadic NP is a well-studied complexity class, and several properties have been shown to be inexpressible in the logic fragment defining this class [2, 3, 16, 23, 25]. We show in this paper that the class monadic NP plays an important role as well in identifying problems that are equivalent to k -CNF-SAT under serf-reducibility. We define a measure (for expressions), and use it to define a logic fragment, *Linear Existential Monadic Second-Order Logic* (LEMSO), consisting of the restriction of existential monadic second-order logic to expressions whose measure is linear in the complexity parameter, where the complexity parameter is defined as in Impagliazzo *et al.* [22]. The class Linear Monadic NP consists of all search problems expressible in LEMSO. All natural problems described in [22] including k -CNF-SAT, INDEPENDENT SET, VERTEX COVER, c -COLORABILITY, which are complete for SNP under serf-reducibility with respect to their natural universe size (number of variables/vertices), are also complete for Linear Monadic NP under serf-reducibility. In fact, each problem in the class of problems defined by Impagliazzo *et al.* [22] consisting of all problems expressible in the SNP logic with a linear complexity parameter (as defined in [22]) is in Linear Monadic NP. We prove that problems such as LINEAR CIRCUIT-SAT, DOMINATING SET, and INDEPENDENT DOMINATING SET are in Linear Monadic NP (and are actually complete for Linear Monadic NP), but *are not* expressible in the SNP logic, thus showing that the set of problems expressible in the SNP logic with a linear complexity parameter is a proper subset of Linear Monadic NP. This implies that the subexponential-time solvability of any of the aforementioned problems is equivalent to the failure of ETH. Our inexpressibility proofs use the framework of logic games, namely the Ajtai-Fagin game and the Ehrenfeucht-Fraïssé game.

Finally, we show using logic games that FEEDBACK VERTEX SET is inexpressible in EMSO, and hence is not in Linear Monadic NP. Whereas it can be easily shown that 3-CNF-SAT is serf-reducible to FEEDBACK VERTEX SET (by composing the two folklore polynomial-time reductions from 3-CNF-SAT to VERTEX COVER and from VERTEX COVER to FEEDBACK VERTEX SET), it is open whether a serf-reduction exists in the other direction. We show that there is a polynomial-time reduction from FEEDBACK VERTEX SET to 3-CNF-SAT with a quasi-linear increase in the universe size, and define a variant of FEEDBACK VERTEX SET that is equivalent to 3-CNF-SAT under serf-reducibility.

2 Preliminaries

2.1 Satisfiability

The CNF-SATISFIABILITY problem, shortly CNF-SAT, is given a formula F in the CNF form, decide whether or not F is satisfiable. The *width* of a clause in a CNF formula F is the number of literals in the clause. The k -CNF-SAT problem, where $k \geq 2$, is the restriction of the CNF-SAT problem to instances in which the width of each clause is at most k . It is well known that the k -CNF-SAT problem for $k \geq 3$ is NP-complete [20]. LINEAR CNF-SAT is the restriction of CNF-SAT to formulas with a linear number of clauses with respect to the number of variables.

A *circuit* is a directed acyclic graph. The vertices of indegree 0 are called the *variables* or *input gates*, and are labeled either by *positive literals* x_i or by *negative literals* \bar{x}_i . The vertices of indegree larger than 0 are called the *gates* and are labeled with Boolean operators \wedge or \vee . A special gate of outdegree 0 is designated as the *output gate*. We do not allow negation gates in the circuit since, by De Morgan's laws, a general circuit can be efficiently converted into the above circuit model. The *size* of a circuit C , denoted $|C|$, is the number of gates in it. A circuit C is *satisfiable* if there is a truth assignment to the input variables of C that makes C evaluate to 1. The CIRCUIT-SATISFIABILITY problem, shortly CIRCUIT-SAT, is given a circuit C , decide whether or not C is satisfiable. LINEAR CIRCUIT-SAT is the restriction of CIRCUIT-SAT to circuits of linear size with respect to the number of variables.

2.2 Subexponential time and SNP

For any $n \in \mathbb{N}$, by $[n]$ we denote the set $\{1, 2, \dots, n\}$. The time complexity functions used in this paper are assumed to be proper complexity functions.

It is clear that CNF-SAT is solvable in time $2^n |F|^{\mathcal{O}(1)}$, where F is the input instance and n is the number of variables in F . We say that CNF-SAT is solvable in *subexponential time* if there exists an algorithm that solves the problem in time $2^{o(n)} |F|^{\mathcal{O}(1)}$. Using the results of [8] and [17], the above definition is equivalent to the following: CNF-SAT is solvable in *subexponential time* if there exists an algorithm that for all $\varepsilon = 1/\ell$, where ℓ is a positive integer, solves the problem in time $2^{\varepsilon n} |I|^{\mathcal{O}(1)}$.

Let Q and Q' be two problems, and let μ and μ' be two parameter functions defined on instances of Q and Q' , respectively. In the case of CNF-SAT, μ and μ' will be the number of variables in the instances of these problems. A *subexponential-time Turing reduction family* [22], see also [17], shortly a *serf-reduction*², is an algorithm A with an oracle to Q' such that there are computable functions $f, g : \mathbb{N} \rightarrow \mathbb{N}$ satisfying: (1) given a pair (I, ε)

² Serf-reductions were introduced in [22]. Here we use the definition given in [17]. There is a slight difference between the two definitions, and the latter definition is more flexible for our purposes.

where $I \in Q$ and $\varepsilon = 1/\ell$ (ℓ is a positive integer), A decides I in time $f(1/\varepsilon)2^{\varepsilon\mu(I)}|I|^{\mathcal{O}(1)}$; and (2) for all oracle queries of the form “ $I' \in Q'$ ” posed by A on input (I, ε) , we have $\mu'(I') \leq g(1/\varepsilon)(\mu(I) + \log |I|)$.

The class SNP consists of all search problems expressible by second-order existential formulas whose first-order part is universal [30]; that is, search problems expressible by second-order formulas of the form $\exists R_1 \dots \exists R_q \forall z_1 \dots \forall z_r \Phi(S, R_1, \dots, R_q, z_1, \dots, z_r)$, where S is the input structure, R_1, \dots, R_q are (bound) relations, z_1, \dots, z_r are first-order variables, and Φ is a quantifier-free Boolean formula. We will refer to the preceding logic fragment as the *SNP logic*. Impagliazzo *et al.* [22] defined a complexity parameter for each such expression equal to $\sum_{i=1}^q |R_i|^{\alpha_i}$, where $|R_i|$ and α_i , $i = 1, \dots, q$, are the encoding length of R_i and its arity, respectively. For monadic relations, the number of bits needed to describe R_i , is the size of the universe on which R_i is interpreted, and hence, as noted in [22], the complexity parameter in this case is linear in the universe size. They formulated the *Exponential Time Hypothesis* (ETH) stating that k -CNF-SAT (for any $k \geq 3$) cannot be solved in subexponential time $2^{o(n)}$, where n is the number of variables in the input formula. Therefore, there exists $c > 0$ such that k -CNF-SAT cannot be solved in time $\mathcal{O}(2^{cn})$. ETH is equivalent to the statement that not all SNP problems are solvable in subexponential time.

3 Linear Monadic NP and Serf-Reducibility to Linear Circuit-Sat

In this section, we present a fragment of existential second-order logic that extends the SNP logic introduced by Impagliazzo *et al.* [22] in the sense that it captures a larger set of problems that are serf-reducible to 3-CNF-SAT. The logic fragment we propose is a restriction of the well-studied existential monadic second-order logic (EMSO) that defines the complexity class Monadic NP (modulo standard syntactic augmentations to allow the expression of optimization problems). To define this logic fragment, we introduce a measure/function that associates with each expression in EMSO a value in terms of the complexity parameter of the expression, and we define Linear Monadic NP to be the restriction of Monadic NP to those expressions whose measure is linear in the complexity parameter. The complexity parameter of the expression we use is the same parameter defined by [22] for expressions in SNP: If R_1, \dots, R_q are the existentially-quantified second-order relations in the expression, where R_i has arity α_i , for $i = 1, \dots, q$, then the complexity parameter of the expression is $\sum_{i=1}^q |R_i|^{\alpha_i}$.

We note that the SNP logic does not restrict the existentially quantified relations to be monadic, and we could have opted to do the same in this paper (*i.e.*, not restrict ourselves to monadic relations and Monadic NP), and the results in this paper would not have been affected. However, observe that the aforementioned complexity parameter defined by [22] and also used by us, when interpreted on a given structure (graph, CNF-formula, etc.) yields a parameter that is equal to $\sum_{i=1}^q |U|^{\alpha_i}$, where U is the natural universe on which the relations R_1, \dots, R_q are interpreted. Hence, this parameter can only be linear in the universe size (number of vertices in a graph, number of variables in a formula, etc.) if all relations in the expression are monadic; otherwise, the complexity parameter will be at least quadratic. To be able to use serf-reductions to claim that a subexponential-time algorithm for 3-CNF-SAT yields a subexponential-time algorithm for the search problem under consideration with respect to a complexity parameter that is linear in the universe size of the problem, the existentially-quantified relations in the SNP logic expression of the problem need to be monadic. We will refer to this fragment of SNP logic as *monadic SNP logic*, which, by the above, is the fragment of interest in the framework of Impagliazzo *et al.* [22] when studying the equivalence between natural problems modulo serf-reducibility.

We refer to the corresponding complexity class as Monadic SNP. By the same token, and without loss of generality, in this paper we restrict ourselves to Monadic NP.

3.1 Linear Monadic NP

The fragment of existential second-order logic that we consider is based on an inductively defined measure. Intuitively, this measure captures the size of the Boolean circuit that can be constructed for a concrete instance of the problem that is satisfiable if and only if there is an assignment to the second-order variables that makes the first-order part of the formula true.

We consider EMSO formulas with a single free monadic second-order variable, to be able to express optimization problems where solutions of a given (exact) size are sought.

In particular, let $\varphi(S) = \exists R_1 \dots \exists R_q Q_1 z_1 \dots Q_r z_r \Phi(S, R_1, \dots, R_q, z_1, \dots, z_r)$ be an existential second-order logic formula, where S is a free monadic second-order variable, where R_1, \dots, R_q are (bound) monadic second-order variables, where z_1, \dots, z_r are first-order variables and $Q_1, \dots, Q_r \in \{\exists, \forall\}$, and where Φ is quantifier-free.

Moreover, let A be an optimization problem consisting of inputs (I, N) , where I is a relational structure over the same relational vocabulary as Φ , and where N is a positive integer.

We say that $\varphi(S)$ expresses the problem A if for each input (I, N) it holds that $(I, N) \in A$ if and only if there is an interpretation S_0 of S in I of size exactly N such that $I \models \varphi(S_0)$.

More generally, we also consider existential second-order logic formulas $\varphi(S)$ where the first-order part does not need to be in prenex form, that is, first-order quantifications and Boolean connectives can freely alternate.

Without loss of generality, we suppose that formulas $\varphi(S)$ are in negation normal form.

For search problems that do not involve an optimization component (*e.g.*, 3-CNF-SAT), we can simply omit the free variable S in the logic expression and the integer N in the problem input. All definitions and results extend straightforwardly to this setting.

For optimization problems where solutions of a given minimum or maximum size are sought, it suffices to investigate the variant of the problem involving solutions of exactly the given size, because it can easily be shown that there is a self-reduction between these different problem variants.

Our size measure s is based on inductively defined measures, s_\forall and s_\exists , that we define below. All measures take as input a second-order logic formula, and return an arithmetic expression over the single variable n . For any formula $\varphi(S)$ that expresses a search problem A , intuitively, the expression $s(\varphi(S)) = f(n)$ denotes an upper bound on the size of the Boolean circuit that expresses a given instance in terms of the input size n .

We inductively define these measures as follows. Here we let Q range over $\{\forall, \exists\}$, and we let \bar{Q} denote the unique quantifier in $\{\forall, \exists\} \setminus \{Q\}$. Moreover, we let \circ range over $\{\wedge, \vee\}$, and we let a denote any atom.

$$s(\exists R_i \psi) = s(\psi); \quad (1) \quad s(\psi_1 \circ \psi_2) = 1 + s(\psi_1) + s(\psi_2); \quad (2)$$

$$s(Qz_i \psi) = \begin{cases} n & \text{if } \psi \text{ does not contain any occurrence of } \overline{Q}; \\ 1 + s_Q(Qz_i \psi) & \text{otherwise;} \end{cases} \quad (3)$$

$$s_Q(\overline{Q}z_i \psi) = 1 + n \cdot s_{\overline{Q}}(\psi); \quad (4) \quad s_Q(a) = s_Q(\neg a) = 0; \quad (6)$$

$$s_Q(Qz_i \psi) = n \cdot s_Q(\psi); \quad (5)$$

$$s_Q(\psi_1 \circ \psi_2) = s_Q(\psi_2) \quad \text{if } \psi_1 \text{ contains no occurrences of second-order variables;} \quad (7)$$

$$s_Q(\psi_1 \circ \psi_2) = s_Q(\psi_1) \quad \text{if } \psi_2 \text{ contains no occurrences of second-order variables;} \quad (8)$$

for the case where both ψ_1 and ψ_2 contain second-order variables, we let:

$$s_{\exists}(\psi_1 \vee \psi_2) = s_{\exists}(\psi_1) + s_{\exists}(\psi_2); \quad (9) \quad s_{\forall}(\psi_1 \vee \psi_2) = 1 + s_{\exists}(\psi_1) + s_{\exists}(\psi_2); \quad (11)$$

$$s_{\forall}(\psi_1 \wedge \psi_2) = s_{\forall}(\psi_1) + s_{\forall}(\psi_2); \quad (10) \quad s_{\exists}(\psi_1 \wedge \psi_2) = 1 + s_{\forall}(\psi_1) + s_{\forall}(\psi_2). \quad (12)$$

The intuition behind the definition of the measure is that $s(\varphi(S))$ measures the size of the Boolean circuit C that is needed to express the formula $\varphi(S)$ when interpreted over the input I ; $s(\varphi(S))$ returns an arithmetic expression with a single symbol, n , that is interpreted as the size of the universe U of the relational structure I . The variables of C are of the form $x_{T,e}$, representing whether an element e in the domain of I is chosen to be part of the interpretation of the monadic second-order variable T . The circuit C then encodes the first-order part of φ .

The computation of $s(\varphi)$ works as follows. The existential second-order quantifiers are disregarded, and the outermost Boolean connectives are dealt with in a straightforward inductive manner.

For every maximal subformula ψ that starts with a first-order quantifier Q , there are two options. Either ψ contains only first-order quantifiers of a single type—in which case the measure s returns n for this subformula—or ψ contains quantifiers of both types. In the latter case, the size of the (sub)circuit to represent ψ is measured inductively using s_Q .

The measure s_Q keeps track of what logic gate is the parent of the subcircuit representing the subformula ψ (i.e., $Q = \forall$ corresponds to an \wedge -gate as parent and $Q = \exists$ corresponds to an \vee -gate as parent), and increases the size only if the subcircuit cannot be integrated/merged with its parent gate. For example, if the output gate of the subcircuit is an \wedge -gate, and its parent is also an \wedge -gate, these two gates can be merged into a single large \wedge -gate.

► **Definition 1.** We define Linear Monadic NP to be the class of NP search problems such that each is expressible using an EMSO formula $\varphi(S)$ with $s(\varphi(S)) = \mathcal{O}(n)$.

The following theorem shows that LINEAR CIRCUIT-SAT can serve as the canonical satisfiability problem for Linear Monadic NP:

► **Theorem 2.** *Let A be a problem in Linear Monadic NP, and suppose that A is expressible using an EMSO formula $\varphi(S)$, where $s(\varphi(S)) = \mathcal{O}(n)$. Then A is self-reducible to LINEAR CIRCUIT-SAT, where the parameter of A is the size of the universe of the input structure, and that for LINEAR CIRCUIT-SAT is the number of variables of the circuit.*

Sketch of Proof. Let A be a search problem with input of the form (I, N) , where I is a relational structure with universe U , and $N \in \mathbb{N}$, such that A is expressed by the EMSO formula $\varphi(S) = \exists R_1, \dots, R_q \Phi(R_1, \dots, R_q, S)$, with $s(\varphi(S)) = \mathcal{O}(n)$.

We know that Φ consists of first-order formulas ψ_1, \dots, ψ_k —each starting with a first-order quantifier—that are combined using the connectives \wedge and \vee , and that $s(\psi_i) = \mathcal{O}(n)$ for each $i \in [k]$.

We reduce the problem to LINEAR CIRCUIT-SAT by transforming each such formula ψ_i into a sequence of (subexponentially many) circuits $C_{i,1}, \dots, C_{i,b_i}$ where each such circuit C_{i,b_i} is of size linear in $|U|$.

If ψ_i contains only universal first-order quantifiers, or only existential first-order quantifiers, constructing such a sequence of circuits can be done using known results [22, Theorems 1 and 2].

Otherwise, if ψ_i contains both universal and existential first-order quantifiers, we transform ψ_i into a single circuit C_i inductively, breaking the formula ψ_i into its subformulas and constructing subcircuits for each subformula.

Intuitively, this circuit C_i is constructed by ‘unfolding’ universal quantifiers into \wedge -gates and existential quantifiers into \vee -gates. Using the inductive definition of the measure s in Equations (1)–(12), and by our assumption that $s(\psi_i)$ is a linear function in n , we can derive that this construction yields a circuit C_i that is of size linear in $|U|$. ◀

► **Definition 3.** A problem A is *Linear Monadic NP-complete under serf-reducibility* if (1) it is in Linear Monadic NP and (2) every problem in Linear Monadic NP is serf-reducible to A .

► **Corollary 4.** LINEAR CIRCUIT-SAT is *Linear Monadic NP-complete under serf-reducibility*.

Note that all problems in Monadic SNP [22] are in Linear Monadic NP. This is because every EMSO expression (in negation normal form) that contains no first-order quantifier alternations has a linear measure—due to Equation (3). Because LINEAR CIRCUIT-SAT is serf-reducible to 3-CNF-SAT [24], Corollary 4 implies the following:

► **Corollary 5.** *Every Monadic SNP-complete problem is Linear Monadic NP-complete under serf-reducibility. Therefore, a Linear Monadic NP-complete problem is solvable in subexponential time if and only if ETH fails.*

The above implies that the well-known SNP-complete problems k -CNF-SAT ($k \geq 3$), c -COLORABILITY, INDEPENDENT SET, CLIQUE, and VERTEX COVER, among others, are Linear Monadic NP-complete. (Note that every problem that is complete for Linear Monadic NP under serf-reducibility is clearly hard for SNP under serf-reducibility.)

3.2 Applications: Expressing Natural Optimization Problems

We saw in the previous subsection that LINEAR CIRCUIT-SAT is in Linear Monadic NP. We prove in Subsection 3.3 that LINEAR CIRCUIT-SAT is not in monadic SNP. The same can be shown for problems that are serf-equivalent to LINEAR CIRCUIT-SAT, such as LINEAR HITTING SET and LINEAR SET COVER. In this subsection, we give several examples of natural graph problems that are in Linear Monadic NP, but are inexpressible in monadic SNP logic as will be shown in Subsection 3.3. The definition of these natural problems can be found in [14].

Our first example is DOMINATING SET, with the number of vertices as the complexity parameter. This problem can be expressed using the following formula $\varphi_{\text{DS}}(S)$:

$$\varphi_{\text{DS}}(S) = \forall x (S(x) \vee \exists y (S(y) \wedge E(x, y))).$$

From Equations (1)–(12), we get that $s(\varphi_{\text{DS}}(S)) = n + 1$. In effect:

$$\begin{aligned} s(\varphi_{\text{DS}}(S)) &= s[\forall x (S(x) \vee \exists y (S(y) \wedge E(x, y)))] &&= 1 + s_{\forall}[\forall x (S(x) \vee \exists y (S(y) \wedge E(x, y)))] \\ &= 1 + n \cdot s_{\forall}[(S(x) \vee \exists y (S(y) \wedge E(x, y)))] &&= 1 + n \cdot (1 + s_{\exists}[S(x)] + s_{\exists}[\exists y (S(y) \wedge E(x, y))]) \\ &= 1 + n \cdot (1 + 0 + n \cdot s_{\exists}[S(y) \wedge E(x, y)]) &&= 1 + n \cdot (1 + 0 + n \cdot s_{\exists}[S(y)]) \\ &= 1 + n \cdot (1 + 0 + n \cdot 0) &&= n + 1. \end{aligned}$$

It follows that DOMINATING SET is in Linear Monadic NP. Our second example is the RED-BLUE DOMINATING SET problem [11], with the number of vertices as the complexity parameter. The expression is $\varphi_{\text{RB-DS}}(S) = \forall x (\neg S(x) \vee R(x)) \wedge \forall y (\neg B(y) \vee \exists z (S(z) \wedge E(z, y)))$, for which it can be easily verified that $s(\varphi_{\text{RB-DS}}(S)) = 2n + 2$.

Similarly, we can express the problem NON-BLOCKER [14], with the number of vertices as complexity parameter, using the following formula $\varphi_{\text{NB}}(S) = \forall x (\neg S(x) \vee \exists y (\neg S(y) \wedge E(x, y)))$. It can be easily verified that $s(\varphi_{\text{NB}}(S)) = n + 1$, and hence NON-BLOCKER is in Linear Monadic NP.

The INDEPENDENT DOMINATING SET problem [11] is also in Linear Monadic NP because it can be expressed as a conjunction of the two expressions for DOMINATING SET and INDEPENDENT SET. Specifically, the formula $\varphi_{\text{IDS}}(S) = \varphi_{\text{DS}}(S) \wedge [\forall x \forall y (\neg E(x, y) \vee \neg S(x) \vee \neg S(y))]$ expresses the problem and $s(\varphi_{\text{IDS}}(S)) = 2n + 2$.

A similar expression to the above shows that the DOMINATING CLIQUE problem [9] is also in Linear Monadic NP.

Further examples of problems that are in Linear Monadic NP are DISTANCE- r DOMINATING SET [21], r -THRESHOLD DOMINATING SET [14], and r -DOMATIC PARTITION [32], for any integer-constant r .

We give the expression for r -DOMINATING SET below, whose measure can be verified to be $\mathcal{O}(n)$. The expressions for the other two problems are omitted.

$$\varphi_{r\text{-DS}}(S) = \forall x \exists y_1 \exists y_2 \dots \exists y_r [(x = y_1) \vee E(x, y_1)] \wedge \dots \wedge [(y_{r-1} = y_r) \vee E(y_{r-1}, y_r)] \wedge S(y_r).$$

Finally, the PERFECT CODE problem [11] is in Linear Monadic NP because it can be expressed using the following formula $\varphi_{\text{PC}}(S)$, for which it can be verified that $s(\varphi_{\text{PC}}(S)) = 3n + 3$:

$$\varphi_{\text{PC}}(S) = [\forall x \forall y (\neg E(x, y) \vee \neg S(x) \vee \neg S(y))] \wedge [\forall z (S(z) \vee \exists y (S(y) \wedge E(z, y)))] \wedge [\forall z \forall x \forall y (\neg S(x) \vee \neg S(y) \vee \neg E(x, z) \vee \neg E(y, z))].$$

Via standard reductions from 3-CNF-SAT, it can be easily shown that all the problems discussed above are complete for Linear Monadic NP under self-reducibility. Therefore, we have the following:

► **Corollary 6.** *For any of the problems DOMINATING SET, RED-BLUE DOMINATING SET, INDEPENDENT DOMINATING SET, DOMINATING CLIQUE, r -THRESHOLD DOMINATING SET, DISTANCE- r DOMINATING SET, r -DOMATIC PARTITION, and PERFECT CODE, the following holds: the problem is solvable in subexponential time if and only if ETH fails.*

3.3 Inexpressibility in Monadic SNP Logic

We prove in this subsection that DOMINATING SET is inexpressible in monadic SNP logic, and show how the proof can be straightforwardly adapted to other considered problems. In particular, we show that there exists no formula φ in this logic with one free set variable such that $(G, S) \models \varphi(S)$ if and only if S is a dominating set in G . This shows that the greater freedom in quantifier alternation offered by our new logic fragment is necessary to express this problem. Before we give a formal proof of this result, we will need several notions from the area of logic; we refer the reader to Libkin's book [26]. The key tool we use to prove inexpressibility is the so-called Ajtai-Fagin game [1, 26], defined below.

► **Definition 7.** Let \mathcal{P} be a property of graphs equipped with a single set, let Γ be a fragment of FO logic, and let $\ell, k \in \mathbb{N}$. Then the $(\mathcal{P}, \ell, k, \Gamma)$ -Ajtai-Fagin game is a 2-player game between the duplicator and the spoiler which proceeds in the following 4 steps:

1. The duplicator selects a graph G equipped with a set S such that $(G, S) \in \mathcal{P}$.
2. The spoiler selects ℓ subsets U_1, \dots, U_ℓ of $V(G)$.
3. The duplicator selects a graph G' equipped with a set S' and also ℓ subsets U'_1, \dots, U'_ℓ of $V(G')$.
4. The duplicator wins if and only if he can prove that $(G, S, U_1, \dots, U_\ell)$ and $(G', S', U'_1, \dots, U'_\ell)$ agree on $\Gamma[k]$.

► **Proposition 8** ([1, 26]). The duplicator has a winning strategy in the $(\mathcal{P}, \ell, k, \Gamma)$ -Ajtai-Fagin game if and only if \mathcal{P} is not definable by any formula with a single free set variable S of the form $\exists X_1, \dots, \exists X_\ell \chi(S, X_1, \dots, X_\ell)$, where χ is a formula in Γ .

The final ingredient we need is a result which links the number of quantifier alternations in an FO formula to the moves of the Ehrenfeucht-Fraïssé game. The j -alternations k -round Ehrenfeucht-Fraïssé game is a restriction of the standard Ehrenfeucht-Fraïssé game to at most j alternations (“switches”) of the structure where the spoiler makes his moves. As a special case of a result by Pezzoli [31], it follows that the duplicator wins the 0-alternations k -round Ehrenfeucht-Fraïssé game on $(\mathcal{A}, \mathcal{B})$ if and only if \mathcal{A} and \mathcal{B} agree on all universal first-order formulas of quantifier rank at most k . We are now ready to prove our inexpressibility result.

► **Lemma 9.** DOMINATING SET *is inexpressible in monadic SNP logic.*

Sketch of Proof. We use the Ajtai-Fagin game and describe a winning strategy for the duplicator for every fixed $\ell, k \in \mathbb{N}$. In Step 1, the duplicator selects a graph G which consists of $2^{2\ell} \cdot k + 1$ copies of K_2 , and a set S which contains a single vertex from each copy of K_2 ; observe that S is a dominating set of G . In Step 2, the spoiler arbitrarily selects his subsets U_1, \dots, U_ℓ of vertices of G .

Before proceeding to Step 3, we need to find a “sufficiently frequent” configuration in (G, U_1, \dots, U_ℓ) for the duplicator to exploit. Let the *type* $T(v)$ of a vertex $v \in G$ be defined as $T(v) = \{i \mid v \in U_i\}$. Let a, b be vertices of G which form a K_2 such that $b \in S$. Then the *configuration* $C(a, b)$ of a, b is the tuple $(T(a), T(b))$. Since the number of distinct types is upper-bounded by 2^ℓ , the number of distinct configurations is upper-bounded by $2^{2\ell}$. By construction, there must exist some configuration which occurs at least $k + 1$ times in G ; let us now fix an arbitrary such configuration (T_1, T_2) .

In Step 3, the duplicator selects a graph H equipped with a set Q . The graph H consists of a copy of G and one isolated vertex p . Let f be a bijection witnessing the isomorphism from $H - p$ to G ; then the duplicator selects H and also the vertex-subsets W_1, \dots, W_ℓ of H as follows: (1) for each vertex $w \in H - p$, $w \in Q$ if and only if $f(w) \in S$; (2) for each vertex $w \in H - p$ and each $i \in [\ell]$, $w \in W_i$ if and only if $f(w) \in U_i$; and (3) $p \in W_i$ if and only if $i \in T_1$.

In Step 4, it suffices to prove that $(G, S, U_1, \dots, U_\ell)$ and $(H, Q, W_1, \dots, W_\ell)$ agree on all first-order formulas with alternation number 0 and quantifier rank at most k . This is done by giving a winning strategy for the duplicator in the 0-alternations k -round Ehrenfeucht-Fraïssé game. On a high level, if the spoiler chooses to play on G , then the duplicator can precisely copy the spoiler’s moved on H . On the other hand, if the spoiler decides to play on H then the duplicator can also precisely copy all spoiler’s moves on G with the exception of a move on p ; there, the duplicator uses the fact that the configuration (T_1, T_2) is sufficiently frequent to find a suitable replacement for p in G . ◀

The same technique can be applied to prove the inexpressibility of other problems considered in Subsection 3.2. For some, it suffices to merely adapt the above construction, while for others, such as LINEAR CIRCUIT-SAT, new constructions are required.

4 Feedback Vertex Set

In the previous section, we gave a logic fragment such that every problem that is expressible in this logic fragment is serf-reducible to 3-CNF-SAT. A natural question to ask is whether there exist NP-complete problems that are not serf-reducible to 3-CNF-SAT, under some plausible complexity-theoretic hypothesis. This question has been answered by several works. For instance, Calabro *et al.* [4] showed that, unless ETH fails, the restriction of CNF-SAT to instances in which the number of clauses is super-linear in the number of variables is not serf-reducible to 3-CNF-SAT. More unlikely complexity-theoretic consequences befall if we replace CNF-SAT restricted to instances with super-linear number of clauses with “harder” satisfiability problems (*e.g.*, general CNF-SAT or CIRCUIT-SAT). While the aforementioned problems are all expressible in EMSO, their expressions yield super-linear measures.

We also saw in the previous section that many natural graph problems, such as c -COLORABILITY (for any integer-constant $c > 0$), INDEPENDENT SET, VERTEX COVER and DOMINATING SET are serf-reducible to 3-CNF-SAT, which raises the question of whether there is any natural graph problem that is not serf-reducible to 3-CNF-SAT. While it is not difficult to define a graph problem that, unless ETH fails, is not serf-reducible to 3-CNF-SAT, we do not know of any natural graph problem for which the aforementioned statement can be proved. In this section, we propose FEEDBACK VERTEX SET as a possible such candidate problem. The reason we believe that FEEDBACK VERTEX SET might be such a problem is that FEEDBACK VERTEX SET implicitly embodies a HITTING SET problem [5] (or a satisfiability problem) with possibly exponentially-many cycles to hit. Whereas in CNF-SAT and in HITTING SET the sets/clauses are given explicitly (*i.e.*, are part of the input), which allows us to quantify over the set of all sets/clauses, and hence, express these problems in monadic NP, the cycles in an instance of FEEDBACK VERTEX SET are implicitly encoded in the input graph. Moreover, enumerating all the cycles in a graph may require exponential time, which surpasses the allowed time in a serf-reduction.

While we are unable to prove that FEEDBACK VERTEX SET is not serf-reducible to 3-CNF-SAT (assuming ETH does not fail), we could prove that FEEDBACK VERTEX SET is inexpressible in EMSO, which rules out the possibility of using the proposed framework in this paper to show that FEEDBACK VERTEX SET is serf-reducible to 3-CNF-SAT. We leave it as an open problem whether FEEDBACK VERTEX SET is serf-reducible to 3-CNF-SAT. We also show in this section that there is a polynomial-time reduction from FEEDBACK VERTEX SET to 3-CNF-SAT that maps an instance of FEEDBACK VERTEX SET with n vertices to an equivalent instance of 3-CNF-SAT with $\mathcal{O}(n \lg n)$ variables, and define a variant of FEEDBACK VERTEX SET that is equivalent to 3-CNF-SAT under serf-reducibility.

4.1 Inexpressibility of FEEDBACK VERTEX SET in EMSO

Recall that, by Theorem 2, we could obtain a serf-reduction from FEEDBACK VERTEX SET to 3-CNF-SAT if there existed an EMSO formula $\varphi(X)$ with linear measure such that $(G, S) \models \varphi(S)$ if and only if S is a feedback vertex set in G . In this subsection, we will prove that this approach cannot work; in particular, we show that feedback vertex set is not even expressible in EMSO. Our first step lies in showing that acyclicity is inexpressible by an EMSO sentence. However, instead of using the Ajtai-Fagin game (which would require a highly technical specification of a strategy for the duplicator), our proof will rely on the classical notion of *Hanf-locality*. We refer to Libkin’s book [26] for the omitted definitions.

► **Lemma 10.** *Acyclicity is inexpressible in EMSO.*

Sketch of Proof. Suppose for a contradiction that acyclicity is definable by an EMSO sentence $\Psi = \exists Z_1 \dots Z_\ell \psi$. Since ψ is a first-order sentence, it is Hanf-local. Let G be a path of length at least r . Since G is acyclic, we have $G \models \Psi$. Let U_1, \dots, U_m witness this fact, that is, $(G, U_1, \dots, U_m) \models \psi$. Let a, b be vertices of distance at least $2d + 2$ from each other such that $N_d^{(G, U_1, \dots, U_m)}(a)$ is isomorphic to $N_d^{(G, U_1, \dots, U_m)}(b)$. Let p be the unique endpoint of the path that is closer to b than to a , and let a' be the unique neighbor of a which is closer to p (than the other neighbor of a), and let b' be that of b . We construct a new graph G' by removing edges aa' and bb' from G and adding edges ab' and ba' into G . We have that, for every vertex c , $N_d^{(G, U_1, \dots, U_m)}(c) \approx N_d^{(G', U_1, \dots, U_m)}(c)$. Observe that G' is not acyclic. To complete the proof, we use the Hanf-locality to prove that G' also satisfies Ψ . ◀

► **Lemma 11.** FEEDBACK VERTEX SET is inexpressible in EMSO logic.

4.2 Reductions between 3-CNF-SAT and FEEDBACK VERTEX SET

We now examine the existence of polynomial-time self-reductions between 3-CNF-SAT and FEEDBACK VERTEX SET. Based on the framework in [13], we can show that it is unlikely that 3-CNF-SAT is polynomial-time self-reducible to FEEDBACK VERTEX SET:

► **Proposition 12.** Unless the polynomial-time hierarchy collapses to its third level, 3-CNF-SAT is not polynomial-time self-reducible to FEEDBACK VERTEX SET.

While we are unable to rule out—under plausible conditions—the existence of a polynomial-time self-reduction from FEEDBACK VERTEX SET to 3-CNF-SAT, we give a polynomial-time reduction from FEEDBACK VERTEX SET to 3-CNF-SAT at the cost of a logarithmic factor increase in the number of variables in the 3-CNF-SAT instances. This reduction relies on the following characterization of FEEDBACK VERTEX SET that we prove: A graph G has a feedback vertex set of cardinality k if and only if there is a bijection $\varphi : V(G) \rightarrow [n]$ such that, for all $uv, uw \in E(G)$, we have

$$(\varphi(u) > k) \wedge (\varphi(v) > k) \wedge (\varphi(w) > k) \implies (\varphi(v) > \varphi(u)) \vee (\varphi(w) > \varphi(u)).$$

Using the above characterization, we can reduce FEEDBACK VERTEX SET to 3-CNF-SAT by encoding each vertex using a block of $\lg n$ Boolean variables, and adding polynomially-many 3-CNF clauses, each of logarithmic width, encoding the existence of a function φ satisfying the above property. We have:

► **Theorem 13.** There is a polynomial-time many-one reduction that takes an instance (G, k) of FEEDBACK VERTEX SET and produces an equivalent instance F of CNF-SAT such that F has $\mathcal{O}(n \lg n)$ variables, $n^{\mathcal{O}(1)}$ clauses, and width $\mathcal{O}(\lg n)$, where $n = n(G)$.

We now define a variant of FEEDBACK VERTEX SET, denoted MONOCHROMATIC 3-FEEDBACK VERTEX SET, that we show to be equivalent under self-reducibility to 3-CNF-SAT. An instance of MONOCHROMATIC 3-FEEDBACK VERTEX SET consists of a graph G and a nonnegative integer k . Each edge $e \in E(G)$ is associated with a set of colors, denoted $\text{Colors}(e)$. The question is to decide if there exists a subset $Q \subseteq V(G)$ of vertices of cardinality at most k such that, for every 3-cycle C in G satisfying $\bigcap_{e \in E(C)} \text{Colors}(e) \neq \emptyset$, C contains at least one vertex of Q . That is, Q breaks every 3-cycle in G whose edges can all be assigned the same color from their color lists. It can be easily shown that MONOCHROMATIC 3-FEEDBACK VERTEX SET is NP-complete by adapting the standard reduction from VERTEX COVER to FEEDBACK VERTEX SET (the color lists of all edges are singletons and consist of the same color). By d -MONOCHROMATIC 3-FEEDBACK VERTEX SET, where $d \geq 1$, we

denote the restriction of MONOCHROMATIC 3-FEEDBACK VERTEX SET to instances in which the total number of colors assigned is at most d (that is, $|\bigcup_{e \in E(G)} \text{Colors}(e)| \leq d$). We have:

► **Theorem 14.** *For any integer $d \geq 10$, 3-CNF-SAT and d -MONOCHROMATIC 3-FEEDBACK VERTEX SET are equivalent under serf-reductions. Therefore, d -MONOCHROMATIC 3-FEEDBACK VERTEX SET is solvable in subexponential time if and only if ETH fails.*

► **Remark.** A natural question to ask is whether serf-reducibility to 3-CNF-SAT is equivalent to the notion of self-sparsification (as in the sparsification lemma for 3-CNF-SAT): The existence of a subexponential-time self-reduction that reduces an instance of the problem to an equivalent instance whose (instance) size is linear in the designated parameter. These two notions, in general, seem to be orthogonal. On one hand, it can be shown that FEEDBACK VERTEX SET is self-sparsifiable [18, 27], but this does not seem to imply that FEEDBACK VERTEX SET is serf-reducible to 3-CNF-SAT. On the other hand, CLIQUE is serf-reducible to 3-CNF-SAT, but unless ETH fails, CLIQUE is not self-sparsifiable since this would imply that CLIQUE is solvable in subexponential time.

References

- 1 Miklós Ajtai and Ronald Fagin. Reachability is harder for directed than for undirected finite graphs. *J. Symb. Log.*, 55(1):113–150, 1990.
- 2 Miklós Ajtai, Ronald Fagin, and Larry J. Stockmeyer. The closure of monadic NP. *J. Comput. Syst. Sci.*, 60(3):660–716, 2000.
- 3 Sanjeev Arora and Ronald Fagin. On winning strategies in Ehrenfeucht-Fraïssé games. *Theor. Comput. Sci.*, 174(1-2):97–121, 1997.
- 4 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 252–260. IEEE Computer Society, 2006.
- 5 Karthekeyan Chandrasekaran, Richard M. Karp, Erick Moreno-Centeno, and Santosh Vempala. Algorithms for implicit hitting set problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 614–629. SIAM, 2011.
- 6 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David Juedes, Iyad Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation*, 201(2):216–231, 2005.
- 7 Jianer Chen, Xiuzhen Huang, Iyad Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. of Computer and System Sciences*, 72(8):1346–1367, 2006.
- 8 Jianer Chen, Iyad Kanj, and Ge Xia. On parameterized exponential time complexity. *Theoretical Computer Science*, 410(27-29):2641–2648, 2009.
- 9 Margaret B. Cozzens and Laura L. Kelleher. Dominating cliques in graphs. *Discrete Mathematics*, 86(1-3):101–116, 1990.
- 10 Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socala. Tight bounds for graph homomorphism and subgraph isomorphism. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1643–1649, 2016.
- 11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 12 Evgeny Dantsin, Andreas Goerdt, Edward A. Hirsch, Ravi Kannan, Jon M. Kleinberg, Christos H. Papadimitriou, Prabhakar Raghavan, and Uwe Schöning. A deterministic

- $(2 - 2/(k+1))^n$ algorithm for k -SAT based on local search. *Theoretical Computer Science*, 289(1):69–83, 2002.
- 13 Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.
 - 14 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, New York, 1999.
 - 15 Ronald Fagin. Generalized first-order spectra, and polynomial. time recognizable sets. *SIAM-AMS Proceedings*, 7:43–73, 1974.
 - 16 Ronald Fagin, Larry J. Stockmeyer, and Moshe Y. Vardi. On monadic NP vs. monadic co-NP. *Information and Computation*, 120(1):78–92, 1995.
 - 17 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer Verlag, Berlin, 2006.
 - 18 Fedor V. Fomin, Serge Gaspers, Artem V. Pyatkin, and Igor Razgon. On the minimum feedback vertex set problem: Exact and enumeration algorithms. *Algorithmica*, 52(2):293–307, 2008.
 - 19 Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Springer Verlag, 2010.
 - 20 Michael R. Garey and David R. Johnson. *Computers and Intractability*. W. H. Freeman and Company, New York, San Francisco, 1979.
 - 21 Adriana Hansberg, Dirk Meierling, and Lutz Volkmann. Distance domination and distance irredundance in graphs. *Electr. J. Comb.*, 14(1), 2007.
 - 22 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. of Computer and System Sciences*, 63(4):512–530, 2001.
 - 23 David Janin and Jerzy Marcinkowski. A toolkit for first order extensions of monadic games. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer*, volume 2010 of *Lecture Notes in Computer Science*, pages 353–364. Springer, 2001.
 - 24 Iyad Kanj and Stefan Szeider. Parameterized and subexponential-time complexity of satisfiability problems and applications. *Theoretical Computer Science*, 607:282–295, 2015.
 - 25 Martin Kreidler and Detlef Seese. Monadic NP and graph minors. In *Proceedings of the 12th International Workshop on Computer Science Logic*, volume 1584 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 1998.
 - 26 Leonid Libkin. *Elements of Finite Model Theory*. Springer Verlag, 2004.
 - 27 Daniel Lokshtanov. Personal communication, 2015.
 - 28 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the European Association for Theoretical Computer Science*, 105:41–72, 2011.
 - 29 Dániel Marx. Can you beat treewidth? *Theory of Computing*, 6:85–112, 2010.
 - 30 Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. of Computer and System Sciences*, 43(3):425–440, 1991.
 - 31 Elena Pezzoli. Computational complexity of ehrenfeucht-fraïssé games on finite structures. In *Computer Science Logic, 12th International Workshop, CSL '98, Annual Conference of the EACSL, Brno, Czech Republic, August 24-28, 1998, Proceedings*, pages 159–170, 1998.
 - 32 Sheung-Hung Poon, William Chung-Kung Yen, and Chin-Ting Ung. Domatic partition on several classes of graphs. In *Combinatorial Optimization and Applications - 6th International Conference, COCOA 2012, Banff, AB, Canada, August 5-9, 2012. Proceedings*, pages 245–256, 2012.