Theoretical Computer Science 802 (2020) 141–146

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

One-variable logic meets Presburger arithmetic

Bartosz Bednarczyk

Institute of Computer Science, University of Wrocław, Joliot-Curie 15, 50-383 Wrocław, Poland

ARTICLE INFO

Article history:
Received 25 October 2018
Received in revised form 16 September 2019
Accepted 17 September 2019
Available online 23 September 2019
Communicated by D. Sannella

Keywords:
Finite satisfiability
Computational complexity
Decidability
Classical decision problem
Arithmetics

ABSTRACT

We consider the one-variable fragment of first-order logic extended with Presburger constraints. The logic is designed in such a way that it subsumes the previously-known fragments extended with counting, modulo counting or cardinality comparison and combines their expressive powers. We prove NP-completeness of the logic by presenting an optimal algorithm for solving its finite satisfiability problem.

1. Introduction

It is well-known that first-order logic FO cannot describe natural quantitative properties like parity or equicardinality of sets. To solve this problem one can think about enlarging the language with special constructs, e.g., generalized quantifiers like counting quantifiers, modulo counting quantifiers, majority quantifiers or the Härtig quantifier. However additional expressive power often comes with an increase in computational complexity. For example consider the two-variable fragment of first order logic, FO^2. It is known that FO^2 becomes undecidable when cardinality comparison via Härtig or Rescher quantifiers is allowed [8]. On the other hand its extension with counting quantifiers is decidable [7,12]. The decidability status of FO^2 with modulo counting quantifiers is currently unknown. Thus there is no hope to obtain a decidable extension of FO^2 which allows all of these features.

In this paper we take a closer look at the one-variable fragment of first-order logic, denoted here by FO^1. The logic is well-understood and its finite satisfiability is known to be only NP-complete. We are aware of three extensions of FO^1 that differ in expressive power: C^1, FO^1_{MOD} and $\mathcal{L}_1[I]$, see e.g. [14,3,8]. The mentioned logics extend FO^1 with counting quantifiers $\exists^{\geq k}$, modulo-counting quantifiers $\exists^{=a (\mathrm{mod}\ b)}$ and the so-called Härtig quantifier I, respectively. The semantics of the first two logics is very intuitive. For the third logic we define $I(\varphi, \psi)$ to be true if the total number of elements satisfying the formula φ is the same as the total number of elements satisfying ψ. It follows from [14] and [3] that the finite satisfiability problem for C^1 and FO^1_{MOD} is NP-complete, even when the numbers in quantifiers are written in binary. Moreover, a practical algorithm for deciding satisfiability of a fragment of C^1 was implemented and tested in [6]. For the third logic, namely $\mathcal{L}_1[I]$ from [8], the authors of the paper stated that the logic is decidable but no proof or complexity bounds were given.

E-mail address: bartosz.bednarczyk@cs.uni.wroc.pl.

https://doi.org/10.1016/j.tcs.2019.09.028

1.1. Our contribution

In this article we present a novel logic P_1 which subsumes previously known logics with counting or cardinality comparison, i.e., C^1, $\mathrm{FO}^1_{\mathrm{MOD}}$ and $\mathcal{L}_1[\mathrm{I}]$ from [14,3,8]. Moreover the logic allows one to express percentage constraints. As an example we can consider a property that the majority of elements of a model satisfies a given formula φ.

We obtain a tight NP upper bound for P_1. The proof goes via a translation of formulae into a system of inequalities and closely follows the techniques presented in [14]. However some technical details differ. As a by-product we fill a gap concerning the complexity of $\mathcal{L}_1[\mathrm{I}]$.

1.2. Our motivations

Our main motivation is to see what the scope of the technique of Pratt-Hartmann [14] is for deciding finite satisfiability for C^1. Moreover, we would like to see how powerful a logic we can obtain while keeping the complexity reasonably low. Last but not least, the proposed logic P_1 is the core part of Presburger Modal Logic [5] and its NP-completeness can be used to establish complexities for reasoning tasks of the family of Euclidean Presburger Modal Logics. A slight generalization of the translation from [9] shows that their local and global satisfiability can be easily reduced to the finite satisfiability of P_1.

We recently learned about the existence of QFBAPA [10], the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic. The logics can express similar properties and share the same complexity of satisfiability, namely NP-completeness. Nevertheless, we strongly believe that the logic P_1 is a very natural logic, arguably more elegant than QFBAPA, and the proof technique used here is much easier to understand.

2. Preliminaries

We employ the standard terminology from model theory and linear algebra. We refer to structures with fraktur letters, and to their universes with the corresponding Roman letters. We always assume that structures have non-empty universes. Here we are interested in *finite* structures over a countable *signature* Σ consisting of *unary* relational symbols only.

Let $\mathcal{L}$ be an arbitrary logic. In the *finite satisfiability problem* for the logic $\mathcal{L}$ we ask whether an input formula φ from $\mathcal{L}$ is finitely satisfiable, i.e., has a finite *model*.

2.1. Linear algebra and integer linear programming

By $\mathbb{Z}_n$ we denote the set of all remainders modulo n, that is the set $\{0, 1, \ldots, n-1\}$.

A *linear inequality* is an expression of the form $t \geq t'$, where t and t' are linear terms. In this paper we are interested only in linear inequalities with integer coefficients (written in binary). It is well known that solving systems of such inequalities over $\mathbb{N}$ is in NP [4].

The following "sparse solution" lemma provides an upper bound on the minimum number of non-zero unknowns in solutions of systems of linear inequalities:

Lemma 1 *([1]). Let $\mathcal{E}$ be a system of I inequalities with integer coefficients such that the absolute value of each coefficient from $\mathcal{E}$ is bounded by C. If $\mathcal{E}$ has a solution over $\mathbb{N}$, then it has also a solution over $\mathbb{N}$ with the number of non-zero unknowns bounded by $2I\log\left(2C\sqrt{I}\right)$.*

2.2. Syntax of the logic P_1

In this article we propose an extension of a function-free one-variable fragment of first-order logic with *counting terms* and *Presburger constraints*. We let P_1 denote the formalism.

The main ingredients of formulae of P_1 are counting terms t_x [8]. Their intuitive role is to count the total number of witnesses of a given formula featuring a single variable x. Such terms can be multiplied by integer constants and added to each other. On the top level we allow for the comparison of values of counting-terms with a given threshold using a greater-than operator $\geq$ and to test congruence modulo some number k using $\equiv_k$. More general formulae can be constructed with Boolean connectives and by means of nesting.

The minimal syntax of the logic P_1 is given by the following BNF grammar:

$$t_x ::= t_x + t_x \mid a \cdot \sharp_x[\varphi(x)]$$
$$\varphi ::= P(x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid t_x \geq b \mid t_x \equiv_c d$$

where $P \in \Sigma$ is a unary relational symbol, $a \in \mathbb{Z} \setminus \{0\}$ is a non-zero integer, $b \in \mathbb{N}$ is a natural number, $c \in \mathbb{Z}_+$ is a positive integer and $d \in \mathbb{Z}_c$ is a remainder modulo c.

A counting term of the form $a_1 \cdot \sharp_x[\varphi_1] + \ldots + a_n \cdot \sharp_x[\varphi_n]$ is abbreviated by $\Sigma_{i=1}^{n} a_i \cdot \sharp_x[\varphi_i]$. Note that all standard logical connectives such as $\vee$, $\rightarrow$, $\leftrightarrow$ as well as other (in)equality symbols like $<, >, \leq$ and $=$ can be easily defined using Boolean combinations and constants. Hence, we will use them as abbreviations.

We write $|\varphi|$ to denote the length of a formula φ, i.e., the number of bits required to encode φ as a string. We will assume that all numbers appearing in φ are written in binary.

2.3. Semantics of the logic P_1

The semantics of the logic P_1 is a straightforward extension of the semantics of first-order logic. For formulae φ not involving counting terms, the semantics $[\![\varphi]\!]^{\mathfrak{M}}$ of φ in a model $\mathfrak{M}$ is the same as in first-order logic. We extend it to counting terms by defining $[\![\sharp_x[\varphi(x)]]\!]^{\mathfrak{M}}$ to be the cardinality of the set $\{a \in M \mid \mathfrak{M} \models \varphi[a]\}$. Addition, multiplication by a constant and comparison are treated in the obvious way.

2.4. Expressive power

We note here that P_1 trivially extends the one-variable fragment of first-order logic. Moreover, the logic can capture a scenario of threshold counting $\exists^{\geq k}\varphi(x)$ (i.e., C^1 from [14]) as well as modulo counting $\exists^{=a(\text{mod } b)}\varphi(x)$ (i.e., $\mathrm{FO}^1_{\mathrm{MOD}}$ from [3]). The logic also allows cardinality comparison, i.e., it can simulate the so-called Härtig and Rescher quantifiers from [8] and percentage constraints, e.g. $\mathrm{I}x.(\varphi(x), \psi(x))$ can be encoded as $\sharp_x[\varphi(x)] - \sharp_x[\psi(x)] = 0$. Hence P_1 can even express some second-order properties.

2.5. Types and normal forms

Let τ be a finite signature, and following a standard terminology, we define an atomic 1-type over τ as a maximal satisfiable set of atoms or negated atoms involving only the variable x. Usually we identify a 1-type with the conjunction of all its elements. We note here that the number of all atomic 1-types is exponential in the size of τ.

When a formula φ is fixed, we often refer to its signature (i.e., the set of unary symbols occurring in φ) with τ_φ. Then, the set of all 1-types over τ_φ is denoted by tp_φ and we refer to its elements with $\pi_1^{\tau_\varphi}, \pi_2^{\tau_\varphi}, \ldots, \pi_{|tp_\varphi|}^{\tau_\varphi}$. Additionally, when both a model $\mathfrak{M}$ and a 1-type π are fixed, we define $|\pi|_{\mathfrak{M}}$ as the total number of elements from a structure M satisfying a 1-type π.

Definition 1. *We say that a formula $\varphi \in \mathrm{P}_1$ is* flat, *if:*

$$\varphi = \bigwedge_{i=1}^{n} \left(\sum_{j=0}^{n_i} a_{i,j} \cdot \sharp_x[\varphi_{i,j}] \right) \bowtie_i b_i$$

where $\bowtie_i$ is a comparison symbol, i.e., $\bowtie_i \in \{\leq, \geq, \equiv_k \mid k \in \mathbb{N}\}$, each $a_{i,j} \in \mathbb{Z} \setminus \{0\}$ is a non-zero integer, each $b_i \in \mathbb{N}$ is a natural number and all formulae $\varphi_{i,j}$ are free of counting terms.

The main purpose of introducing a flat form for P_1 formulae is to avoid nesting of counting terms and to simplify reasoning about satisfaction of a formula. The following lemma shows that every satisfiable P_1 formula can be flattened in NP:

Lemma 2. *There exists a non-deterministic polynomial time procedure, taking as its input a P_1 formula over a signature τ and producing a flat formula φ' over the same signature τ, such that φ is satisfiable iff the procedure has a run producing a satisfiable φ'.*

Sketch of proof. The proof goes in a standard fashion, similarly to the proof of Theorem 1 in [14]. The main idea of the algorithm is to take the innermost expression e, from the original formula φ, of the form $\Sigma a_i \sharp_x[\varphi_i] \geq a$ or $\Sigma a_i \sharp_x[\varphi_i] \equiv_b c$. Since we are designing an NP procedure and an expression e speaks only globally about the total number of elements, we can guess whether e is satisfied or not. Then depending on a guess we replace e with $\top$ or $\bot$ and we put, respectively, e or $\neg e$ in front of the formula. Additionally, in the case when $\neg e$ contains a modulo constraint, we guess a proper remainder c' and replace $\neg \Sigma a_i \sharp_x[\varphi_i] \equiv_b c$ with $\Sigma a_i \sharp_x[\varphi_i] \equiv_b c'$. We repeat the whole process until we obtain a flat formula. □

3. The finite satisfiability of P_1

In this section we will show that the one-variable fragment of first-order logic remains NP-complete even if we extend it with Presburger constraints. As we mentioned in the beginning of the paper, we are interested only in finite models since e.g. modulo constraints do not make sense over infinite structures. Our proof will strongly rely on techniques presented in [14], namely reducing our problem to integer linear programming.

3.1. Overview of the method

Throughout this section, we fix a satisfiable P_1 formula φ. Due to Lemma 2 we can always produce a flat version of φ, thus we assume that φ is flat.

We will first sketch our approach. A crucial observation leading to a simple description of P_1 models is that the logic cannot speak about any kind of connection between two distinct elements of a model. Thus any model $\mathfrak{M}$ of φ can be described up to isomorphism by the information about the total number of elements of given 1-types. We call such information a *characteristic vector* χ_φ. It could be defined in the following way:

$$\chi_\varphi \stackrel{\text{def}}{=} \left(|\pi_0^{\tau_\varphi}|_{\mathfrak{M}},\ |\pi_1^{\tau_\varphi}|_{\mathfrak{M}},\ \ldots,\ |\pi_{|tp_\varphi|}^{\tau_\varphi}|_{\mathfrak{M}}\right),$$

where the i-th element of χ_φ is simply the total number of elements from M of the i-th 1-type.

Our goal is to translate a formula φ into a system of inequalities and congruences $\mathcal{E}$, whose solution will be a tuple χ_φ. Then, we will get rid of congruences, i.e., replace each of them with inequalities, at the expense of introducing polynomially many fresh variables. The obtained system $\mathcal{E}'$, as well as some of its coefficients, will be exponential due to the binary encoding of numbers. Since integer linear programming is in NP [4] we will obtain an NExpTime upper bound. To improve the complexity of the algorithm, we will use Lemma 1, which states that if there is a solution for $\mathcal{E}$, there is also a "sparse" solution, i.e., assigning only polynomially many non-zero values to unknowns.

It is worth pointing out that due to the presence of exponential coefficients we cannot easily adapt the lemma about small solutions from [14]. The technique we use, namely Lemma 1, is more sophisticated and requires a more difficult proof. We will use it as a black box.

3.2. A translation into a system of inequalities and congruences

We are going to describe a potential model $\mathfrak{M}$ of the formula φ in terms of unknowns and inequalities. In the desired system of inequalities, we will have exponentially many variables x_k, where each x_k corresponds to $|\pi_k|_{\mathfrak{M}}$ in a characteristic vector and each inequality or congruence corresponds to a threshold given in some conjunct from φ.

Let φ_i be the i-th conjunct from φ, i.e., $\varphi_i = \left(\sum_{j=0}^{n_i} a_{i,j} \cdot \sharp_x[\varphi_{i,j}]\right) \bowtie_i b_i$. Then, for every 1-type π_k we will associate an indicator $\mathbb{1}_{i,j,k}$, whose intuitive role will be to tell us whether the k-th type π_k is compatible with the formula $\varphi_{i,j}$. More formally:

$$\mathbb{1}_{i,j,k} = \begin{cases} 1, & \text{if } \models \pi_k \rightarrow \varphi_{i,j} \\ 0, & \text{otherwise} \end{cases}$$

With the above definition it is not hard to see that the value of a counting term $\sharp_x[\varphi_{i,j}]$ is equal to $\Sigma_{k=1}^{|tp_\varphi|} \mathbb{1}_{i,j,k} \cdot x_k$. By multiplying such value with constants $a_{i,j}$ and summing it over j, the whole formula φ_i can be represented as the following inequality or congruence:

$$\left(\sum_{j=0}^{n_i} a_{i,j} \cdot \left(\Sigma_{k=1}^{|tp_\varphi|} \mathbb{1}_{i,j,k} \cdot x_k\right)\right) \bowtie_i b_i$$

After rearranging the left-hand side of the above expression, we obtain a linear term with unknowns $x_1, x_2, \ldots, x_{|tp_\varphi|}$. Note that coefficients in front of variables x_k are exponential due to the binary encoding. We construct a system of inequalities and congruences $\mathcal{E}_\varphi$ by translating each conjunct φ_i from φ in the presented way.

The following lemma follows directly from the fact that each model $\mathfrak{M}$ of P_1 formula can be described up to isomorphism by a characteristic vector and from the construction of $\mathcal{E}_\varphi$.

Lemma 3. *Each solution of $\mathcal{E}_\varphi$ is a characteristic vector of some model $\mathfrak{M}$ of a P_1 formula φ.*

3.3. Getting rid of congruences

The obtained system $\mathcal{E}_\varphi$ can still contain linear terms with congruences. We will show a way how to replace them with inequalities. Let us assume that the i-th equation of the system $\mathcal{E}_\varphi$ is a congruence of the following form:

$$a_1^i \cdot x_1 + a_2^i \cdot x_2 + \ldots + a_{|tp_\varphi|}^i \cdot x_{|tp_\varphi|} \equiv_{k_i} b_i$$

For any natural number S_i, there exists a *remainder* $r_i \in \mathbb{Z}_{k_i}$ and a *quotient* $q_i \in \mathbb{N}$, such that $S_i = r_i + q_i k_i$. Thus we only need to ensure that the remainder r_i is equal to b_i. Since we do not know the precise value of the quotient q_i, we introduce a fresh variable y_i to represent it. We can rewrite the above congruence as $\sum_{j=1}^{|tp_\varphi|} a_j^i = b_i + k_i \cdot y_i$, which is equivalent to:

$$a^i_1 \cdot x_1 + a^i_2 \cdot x_2 + \ldots + a^i_{|tp_\varphi|} \cdot x_{|tp_\varphi|} - b_i - k_i \cdot y_i \leq 0,$$
$$a^i_1 \cdot x_1 + a^i_2 \cdot x_2 + \ldots + a^i_{|tp_\varphi|} \cdot x_{|tp_\varphi|} - b_i - k_i \cdot y_i \geq 0,$$

Let $\mathcal{E}'_\varphi$ be the system of inequalities obtained from $\mathcal{E}_\varphi$ by exhaustive elimination of all congruences. Since each step of the "congruence-elimination" procedure described above is sound, together with Lemma 3 we establish:

Lemma 4. *Each solution of $\mathcal{E}'_\varphi$ is a characteristic vector of some model $\mathfrak{M}$ of a* P_1 *formula φ.*

One can observe that the number of equations in $\mathcal{E}'_\varphi$ is bounded by $2n$ (i.e., where n is the number of conjuncts from flat φ), which is clearly of polynomial size in $|\varphi|$. Integer coefficients of the system $\mathcal{E}'_\varphi$ can be bounded by the sum of the absolute values of the numbers occurring in the formula φ. Since every number can be exponential in $|\varphi|$ (due to the binary encoding) and the mentioned sum contains at most polynomially many elements, we can conclude that each coefficient from the system $\mathcal{E}'_\varphi$ is bounded exponentially in $|\varphi|$.

3.4. Algorithm

By using Lemma 1 we know that the minimum number of non-zero unknowns in a sparse solution of $\mathcal{E}'_\varphi$ can be bounded by a polynomial function of $|\varphi|$. Hence we non-deterministically guess which unknowns will be non-zero and we construct a corresponding system $\mathcal{E}''_\varphi$ directly for them. The obtained system has polynomial size in $|\varphi|$, thus it is solvable in NP.

Below we present a non-deterministic polynomial time algorithm for testing whether a given P_1 formula has a finite model.

Procedure 1: Satisfiability test for P_1.

Input: A formula $\varphi \in \mathrm{P}_1$
guess φ' – a flat version of φ `// in NP, Lemma 2`
guess which 1-types are realized at least once. `// polynomially many, Lemma 1`
Write the system of inequalities $\mathcal{E}''_\varphi$ for the guessed 1-types. `// of poly size`
Return *True* iff $\mathcal{E}''_\varphi$ has a solution over $\mathbb{N}$. `// in NP [4]`

To ensure the correctness of the algorithm, we prove the following lemma:

Lemma 5. *A formula $\varphi \in \mathrm{P}_1$ has a finite model if and only if Procedure 1 returns True.*

Proof. We first assume that an input formula φ has a finite model. Therefore, we can obtain a flat finitely satisfiable formula φ' (by Lemma 2) and describe its model in terms of linear inequalities and congruences $\mathcal{E}_{\varphi'}$ (by Lemma 3). Clearly the system has a solution over $\mathbb{N}$ (e.g., a characteristic vector $\chi_{\varphi'}$), hence also suitable choices for $\mathcal{E}'_{\varphi'}$ and $\mathcal{E}''_{\varphi'}$ have solutions. Hence Procedure 1 returns *True*.

Conversely, suppose that Procedure 1 returns *True* for its input formula φ. We construct a model for φ. We do it simply by taking a proper number of realizations of each 1-type, exactly as described in the solution of the constructed system of linear inequalities $\mathcal{E}''_{\varphi'}$. □

Using the above lemma, one can conclude the following theorem:

Theorem 6. *The finite satisfiability problem for* P_1 *is* NP*-complete.*

Proof. The lower bound comes trivially from Boolean satisfiability problem or from the earlier works on C^1 [14]. For the upper bound it is enough to note that Procedure 1 works in NP. It follows from (i) the fact that flattening can be done in NP (Lemma 2), (ii) correspondence between systems of inequalities and characteristic vectors of P_1 models (Lemma 3), (iii) existence of sparse solutions of systems of inequalities (Lemma 1), and (iv) an NP algorithm for solving systems of inequalities with polynomially many unknowns [4]. □

4. Conclusions and future work

4.1. Conclusions

In this article we proposed a new logic called P_1 which significantly increase the expressive power of the one-variable fragment of first-order logic. The obtained logic generalizes previously known concepts of counting, i.e., threshold counting, modulo counting and cardinality comparison. By using a generic method of transforming a formula into a system of inequalities, we prove that every satisfiable P_1 formula can be represented as a system of inequalities of polynomial size. By using a well-known theorem that integer linear programming is in NP we obtained a tight NP upper bound for finite satisfiability for the logic P_1. This proves that the complexity of P_1 with expressive numerical constraints does not differ from the classical one-variable fragment of FO, or even from Boolean satisfiability, which is rather surprising.

4.2. Future work

For future work we would like to investigate other classical decidable fragments of first-order logic and see how their complexity and decidability status behaves after adding some form of Presburger constraints.

One candidate could be the two-variable fragment of first-order logic FO^2. However in the presence of cardinality comparison the logic becomes undecidable [8].

Another prominent logic is a two-variable fragment of the guarded fragment of first-order logic GF^2, which is known to be decidable even in the presence of counting quantifiers [13]. However, even adding modulo constraints to the logic is a challenging task and currently we do not even have a decidability proof. On the other hand, some decidable fragments of GF^2 extended with Presburger constraints are known. We already know that the complexity of the modal logic $\mathcal{K}$ or the description logic $\mathcal{ALC}$ do not differ from their Presburger versions, see [2,5,11]. We believe that to obtain tight complexity bounds for Presburger GF^2 one should start with a more modest goal, i.e., to establish the exact complexity of Presburger $\mathcal{ALCI}$, namely an extension of $\mathcal{ALC}$ with inverse relations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the Polish Ministry of Science and Higher Education program "Diamentowy Grant" no. DI2017 006447. The author would also like to thank the two anonymous reviewers as well as Witold Charatonik, Emanuel Kieroński and Antti Kuusisto for their careful proofreading and for pointing out numerous grammatical mistakes.

References

[1] Iskander Aliev, Jesús A. De Loera, Friedrich Eisenbrand, Timm Oertel, Robert Weismantel, The support of integer optimal solutions, SIAM J. Optim. 28 (3) (2018) 2152–2157.
[2] Franz Baader, A new description logic with set constraints and cardinality constraints on role successors, in: Clare Dixon, Marcelo Finger (Eds.), Proceedings of the Frontiers of Combining Systems – 11th International Symposium, FroCoS 2017, Brasília, Brazil, September 27–29, 2017, in: Lecture Notes in Computer Science, vol. 10483, Springer, 2017, pp. 43–59.
[3] Bartosz Bednarczyk, On one variable fragment of first order logic with modulo counting quantifier, in: Karoliina Lohiniva, Johannes Wahle (Eds.), ESSLLI 2017 Student Session, 29th European Summer School in Logic, Language & Information, Toulouse, France, July 17–28, 2017, pp. 7–13.
[4] I. Borosh, M. Flahive, B. Treybig, Small solutions of linear Diophantine equations, Discrete Math. 58 (3) (1986) 215–220.
[5] Stéphane Demri, Denis Lugiez, Complexity of modal logics with Presburger constraints, J. Appl. Log. 8 (3) (2010) 233–252.
[6] Marcelo Finger, Glauber De Bona, Algorithms for Deciding Counting Quantifiers over Unary Predicates, in: Satinder P. Singh, Shaul Markovitch (Eds.), Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, February 4–9, AAAI Press, 2017, pp. 3878–3884.
[7] Erich Grädel, Martin Otto, Eric Rosen, Two-variable logic with counting is decidable, in: Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29–July 2, IEEE Computer Society, 1997, pp. 306–317.
[8] Erich Grädel, Martin Otto, Eric Rosen, Undecidability results on two-variable logics, Arch. Math. Log. 38 (4–5) (1999) 313–354.
[9] Yevgeny Kazakov, Ian Pratt-Hartmann, A note on the complexity of the satisfiability problem for graded modal logics, in: Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, Los Angeles, CA, USA, August 11–14, IEEE Computer Society, 2009, pp. 407–416.
[10] Viktor Kuncak, Martin C. Rinard, Towards efficient satisfiability checking for boolean algebra with presburger arithmetic, in: Frank Pfenning (Ed.), Proceedings of the 21st International Conference on Automated Deduction, CADE-21, Bremen, Germany, July 17–20, in: Lecture Notes in Computer Science, vol. 4603, Springer, 2007, pp. 215–230.
[11] Clemens Kupke, Dirk Pattinson, Lutz Schröder, Reasoning with global assumptions in arithmetic modal logics, in: Adrian Kosowski, Igor Walukiewicz (Eds.), Fundamentals of Computation Theory – 20th International Symposium, Proceedings, FCT 2015, Gdańsk, Poland, August 17–19, 2015, in: Lecture Notes in Computer Science, vol. 9210, Springer, 2015, pp. 367–380.
[12] Leszek Pacholski, Wieslaw Szwast, Lidia Tendera, Complexity results for first-order two-variable logic with counting, SIAM J. Comput. 29 (4) (2000) 1083–1117.
[13] Ian Pratt-Hartmann, Complexity of the guarded two-variable fragment with counting quantifiers, J. Log. Comput. 17 (1) (2007) 133–155.
[14] Ian Pratt-Hartmann, On the computational complexity of the numerically definite syllogistic and related logics, Bull. Symb. Log. 14 (1) (2008) 1–28.