Complexity of Existential Positive First-Order Logic

Manuel Bodirsky, Miki Hermann, and Florian Richoux
{manuel.bodirsky, miki.hermann, florian.richoux}@lix.polytechnique.fr

LIX, École Polytechnique, 91128 Palaiseau, France

Abstract. Let Γ be a (not necessarily finite) structure with a finite relational signature. We prove that deciding whether a given existential positive sentence holds in Γ is in LOGSPACE or complete for the class $\mathrm{CSP}(\Gamma)_{\mathrm{NP}}$ under deterministic polynomial-time many-one reductions. Here, $\mathrm{CSP}(\Gamma)_{\mathrm{NP}}$ is the class of problems that can be reduced to the *constraint satisfaction problem* of Γ under *non-deterministic* polynomial-time many-one reductions.

Key words: Computational Complexity, Existential Positive First-Order Logic, Constraint Satisfaction Problems

1 Introduction

We study the computational complexity of the following class of computational problems. Let Γ be a structure with finite or infinite domain and with a finite relational signature. The model-checking problem for existential positive first-order logic, parametrized by Γ, is the following problem.

Problem: $\mathrm{ExPos}(\Gamma)$
Input: An existential positive first-order sentence Φ.
Question: Does Γ satisfy Φ?

An *existential positive first-order formula* over Γ is defined as follows:
- if R is a relation symbol of a relation from Γ with arity k and $x_1, \dots, x_k$ are variables, then $R(x_1, \dots, x_k)$ is a existential positive first-order formula;
- if φ and ψ are existential positive first-order formulas, then $\varphi \wedge \psi$ and $\varphi \vee \psi$ are existential positive first-order formulas;
- if φ is an existential positive first-order formula with a free variable x then $\exists x\, \varphi$ is an existential positive first-order formula.

An *existential positive first-order sentence* is an existential positive first-order formula without free variables. Note that we do not allow equality symbols in the existential positive sentences; this only makes our results stronger, since one might always add a relation symbol $=$ for the equality relation into the signature of Γ to obtain the result for the case where equality symbols are allowed.

The sentence does not need to be in prenex normal form; however, every existential positive first-order sentence can be transformed in an equivalent one in this form without an exponential blowup, thanks to the absence of universal quantifiers and negation symbols.

The *constraint satisfaction problem* $\mathrm{CSP}(\Gamma)$ for Γ is defined similarly, but its input consists of a *primitive positive* sentence, that is, a existential positive sentence without disjunctions. Constraint satisfaction problems frequently appear in many areas of computer science, and have attracted a lot of attention, in particular in combinatorics, artificial intelligence, finite model theory and universal algebra; we refer to the recent monograph with survey articles on this subject [7]. The class of constraint satisfaction problems for infinite structures Γ is a rich class of problems; it can be shown that for every computational problem there exists a relational structure Γ such that CSP(Γ) is equivalent to that problem under polynomial-time Turing reductions [1].

In this paper, we show that the complexity classification for existential positive first-order sentences over infinite structures can be reduced to the complexity classification for constraint satisfaction problems.

For finite structures Γ, our result implies that $\mathrm{ExPos}(\Gamma)$ is in LOGSPACE or NP-complete. The polynomial-time solvable cases of $\mathrm{ExPos}(\Gamma)$ are in this case precisely those relational structures Γ with an element a where all non-empty relations in Γ contain the tuple $(a, \ldots, a)$ composed only from the element a; in this case, $\mathrm{ExPos}(\Gamma)$ is called a-*valid*. Interestingly, this is no longer true for infinite structures Γ. Consider the structure $\Gamma := (\mathbb{N}, \neq)$, which is clearly not a-valid. However, $\mathrm{ExPos}(\Gamma)$ can be reduced to the Boolean formula evaluation problem (which is known to be in LOGSPACE) as follows: atomic formulas in Φ of the form $x \neq y$ are replaced by *true*, and atomic formulas of the form $x \neq x$ are replaced by *false*. The resulting Boolean formula is equivalent to true if and only if Φ is true in Γ.

A universal-algebraic study of the model-checking problem for finite structures Γ and various other syntactic restrictions of first-order logic (for instance positive first-order logic) can be found in [6].

2 Result

We write $L \leq_m L'$ if there exists a deterministic polynomial-time many-one reduction from L to L'.

Definition 1 (from [4]). *A problem A is* non-deterministic polynomial-time many-one reducible *to a problem B ($A \leq_{\mathrm{NP}} B$) if there is a nondeterministic polynomial-time Turing machine M such that $x \in A$ if and only if there exists a y computed by M on input x with $y \in B$. We denote by A_{NP} the smallest class that contains A and is closed under $\leq_{\mathrm{NP}}$.*

Observe that $\leq_{\mathrm{NP}}$ is transitive [4]. To state the complexity classification for existential positive first-order logic, we need the following concept. The *Γ-localizer $F(\psi)$* of a formula ψ is defined as follows:

- $F(\exists x\, \psi) = F(\psi)$
- $F(\varphi \wedge \psi) = F(\varphi) \wedge F(\psi)$
- $F(\varphi \vee \psi) = F(\varphi) \vee F(\psi)$
- $F(R(x_1, \ldots, x_k)) = \begin{cases} \mathit{true} & \text{if } R \text{ is a relation of } \Gamma \text{ and } R \neq \emptyset \\ \mathit{false} & \text{otherwise} \end{cases}$

Definition 2. *We call a structure Γ* locally refutable *if every existential positive sentence Φ is true in Γ* if and only if *the Γ-localizer $F(\Phi)$ is logically equivalent to* true.

Proposition 1. *A structure Γ is locally refutable if and only if every unsatisfiable conjunction of atomic formulas contains an unsatisfiable conjunct.*

Proof. First suppose that Γ is locally refutable, and let Φ be a conjunction of atomic formulas with variables $x_1, \ldots, x_n$. Then every conjunct of Φ is satisfiable in Γ if and only if $F(\exists x_1, \ldots, x_n.\Phi)$ is true. By local refutability of Γ this is the case if and only if $\exists x_1, \ldots, x_n.\Phi$ is true in Γ, which shows the claim.

Now suppose that Γ is not locally refutable, that is, there is an existential sentence Φ that is false in Γ such that $F(\Phi)$ is true. Now, we define recursively for each subformula Ψ of Φ where $F(\Psi)$ is true the formula $T(\Psi)$ as follows. If Ψ is of the form $\Psi_1 \vee \Psi_2$, then for some $i \in \{1, 2\}$ the formula $F(\Psi_i)$ must be true, and we set $T(\Psi)$ to be $T(\Psi_i)$. If Ψ is of the form $\Psi_1 \wedge \Psi_2$, then for both $i \in \{1, 2\}$ the formula $F(\Psi_i)$ must be true, and we set $T(\Psi)$ to be $T(\Psi_1) \wedge T(\Psi_2)$.

Each conjunct Φ in $T(\Phi)$ is satisfiable in Γ since $F(\Phi)$ is true. From the construction it is clear that if $T(\Phi)$ is satisfiable, then Φ is also satisfiable in Γ, a contradiction. Hence, $T(\Phi)$ is unsatisfiable in Γ but each conjunct is satisfiable. □

In Section 3, we will show the following result.

Theorem 1. *Let Γ be a structure with a finite relational signature τ. If Γ is locally refutable then the problem* $\textsc{ExPos}(\Gamma)$ *to decide whether an existential positive sentence is true in Γ is in* LOGSPACE. *If Γ is not locally refutable, then* $\textsc{ExPos}(\Gamma)$ *is complete for the class* $\mathrm{CSP}(\Gamma)_{\mathrm{NP}}$ *under polynomial-time many-one reductions.*

In particular, $\textsc{ExPos}(\Gamma)$ is in P or is NP-hard (under deterministic polynomial-time many-one reductions). If Γ is finite, then $\textsc{ExPos}(\Gamma)$ is in P or NP-complete, because finite domain constraint satisfaction problems are clearly in NP. The observation that $\textsc{ExPos}(\Gamma)$ is in P or NP-complete has previously been made in [3] and independently in [5]. However, our proof remains the same for finite domains and is simpler than the previous proofs.

3 Proof

Before we prove Theorem 1, we start with the following simpler result.

Theorem 2. *Let Γ be a structure with a finite relational signature τ. If Γ is locally refutable, then the problem* $\textsc{ExPos}(\Gamma)$ *to decide whether an existential positive sentence is true in Γ is in* LOGSPACE. *If Γ is not locally refutable, then* $\textsc{ExPos}(\Gamma)$ *is* NP*-hard (under polynomial-time many-one reductions).*

To show Theorem 2, we first prove the following lemma.

Lemma 1. *A structure Γ is not locally refutable if and only if there are existential positive formulas ψ_0 and ψ_1 with the property that*

– ψ_0 *and* ψ_1 *define non-empty relations over* Γ*;*
– $\psi_0 \wedge \psi_1$ *defines the empty relation over* Γ*.*

Proof. The "if"-part of the statement is immediate. To show the "only if"-part, suppose that Γ is not locally refutable. Then by Proposition 1 there is an unsatisfiable conjunction Ψ of satisfiable atomic formulas. Among all such formulas Ψ, let Ψ be one with minimal length. Let ψ_0 be one of the atomic formulas in Ψ, and let ψ_1 be the conjunction over the remaining conjuncts in Ψ. Since Ψ was chosen to be minimal, the formula ψ_1 must be satisfiable. By construction ψ_0 is also satisfiable and Ψ is unsatisfiable, which is what we had to show. □

Proof of Theorem 2: If Γ is locally refutable, then $\text{ExPos}(\Gamma)$ can be reduced to the positive Boolean formula evaluation problem, which is known to be LOGSPACE-complete. We only have to construct from an existential positive sentence Φ a Boolean formula $F := F_\Gamma(\Phi)$ as described before Definition 2. Clearly, this construction can be performed with logarithmic work-space. We evaluate F, and reject if F is false, and accept otherwise.

If Γ is not locally refutable, we show NP-hardness of $\text{ExPos}(\Gamma)$ by reduction from 3-SAT. Let I be a 3-SAT instance. We construct an instance Φ of $\text{ExPos}(\Gamma)$ as follows. Let ψ_0 and ψ_1 be the formulas from Lemma 1 (suppose they are d-ary). Let $v_1, \ldots, v_n$ be the Boolean variables in I. For each v_i we introduce d new variables $\bar{x}_i = x_i^1, \ldots, x_i^d$. Let Φ be the instance of $\text{ExPos}(\Gamma)$ that contains the following conjuncts:

– For each $1 \leq i \leq n$, the formula $\psi_0(\bar{x}_i) \vee \psi_1(\bar{x}_i)$
– For each clause $l_1 \vee l_2 \vee l_3$ in I, the formula $\psi_{i_1}(\bar{x}_{j_1}) \vee \psi_{i_2}(\bar{x}_{j_2}) \vee \psi_{i_3}(\bar{x}_{j_3})$ where $i_p = 0$ if l_p equals $\neg x_{j_p}$ and $i_p = 1$ if l_p equals x_{j_p}, for all $p \in \{1, 2, 3\}$.

It is clear that Φ can be computed in deterministic polynomial time from I, and that Φ is true in Γ if and only if I is satisfiable. □

Note that, applied to finite relational structures Γ, we obtain again the dichotomy from [3] and [5], that is, $\text{ExPos}(\Gamma)$ is in P if Γ is a-valid and NP-complete otherwise. We prove in the following proposition that, over a finite domain D, Γ is locally refutable if and only if it is a-valid for an element $a \in D$.

Proposition 2. *Let* Γ *be a relational structure with a finite domain. Then* Γ *is locally refutable if and only if it is* a*-valid.*

Proof. Suppose that Γ is a-valid, and let Φ be an existential positive sentence over the signature of Γ. To show that Γ is locally refutable, we only have to show that Φ is true in Γ when $F(\Phi)$ is equivalent to true (since the other direction holds trivially). But this follows from the fact that if an atomic formula $R(x_1, \ldots, x_n)$ is satisfiable in Γ then in fact this formula can be satisfied by setting all variables to a.

For the opposite direction of the statement, suppose that Γ is not a-valid for all $a \in D$. That it, when $D = \{a_1, \ldots, a_n\}$ we have that for each $a_i \in D$ there exists a non-empty relation R_i of arity r_i in Γ such that $(a_i, \ldots, a_i) \notin R$. Let r be $\sum_{i=1}^n r_i$, and let $x_1, \ldots, x_{rn}$ be distinct variables. Consider the formula

$$\Psi = \bigwedge_{\overline{y} \in \{x_1, \ldots, x_{rn}\}^r} R_1(y_1, \ldots, y_{r_1}) \wedge \cdots \wedge R_n(y_{r-r_n+1}, \ldots, y_r) . \quad (1)$$

By the pigeonhole principle, for every mapping $f : \{x_1, \ldots, x_{rn}\} \to D$ at least r variables are mapped to the same value, say to a_i. For a vector $\overline{y}$ that contains exactly these r variables, for some l there is a conjunct $R_i(y_{l+1}, \ldots, y_{l+r_i})$ in Ψ; but by assumption, R_i does not contain the tuple $(a_i, \ldots, a_i)$. This shows that $\exists x_1, \ldots, x_{rn}.\Psi$ is not true in Γ. On the other hand, since each relation R_i is non-empty, it is clear that the Boolean formula $F(\exists x_1, \ldots, x_{rn}.\Psi)$ is true. Therefore, Γ is not locally refutable. □

Proof of Theorem 1: If Γ is locally refutable then the statement has been shown in Theorem 2. Suppose that Γ is not locally refutable. To show that $\text{ExPos}(\Gamma)$ is contained in $\text{CSP}(\Gamma)_{\text{NP}}$, we construct a non-deterministic Turing machine T which takes as input an instance Φ of $\text{ExPos}(\Gamma)$, and which outputs an instance $T(\Phi)$ of $\text{CSP}(\Gamma)$ as follows.

On input Φ the machine T proceeds recursively as follows:

- if Φ is of the form $\exists x.\varphi$ then return $\exists x.T(\varphi)$;
- if Φ is of the form $\varphi_1 \wedge \varphi_2$ then return $T(\varphi_1) \wedge T(\varphi_2)$;
- if Φ is of the form $\varphi_1 \vee \varphi_2$ then non-deterministically return either $T(\varphi_1)$ or $T(\varphi_2)$;
- if Φ is of the form $R(x_1, \ldots, x_k)$ then return $R(x_1, \ldots, x_k)$.

The output of T can be viewed as an instance of $\text{CSP}(\Gamma)$, since it can be transformed to a primitive positive sentence (by moving all existential quantifiers to the front). It is clear that T has polynomial running time, and that Φ is true in Γ if and only if there exists a computation of T on Φ that computes a sentence that is true in Γ.

We now show that $\text{ExPos}(\Gamma)$ is hard for $\text{CSP}(\Gamma)_{\text{NP}}$ under $\leq_m$-reductions. Let L be a problem with a non-deterministic polynomial-time many-one reduction to $\text{CSP}(\Gamma)$, and let M be the non-deterministic Turing machine that computes the reduction. We have to construct a deterministic Turing machine M' that computes for any input string s in polynomial time in $|s|$ an instance Φ of $\text{ExPos}(\Gamma)$ such that Φ is true in Γ if and only if there exists a computation of M on s that computes a satisfiable instance of $\text{CSP}(\Gamma)$.

Say that the running time of M on s is in $O(|s|^e)$ for a constant e. Hence, there are constants s_0 and c such that for $|s| > s_0$ the running time of M and hence also the number of constraints in the input instance of $\text{CSP}(\Gamma)$ produced by the reduction is bounded by $t := c|s|^e$. The non-deterministic computation of M can be viewed as a deterministic computation with access to non-deterministic advice bits as shown in [2]. We also know that for $|s| > s_0$, the machine M can access at most t non-deterministic bits. If w is a sufficiently long bit-string, we write M_w for the deterministic Turing machine obtained from M by using the bits in w as the non-deterministic bits, and $M_w(s)$ for the instance of $\text{CSP}(\Gamma)$ computed by M_w on input s.

If $|s| \leq s_0$, then M' returns $\exists \bar{x}.\psi_1(\bar{x})$ if there is an $w \in \{0,1\}^*$ such that $M_w(s)$ is a satisfiable instance of $\text{CSP}(\Gamma)$, and M' returns $\exists \bar{x}.\psi_0(\bar{x}) \wedge \psi_1(\bar{x})$ otherwise (i.e., it returns a false instance of $\text{ExPos}(\Gamma)$; ψ_0 and ψ_1 are defined in Lemma 1). Since s_0 is a fixed finite value, M' can perform these computations in constant time.

It is convenient to assume that Γ has just a single relation R (we can always find a CSP which is deterministic polynomial-time equivalent and where the template is of this form[1]). Let l be the arity of R. Then instances of $\text{CSP}(\Gamma)$ with variables $x_1, \ldots, x_n$

[1] If $\Gamma = (D; R_1, \ldots, R_n)$ where R_i has arity r_i and is not empty, then $\text{CSP}(\Gamma)$ is equivalent to $\text{CSP}(D; R_1 \times \cdots \times R_n)$ where $R_1 \times \cdots \times R_n$ is the $\sum_{i=1}^{n} r_i$-ary relation defined

can be encoded as sequences of numbers that are represented by binary strings of length $\lceil \log t \rceil$ as follows: The i-th number m in this sequence indicates that the $(((i-1) \bmod l) + 1)$-st variable in the $(((i-1)\ div\ l) + 1)$-st constraint is x_m.

For $|s| > s_0$, we use a construction from the proof of Cook's theorem given in [2]. In this proof, a computation of a non-deterministic Turing machine T accepting a language L is encoded by Boolean variables that represent the state and the position of the read-write head of T at time r, and the content of the tape at position j at time r. The tape content at time 0 consists of the input x, written at positions 1 through n, and the non-deterministic advice bit string w, written at positions -1 through $-|w|$. The proof in [2] specifies a deterministic polynomial-time computable transformation f_L that computes for a given string s a SAT instance $f_L(s)$ such that there is an accepting computation of T on s if and only if there is a satisfying truth assignment for $f_L(s)$.

In our case, the machine M computes a reduction and thus computes an output string. Recall our binary representation of instances of the CSP: M writes on the output tape a sequence of numbers represented by binary strings of length $\lceil \log t \rceil$. It is straightforward to modify the transformation f_L given in the proof of Theorem 2.1 in [2] to obtain for all positive integers a, b, c where $a \leq t$, $b \leq l$, $c \leq \lceil \log t \rceil$, and $d \in \{0, 1\}$, a deterministic polynomial-time transformation $g^d_{a,b,c}$ that computes for a given string s a SAT instance $g^d_{a,b,c}(s)$ with distinguished variables $z_1, \ldots, z_p$, $p \leq t$ for the non-deterministic bits in the computation of M such that the following are equivalent:

- $g^d_{a,b,c}(s)$ has a satisfying assignment where z_i is set to $w_i \in \{0, 1\}$ for $1 \leq i \leq p$;
- the c-th bit in the b-th variable of the a-th constraint in $M_w(s)$ equals d.

We use the transformations $g^d_{a,b,c}$ to define M' as follows. The machine M' first computes the formulas $g^d_{a,b,c}(s)$. For every Boolean variable v in these formulas we introduce a new conjunct $\varphi_0(\overline{x}_v) \vee \varphi_1(\overline{x}_v)$ where $\overline{x}_v$ is a d-tuple of fresh variables. Then, every positive literal v in the original conjuncts of the formula is replaced by $\varphi_1(\overline{x}_v)$, and every negative literal $l = \neg v$ by $\varphi_0(\overline{x}_v)$. We then existentially quantify over all variables except for $\bar{x}_{z_1}, \ldots, \bar{x}_{z_p}$. Let $\psi^d_{a,b,c}(s)$ denote the resulting existential positive formula. For positive integers k and i, we denote $k[i]$ the i-th bit in the binary representation of k. Let n be the total number of variables in the CSP instance $M_w(s)$ (in particular, $n \leq t$). It is clear that the formula

$$\exists y_1, \ldots, y_n, \bar{x}_{z_1}, \ldots, \bar{x}_{z_p}. \bigwedge_{1 \leq a, k_1, \ldots, k_l \leq t} \left(\left(\bigwedge_{b \leq l, c} \psi^{k_b[c]}_{a,b,c}(s) \right) \rightarrow R(y_{k_1}, \ldots, y_{k_l}) \right)$$

can be re-written in existential positive form Φ without blow-up: we can replace implications $\alpha \rightarrow \beta$ by $\neg\alpha \vee \beta$, and then move the negation to the atomic level, where we can remove negation by exchanging the role of φ_0 and φ_1. Hence, Φ can be computed by M' in polynomial time.

---

as the Cartesian product of the relations $R_1, \ldots, R_n$. Similarly, ExPos(Γ) is equivalent to ExPos($D; R_1 \times \cdots \times R_n$).

We claim that the formula Φ is true in Γ if and only if there exists a computation of M on s that computes a satisfiable instance of CSP(Γ). To see this, let w be a sufficiently long bitstring such that $M_w(s)$ is a satisfiable instance of CSP(Γ). Suppose for the sake of notation that the n variables in $M_w(s)$ are the variables $y_1, \ldots, y_n$. Let $a_1, \ldots, a_n$ be a satisfying assignment to those n variables. Then, if for $1 \leq i \leq n$ the variable y_i in the formula Φ is set to a_i, and for $1 \leq i \leq p$ the variables $\bar{x}_{z_i}$ are set to a tuple that satisfies ψ_d where d is the i-th bit in w, we claim that the inner part of Φ is true in Γ. The reason is that, due to the way how we set the variables of the form $\bar{x}_{z_i}$, the precondition $\left(\bigwedge_{b \leq l,c} \psi^{k_b[c]}_{a,b,c}(s)\right)$ is true if and only if $R(y_{k_1}, \ldots, y_{k_l})$ is a constraint in $M_w(s)$. Therefore, all the atomic formulas of the form $R(y_{k_1}, \ldots, x_{k_l})$ are satisfied due to the way how we set the variables y_i, and hence Φ is true in Γ. It is straightforward to verify that the opposite implication holds as well, and this shows the claimed equivalence. □

4 Structures with function symbols

In this section, we briefly discuss the complexity of ExPos(Γ) when Γ might also contain functions. That is, we assume that the signature of Γ consists of a finite set of relation and function symbols, and that the input formulas for the problem ExPos(Γ) are existential positive first-order formulas over this signature. It is easy to see from the proofs in the previous section that when Γ is not locally refutable, then ExPos(Γ) is still NP-hard (with the same definition of local refutability as before).

The case when Γ is locally refutable becomes more intricate when Γ has functions. We present an example of a locally refutable structure Γ where ExPos(Γ) is NP-hard. Let the signature of Γ be the structure $(2^{\mathbb{N}}; \neq, \cap, \cup, c, \mathbf{0}, \mathbf{1})$ where $\neq$ is the binary disequality relation, $\cap$ and $\cup$ are binary functions for intersection and union, respectively, c is a unary function for complementation, and $\mathbf{0}, \mathbf{1}$ are constants (i.e., 0-ary functions) for the empty set and the full set $\mathbb{N}$, respectively.

Proposition 3. *The structure* $(2^{\mathbb{N}}; \neq, \cap, \cup, c, \mathbf{0}, \mathbf{1})$ *is locally refutable.*

Proof. By Lemma 1 is suffices to show that if Ψ is a conjunction of atomic formulas that are satisfiable in Γ, then Ψ is satisfiable over Γ. Since the only relation symbol in the structure is $\neq$, every conjunct in Ψ is of the form $t_1 \neq t_2$, where t_1 and t_2 are terms formed by variables and the function symbols $\cap, \cup, c$, $\mathbf{1}$ and $\mathbf{0}$. By Boole's fundamental theorem of Boolean algebras, $t = t'$ can be re-written as $t'' = \mathbf{0}$. Therefore, Ψ can be written as $t_1 \neq \mathbf{0} \wedge \cdots \wedge t_n \neq \mathbf{0}$. Since Γ is an infinite Boolean algebra, it now follows from known results in [8] (Theorem 5.1) that if $t_i \neq \mathbf{0}$ is satisfiable in Γ for all $i \leq n$, then Ψ is satisfiable in Γ as well. □

Proposition 4. *The problem* ExPos$(2^{\mathbb{N}}; \neq, \cap, \cup, c, \mathbf{0}, \mathbf{1})$ *is* NP*-hard.*

Proof. The proof is by reduction from SAT. Given a Boolean formula Ψ in CNF with variables $x_1, \ldots, x_n$, we replace each conjunction in Ψ by $\cap$, each disjunction by $\cup$, and each negation by c. Let t be the resulting term over the signature $\{\cap, \cup, c\}$ and variables $x_1, \ldots, x_n$. It is easy to verify that $\exists x_1, \ldots, x_n.t \neq \mathbf{0}$ is true in Γ if and only if Ψ is a satisfiable Boolean formula. □

Acknowledgment

We would like to thank Víctor Dalmau for helpful suggestions, and Moritz Müller for the encouragement to study the case where the structure Γ contains function symbols.

References

1. M. Bodirsky and M. Grohe. Non-Dichotomies in Constraint Satisfaction Complexity. *Proceedings 35th International Colloquium on Automata, Languages and Programming (ICALP 2008), Part II, Reykjavik (Iceland)*, 5126, 184–196, 2008.
2. M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. *W.H. Freeman and Co*, 1979.
3. M. Hermann and F. Richoux. On the Computational Complexity of Monotone Constraint Satisfaction Problems. *Proceedings 3rd Annual Workshop on Algorithms and Computation (WALCOM 2009), Kolkata (India)*, 286–297, 2009.
4. R. E. Ladner, N. A. Lynch and A. L. Selman. A Comparison of Polynomial-Time Reducibilities. *Theoretical Computer Science*, 1(2), 103–124, 1975.
5. B. Martin. Dichotomies and Duality in First-order Model Checking Problems. *CoRR abs/cs/0609022*, 2006.
6. B. Martin. First-Order Model Checking Problems Parameterized by the Model. *Proceedings 4th Conference on Computability in Europe (CiE 2008), Athens (Greece)*, 417–427, 2008.
7. N. Creignou, Ph. G. Kolaitis, and H. Vollmer, editors. *Complexity of Constraints — An Overview of Current Research Themes*, volume 5250 of Lecture Notes in Computer Science, *Springer Verlag*, 2008.
8. Kim Marriott and Martin Odersky. Negative boolean constraints. *Theor. Comput. Sci.*, 160(1&2):365–380, 1996.