A logic for PTIME and a parameterized halting problem

Yijia Chen
Department of Computer Science
Shanghai Jiaotong University
yijia.chen@cs.sjtu.edu.cn

Jörg Flum
Mathematisches Institut
Albert-Ludwigs Universität Freiburg
joerg.flum@math.uni-freiburg.de

Abstract

*In [7] Nash, Remmel, and Vianu have raised the question whether a logic $L_\leq$, already introduced by Gurevich in 1988, captures polynomial time, and they give a reformulation of this question in terms of a parameterized halting problem p-*ACC$_\leq$ *for nondeterministic Turing machines. We analyze the precise relationship between $L_\leq$ and p-*ACC$_\leq$*. We show that p-*ACC$_\leq$ *is not fixed-parameter tractable if* "P $\neq$ NP *holds for all time constructible and increasing functions." Moreover, a slightly stronger complexity theoretic hypothesis implies that $L_\leq$ does not capture polynomial time. Furthermore, we analyze the complexity of various variants of p-*ACC$_\leq$ *and address its construction problem.*

1. Introduction

The existence of a logic expressing precisely the polynomial time properties of structures remains the central problem in descriptive complexity (recent articles addressing this question are [4, 7]). A proof that such a logic does not exist would yield that P $\neq$ NP. By a result due to Immerman [6] and Vardi [8] least fixed-point logic LFP captures polynomial time on *ordered* structures. However the property of an *arbitrary* structure of having a universe of even cardinality is not expressible in LFP. There are artificial logics capturing polynomial time on arbitrary structures, but they do not fulfill a natural requirement to logics in this context:

> There is an algorithm that decides whether $\mathcal{A}$ is a model of φ for all structures $\mathcal{A}$ and sentences φ of the logic and that does this for fixed φ in time polynomial in the size $\|\mathcal{A}\|$ of $\mathcal{A}$. (1)

In [5] the author introduces a logic $L_\leq$ related to LFP (cf. Section 3 for its definition), in which precisely the polynomial time properties are expressible; one conjectures that $L_\leq$ does not satisfy the effectivity condition (1). In [7] it has been shown that the statement "$L_\leq$ satisfies (1)" can be equivalently formulated as a statement concerning the complexity of a halting problem for nondeterministic Turing machines (NTM). This reformulation is best expressed in the terminology of parameterized complexity. We consider the *parameterized acceptance problem* p-ACC$_\leq$ *for NTMs*:

p-ACC$_\leq$	
Instance:	An NTM $\mathbb{M}$ and $n \in \mathbb{N}$ in unary.
Parameter:	$\|\mathbb{M}\|$, the size of $\mathbb{M}$.
Question:	Does $\mathbb{M}$ accept the empty input tape in at most n steps?

Then (see Theorem 10 (3), (4) for the precise statement)

$$L_\leq \text{ satisfies (1)} \quad \text{if and only if} \quad p\text{-ACC}_\leq \in \text{XP}. \qquad (2)$$

In this paper we mainly deal with two questions:

(a) What does "p-ACC$_\leq$ is fixed-parameter tractable" mean for the logic $L_\leq$?

(b) What is the complexity of p-ACC$_\leq$?

While we can answer question (a), we are only able to relate the statements "p-ACC$_\leq$ $\in$ XP" and "p-ACC$_\leq$ $\in$ FPT" with other open problems of complexity theory.

More precisely, the content of the different sections is the following. It is known that the time bound for the model-checking problem for LFP, that is, for the evaluation of a sentence φ of LFP in a structure $\mathcal{A}$, contains a factor $\|\mathcal{A}\|^{O(|\varphi|)}$; an analysis of the corresponding algorithm shows that (at least for LFP-sentences in normal form) a factor of the form $\|\mathcal{A}\|^{O(\text{width}(\varphi))}$ suffices, where width(φ), the width of φ, essentially is the maximum number of free variables in a subformula of φ. The main result of Section 4 shows that the existence of a bound of this type for the model-checking problem of the logic $L_\leq$ is equivalent to p-ACC$_\leq$ $\in$ FPT.

Let P[TC] $\neq$ NP[TC] mean that for all <u>t</u>*ime* <u>c</u>*onstructible* and increasing functions h the class of problems decidable in *deterministic* <u>p</u>*olynomial time* in h and the class of problems decidable in <u>n</u>*ondeterministic* <u>p</u>*olynomial time* in

h are distinct, that is, DTIME($h^{O(1)}$) $\neq$ NTIME($h^{O(1)}$). In Section 6 we show that P[TC] $\neq$ NP[TC] implies that p-ACC$_{\leq}$ $\notin$ FPT. Furthermore a stronger hypothesis where DTIME($h^{O(1)}$) $\neq$ NTIME($h^{O(1)}$) is replaced by NTIME($h^{O(1)}$) $\not\subseteq$ DTIME($h^{O(\log h)}$) implies that p-ACC$_{\leq}$ $\notin$ XP (and thus by (2), it implies that $L_{\leq}$ does not capture polynomial time). In [2] we related these hypotheses to other statements of complexity theory; in particular, we saw that P[TC] $\neq$ NP[TC] holds if there is a P-*bi-immune* problem in NP.

We also study some variants of p-ACC$_{\leq}$. First we deal with p-ACC$_{=}$, the problem obtained from p-ACC$_{\leq}$ by asking for an accepting run of *exactly* n steps. We show that p-ACC$_{=}$ is related to a logic $L_{=}$ as p-ACC$_{\leq}$ is to the logic $L_{\leq}$. In Section 5 we improve a result of [1] by showing that p-ACC$_{=}$ $\in$ FPT if and only if E $=$ NE (that is, DTIME($2^{O(n)}$) $=$ NTIME($2^{O(n)}$)). Furthermore, in Section 7 we introduce a halting problem for deterministic Turing machines, the "deterministic version" of p-ACC$_{\leq}$, and show that it is an example of a problem *nonuniformly* fixed-parameter tractable but not contained in *uniform* XP, to the best of our knowledge, the first natural such example.

Finally, in Section 8, we consider the construction problem associated with p-ACC$_{\leq}$ and show that it is not fpt Turing reducible to p-ACC$_{\leq}$ in case p-ACC$_{\leq}$ $\notin$ XP.

2. Preliminaries

In this section we review some of the basic concepts of parameterized complexity and of logics and their complexity. We refer to [3] for notions not defined here.

2.1. Parameterized complexity. We identify problems with subsets Q of $\{0,1\}^*$. Clearly, as done mostly, we present concrete problems in a verbal, hence uncodified form. All Turing machines have $\{0,1\}$ as their alphabet.

We view *parameterized problems* as pairs (Q,κ) consisting of a problem $Q \subseteq \{0,1\}^*$ and a *parameterization* $\kappa : \{0,1\}^* \to \mathbb{N}$, which is required to be polynomial time computable. We will present parameterized problems in the form as we did for p-ACC$_{\leq}$ in the Introduction.

Recall that a parameterized problem (Q,κ) is *fixed-parameter tractable* (or, in the class FPT) if $x \in Q$ is solvable by an *fpt-algorithm*, that is, by an algorithm running in time $f(\kappa(x)) \cdot |x|^{O(1)}$ for some computable $f : \mathbb{N} \to \mathbb{N}$. The parameterized problem (Q,κ) is in the class XP if $x \in Q$ is solvable in time $O(|x|^{f(\kappa(x))})$ for some computable $f : \mathbb{N} \to \mathbb{N}$.

Besides these classes of the usual (strongly uniform) parameterized complexity theory we need their *uniform* versions FPT$_{\text{uni}}$ and XP$_{\text{uni}}$ and their *nonuniform* versions FPT$_{\text{nu}}$ and XP$_{\text{nu}}$. For example, $(Q,\kappa) \in$ FPT$_{\text{uni}}$ if there is an algorithm solving $x \in Q$ in time $f(\kappa(x)) \cdot |x|^{O(1)}$ for some *arbitrary* $f : \mathbb{N} \to \mathbb{N}$; and $(Q,\kappa) \in$ FPT$_{\text{nu}}$ if there is a constant $c \in \mathbb{N}$, an arbitrary function $f : \mathbb{N} \to \mathbb{N}$, and for every $k \in \mathbb{N}$ an algorithm solving the problem $x \in Q$ for all x with $\kappa(x) = k$ in time $f(k) \cdot |x|^c$.

We write $(Q,\kappa) \leq^{\text{fpt}} (Q',\kappa')$ if there is an fpt reduction from (Q,κ) to (Q',κ') (this concept refers to the strongly uniform parameterized complexity theory).

2.2. Logic. A *vocabulary* τ is a finite set of relation symbols. Each relation symbol has an *arity*. A *structure* $\mathcal{A}$ of vocabulary τ, or τ-*structure* (or, simply structure), consists of a nonempty set A called the *universe*, and an interpretation $R^{\mathcal{A}} \subseteq A^r$ of each r-ary relation symbol $R \in \tau$. We say that $\mathcal{A}$ is finite, if A is a finite set. *All structures in this paper are assumed to be finite.*

For a structure $\mathcal{A}$ we denote by $\|\mathcal{A}\|$ the size of $\mathcal{A}$, that is, the length of a reasonable encoding of $\mathcal{A}$ as string in $\{0,1\}^*$ (e.g., cf. [3] for details). If necessary, we can assume that the universe of a finite structure is $[m] := \{1,\dots,m\}$ for some natural number $m \geq 1$, as all the properties of structures we consider are invariant under isomorphisms; in particular, it suffices that from the encoding of $\mathcal{A}$ we can recover $\mathcal{A}$ up to isomorphism. The reader will easily convince himself that we can assume that there is a computable function lgth such that for every vocabulary τ and $m \geq 1$ (we just collect the properties of lgth we use in Section 4):

(a) $\|\mathcal{A}\| = \text{lgth}(\tau,m)$ for every τ-structure $\mathcal{A}$ with universe of cardinality m (that is, for fixed τ and m, the encoding of each τ-structure with universe of m elements has length equal to lgth(τ,m));

(b) lgth$(\tau,m) \geq \max\{2,m\}$;

(c) for fixed τ, the function $m \mapsto \text{lgth}(\tau,m)$ is time constructible and lgth(τ,m) is polynomial in m;

(d) lgth$(\tau,m) <$ lgth(τ',m') for all τ,τ' with $\tau \subseteq \tau'$ and m,m' with $m < m'$;

(e) lgth$(\tau,m) = O(\log|\tau| \cdot |\tau| \cdot m)$ for every τ containing only unary relation symbols;

(f) lgth$(\tau \cup \{R\},m) = O(\text{lgth}(\tau,m) + m^2)$ for every binary relation symbol R not in τ.

In effectivity considerations for a structure $\mathcal{A}$ we denote by $<_A$ the ordering on A given by the encoding of $\mathcal{A}$.

We assume familiarity with *first-order logic* FO and its extension *least fixed-point logic* LFP. We denote by FO$[\tau]$ and LFP$[\tau]$ the set of sentences of vocabulary τ of FO and of LFP, respectively. It is known that LFP captures P on the class of ordered structures.

As we will introduce further semantics for the formulas of least fixed-point logic, we write $\mathcal{A} \models_{\text{LFP}} \varphi$ if the structure $\mathcal{A}$ is a model of the LFP-sentence φ. An algorithm based on the inductive definition of the satisfaction relation for LFP shows (see [9]):

Proposition 1. *The model-checking problem $\mathcal{A} \models_{\text{LFP}} \varphi$ for structures $\mathcal{A}$ and* LFP*-sentences φ can be solved in time*

$$|\varphi| \cdot \|\mathcal{A}\|^{O(|\varphi|)}.$$

It is known that every LFP-sentence is equivalent to an LFP-sentence in normal form, where an LFP-sentence φ is in *normal form* if it has the form

$$\varphi = \exists y_1 \dots \exists y_\ell [\text{LFP}_{x_1 \dots x_\ell, X} \psi(x_1 \dots x_\ell, X)]\, \bar{y}, \qquad (3)$$

with a first-order formula ψ, an ℓ-ary relation variable X and $\bar{y} = y_1 \dots y_\ell$; moreover, $x_1, \dots, x_\ell$ are the first-order variables free in ψ. If in addition every subformula of φ has at most ℓ free first-order variables, then the problem $\mathcal{A} \models_{\text{LFP}} \varphi$ can be solved in time $O(|\varphi| \cdot \|\mathcal{A}\|^{2\ell} \cdot \ell)$. To state the corresponding result for arbitrary LFP-sentences we introduce the width and the depth of LFP-formulas.

Let $\varphi(x_1, \dots, x_r, Y_1, \dots, Y_s)$ be an LFP-formula and let the pairwise distinct $x_1, \dots, x_r$ be the first-order variables free in φ and the pairwise distinct $Y_1, \dots, Y_s$ be the second-order variables free in φ. The *variable-weight of* φ is

$$r + \sum_{i \in [s]} \text{ar}(Y_i),$$

where $\text{ar}(Y_i)$ is the arity of Y_i. The *width* of φ, denoted by $\text{width}(\varphi)$, is the maximum of the variable-weights of the subformulas of φ. By $\text{depth}(\varphi)$, the *depth* of φ, we denote the maximum nesting depth of LFP-operators in φ.

Proposition 2. *The model-checking problem $\mathcal{A} \models_{\text{LFP}} \varphi$ for structures $\mathcal{A}$ and* LFP*-sentences φ can be solved in time*

$$|\varphi| \cdot \|\mathcal{A}\|^{O\left((1+\text{depth}(\varphi)) \cdot \text{width}(\varphi)\right)}.$$

2.3. Logics capturing P. A *logic* L consists

- for every vocabulary τ of a decidable set $L[\tau]$ of strings, the set of *L-sentences of vocabulary* τ;
- of a *satisfaction relation* $\models_L$; if $(\mathcal{A}, \varphi) \in \models_L$, then, for some τ, we have that $\mathcal{A}$ is a τ-structure and $\varphi \in L[\tau]$; furthermore for each $\varphi \in L[\tau]$ the class of structures $\mathcal{A}$ with $\mathcal{A} \models_L \varphi$ is closed under isomorphisms.

We say that $\mathcal{A}$ is a *model* of φ if $\mathcal{A} \models_L \varphi$ (that is, if $(\mathcal{A}, \varphi) \in \models_L$). We set $\text{Mod}_L(\varphi) := \{\mathcal{A} \mid \mathcal{A} \models_L \varphi\}$ and say that φ *axiomatizes* the class $\text{Mod}_L(\varphi)$.

We partly take over the following terminology from [7].

Definition 3. Let L be a logic.

(a) *L is a logic for* P if for all vocabularies τ and all classes C (of encodings) of τ-structures closed under isomorphisms we have

$$C \in \text{P} \iff C = \text{Mod}_L(\varphi) \text{ for some } \varphi \in L[\tau].$$

(b) *L is a* P*-bounded logic for* P if (a) holds and if there is an algorithm $\mathbb{A}$ deciding $\models_L$, that is, for every structure $\mathcal{A}$ and L-sentence φ the algorithm $\mathbb{A}$ decides whether $\mathcal{A} \models_L \varphi$. Moreover, for every fixed φ the algorithm $\mathbb{A}$ runs in time polynomial in $\|\mathcal{A}\|$.

Hence, if L is a P-bounded logic for P, then for every L-sentence φ the algorithm $\mathbb{A}$ witnesses that $\text{Mod}_L(\varphi) \in \text{P}$. However, we do not necessarily know ahead of time the bounding polynomial.

(c) *L is an effectively* P*-bounded logic for* P if L is a P-bounded logic for P and if in addition to the algorithm $\mathbb{A}$ as in (b) there is a computable function that assigns to every L-sentence φ a polynomial $q \in \mathbb{N}[X]$ such that $\mathbb{A}$ decides whether $\mathcal{A} \models_L \varphi$ in $\leq q(\|\mathcal{A}\|)$ steps.

3. Order-invariant variants of LFP

For a vocabulary τ let $\tau_< := \tau \cup \{<\}$, where $<$ is a binary relation symbol not in τ.

For every class of τ-structures C in P closed under isomorphisms the class of $\tau_<$-structures

$$C_< := \left\{(\mathcal{A}, <^A) \;\middle|\; \mathcal{A} \in C \text{ and } <^A \text{ an ordering of } A\right\} \quad (4)$$

is in P, too; hence, as the logic LFP captures polynomial time on the class of ordered structures, there is a LFP$[\tau_<]$-sentence axiomatizing $C_<$. However, we are interested in a sentence axiomatizing the class C.

In order to obtain a logic that captures polynomial time on all structures one has considered variants of LFP obtained by restricting to order-invariant sentences or by modifying the semantics such that all sentences are order-invariant. In this section we recall the corresponding logics and start by introducing the respective notions of invariance.

Definition 4. (a) A pair $(\varphi, \mathcal{A})$ is in the relation INV if

- for some vocabulary τ we have that $\mathcal{A}$ is a τ-structure and $\varphi \in \text{LFP}[\tau_<]$;
- (φ *is invariant in $\mathcal{A}$ under a change of the ordering*) for all orderings $<_1$ and $<_2$ on A we have

$$(\mathcal{A}, <_1) \models_{\text{LFP}} \varphi \iff (\mathcal{A}, <_2) \models_{\text{LFP}} \varphi.$$

(b) An LFP$[\tau_<]$-sentence φ is *order-invariant* if $(\varphi, \mathcal{A}) \in$ INV for all τ-structures $\mathcal{A}$.

(c) For an LFP$[\tau_<]$-sentence φ and $m \in \mathbb{N}$ we write

$$(\varphi, m) \in \text{INV}$$

if $(\varphi, \mathcal{A}) \in$ INV for all τ-structures $\mathcal{A}$ with $|A| = m$.

(d) For an LFP$[\tau_<]$-sentence φ and $m \in \mathbb{N}$ we write

$$(\varphi, \leq m) \in \text{INV}$$

if $(\varphi, \mathcal{A}) \in$ INV for all τ-structures $\mathcal{A}$ with $|A| \leq m$.

Note that every LFP$[\tau_<]$-sentence axiomatizing a class of the form $C_<$ (see (4)) is order-invariant.

The different degrees of invariance lead to the following different logics. For all logics L we let

$$L[\tau] := \text{LFP}[\tau_<].$$

Hence, these logics only differ in their semantics. The logic L_{inv} is the first naive attempt to get an (effectively) P-bounded logic for P. Its semantics is fixed by

$$\mathcal{A} \models_{L_{\text{inv}}} \varphi \iff \big(\varphi \text{ is order-invariant and } (\mathcal{A}, <_A) \models_{\text{LFP}} \varphi\big)$$

(recall that $<_A$ denotes the ordering on A given by the encoding of $\mathcal{A}$).

Clearly (and this remark will also apply to the logics $L_{\text{str}}, L_=$, and $L_\leq$ to be defined yet), for all classes $C \in$ P of τ-structures closed under isomorphisms every LFP$[\tau_<]$-sentence axiomatizing the class $C_<$ is an $L_{\text{inv}}[\tau]$-sentence axiomatizing C. Thus all properties in P are expressible in L_{inv}.

As $\text{Mod}_{L_{\text{inv}}}(\varphi) = \text{Mod}_{\text{LFP}}(\varphi)$ if $\varphi \in \text{LFP}[\tau_<]$ is invariant and $\text{Mod}_{L_{\text{inv}}}(\varphi) = \emptyset$ otherwise, L_{inv} is a logic for P. However, as already remarked in [5], a simple application of a theorem of Trachtenbrot shows that the set of invariant LFP$[\tau_<]$-sentences is not decidable and thus $\models_{L_{\text{inv}}}$ is not decidable; hence L_{inv} is not a P-bounded logic for P.

For the logic L_{str} we require invariance in the corresponding structure:

$$\mathcal{A} \models_{L_{\text{str}}} \varphi \iff \Big((\varphi, \mathcal{A}) \in \text{INV and } (\mathcal{A}, <_A) \models_{\text{LFP}} \varphi\Big).$$

For a binary relation symbol E, consider an FO$\big[\{E\}_<\big]$-sentence φ expressing that E is not a graph or that in the ordering $<$ there are two consecutive elements which are not related by an edge. The class $\text{Mod}_{L_{\text{str}}}(\varphi)$ is the complement of the class of graphs having a Hamiltonian path and hence it is coNP-complete (a different coNP-complete class is axiomatized in [5, Theorem 1.16]).

As an easy consequence we get:

Proposition 5. *L_{str} is a logic for* P *iff* P $=$ NP *iff L_{str} is an effectively* P*-bounded logic for* P.

Proof: Assume that P $=$ NP. To show that L_{str} is an effectively P-bounded logic for P we consider the problem

Instance:	An L_{str}-sentence φ, a structure $\mathcal{A}$ and the number $\|\mathcal{A}\|^{|\varphi|}$ in unary.
Problem:	Is $(\mathcal{A}, \varphi) \in \text{INV}$?

By Proposition 1, it is in coNP and hence it is solvable in polynomial time. This yields the algorithm $\mathbb{A}$ as required by part (b) and (c) of Definition 3. □

As coNP-complete problems can be axiomatized in L_{str}, we define the logic $L_=$ using a stronger invariance property:

$$\mathcal{A} \models_{L_=} \varphi \iff \Big((\varphi, |A|) \in \text{INV and } (\mathcal{A}, <_A) \models_{\text{LFP}} \varphi\Big);$$

in particular, $\mathcal{A} \models_{L_=} \varphi$ can only hold if φ is invariant in all structures with universe of the same cardinality as $\mathcal{A}$.

Note that for an $L_=$-sentence φ it is not clear whether the class of models of φ is in P. In fact, in Section 7 we show:

Proposition 6. *$L_=$ is a logic for* P *iff* E $=$ NE *iff $L_=$ is an effectively* P*-bounded logic for* P.

Finally we introduce the logic $L_\leq$, where invariance in all structures of the same or smaller cardinality is required:

$$\mathcal{A} \models_{L_\leq} \varphi \iff \Big((\varphi, \leq |A|) \in \text{INV and } (\mathcal{A}, <_A) \models_{\text{LFP}} \varphi\Big).$$

If an LFP$[\tau_<]$-sentence φ is not order-invariant, then the class $\text{Mod}_{L_\leq}(\varphi)$ only contains (up to isomorphism) finitely many structures and hence it is in P. Therefore $L_\leq$ (as L_{inv}) is a logic for P.

In particular, L_{inv} and $L_\leq$ have less expressive power than L_{str} (if P $\neq$ NP) and less than $L_=$ (if E $\neq$ NE). Clearly if P $=$ NP (and hence E $=$ NE), then all, L_{inv}, L_{str}, $L_=$ and $L_\leq$, have the same expressive power. Otherwise we have:

Proposition 7. *(1) If* P $\neq$ NP*, then there is a class axiomatizable in L_{str} but not in $L_=$.*

(2) If E $\neq$ NE*, then there is a class axiomatizable in $L_=$ but not in L_{str}.*

Proof: To get (1) we observe that the complement of the class of graphs having a Hamiltonian path, a class axiomatizable in L_{str} as we have seen, is not axiomatizable in $L_=$ if P $\neq$ NP; this is shown by the following claim.

Claim 1: Let C be a class of τ-structures. Assume that $C \notin$ P. Furthermore assume that for every $m \in \mathbb{N}$ with $m \geq 2$ there is a structure $\mathcal{A}_m \in C$ such that $|A_m| = m$. Then C is not axiomatizable in $L_=$.

Proof of Claim 1: Assume that $C = \text{Mod}_{L_=}(\varphi)$. For $m \geq 2$ we have $(\varphi, m) \in \text{INV}$, as $\mathcal{A}_m \models_{L_=} \varphi$. Clearly, $(\varphi, 1) \in \text{INV}$. Hence φ is order-invariant and thus $\text{Mod}_{\text{LFP}}(\varphi) = C_<$. So $C_<$ and hence C are in P, a contradiction. ⊣

A proof of part (2), based on the following claim, will be presented in Section 4.1.

Claim 2: Let X be a set of natural numbers in unary with $X \notin$ P. Assume that the class C of τ-structures has the property: For all τ-structures $\mathcal{A}$

$$\mathcal{A} \in C \iff |A| \in X.$$

Then C is not axiomatizable in L_{str}.

Proof of Claim 2: Assume that $C = \text{Mod}_{L_{\text{str}}}(\varphi)$ with $\varphi \in L_{\text{str}}[\tau]$. Clearly $(\psi, \mathcal{A}) \in \text{INV}$ for all LFP$[\tau_<]$-sentences ψ and every τ-structure $\mathcal{A}$ of the form $\mathcal{A} = (A, (\emptyset)_{P\in\tau})$ (all relations $P \in \tau$ are interpreted by the empty set). Then for every $m \geq 1$ and for the natural ordering $<$ on $[m]$:

$$\big([m], (\emptyset)_{P\in\tau}, <\big) \models_{\text{LFP}} \varphi \iff \big([m], (\emptyset)_{P\in\tau}\big) \models_{L_{\text{str}}} \varphi \iff m \in X$$

As $\big([m], (\emptyset)_{P\in\tau}, <\big) \models_{\text{LFP}} \varphi$ can be checked in time polynomial in $\big\|\big([m], (\emptyset)_{P\in\tau}, <\big)\big\|$ and hence, polynomial in m, we see that $X \in \text{P}$, a contradiction. $\square$

This paper mainly addresses the question whether $L_\leq$ is an effectively P-bounded logic for P, a question raised in [7]. It is conjectured that it is not a P-bounded logic for P. In [7] this question (or conjecture) is reformulated as an effectivity property for a halting problem for nondeterministic Turing machines. We analyze the relationship between the logic and the halting problem in the next section.

4. A halting problem and its relationship to $L_\leq$

We defined the parameterized acceptance problem p-ACC$_\leq$ for NTMs in the Introduction. We shall see in this section how the complexity of this problem is related to properties of the logic $L_\leq$. We start with the following simple observation on the complexity of p-ACC$_\leq$.

Proposition 8. *The problem p-ACC$_\leq$ is in the class* FPT$_{\text{nu}}$.

Proof: Fix $k \in \mathbb{N}$; then there are only finitely many NTMs $\mathbb{M}$ with $\|\mathbb{M}\| = k$, say, $\mathbb{M}_1, \dots, \mathbb{M}_s$. For each $i \in [s]$ let ℓ_i be the smallest natural number ℓ such that there exists an accepting run of $\mathbb{M}_i$, started with empty input tape, of length ℓ. We set $\ell_i = \infty$ if $\mathbb{M}_i$ does not accept the empty input tape. Consider the algorithm $\mathbb{A}_k$ that on any instance $(\mathbb{M}, n)$ of p-ACC$_\leq$ with $\|\mathbb{M}\| = k$ determines the i with $\mathbb{M} = \mathbb{M}_i$, and then accepts if and only if $\ell_i \leq n$. It has running time $O(\|\mathbb{M}\| + n)$; thus it witnesses that p-ACC$_\leq \in$ FPT$_{\text{nu}}$. $\square$

This observation can easily be generalized. We call a parameterized problem (Q, κ) *slicewise monotone* if its instances have the form (x, n), where $x \in \{0,1\}^*$ and $n \in \mathbb{N}$ is given in unary, if $\kappa(x, n) = |x|$, and finally if for all $x \in \{0,1\}^*$ and $n, n' \in \mathbb{N}$ we have

$$(x, n) \in Q \text{ and } n < n' \text{ imply } (x, n') \in Q.$$

In particular, p-ACC$_\leq$ is slicewise monotone and the preceding argument shows:

Lemma 9. $(Q, \kappa) \in$ FPT$_{\text{nu}}$ *for slicewise monotone* (Q, κ).

Is p-ACC$_\leq \in$ FPT$_{\text{uni}}$ or, at least, p-ACC$_\leq \in$ XP$_{\text{uni}}$? By [7] the conjecture "$L_\leq$ is not a P-bounded logic for P" mentioned in the previous section is equivalent to the statement p-ACC$_\leq \notin$ XP$_{\text{uni}}$ (and similarly, the statement "$L_\leq$ is not an effectively P-bounded logic for P" is equivalent to p-ACC$_\leq \notin$ XP).

However it is not even clear whether p-ACC$_\leq \notin$ FPT. Do the statements p-ACC$_\leq \in$ FPT and p-ACC$_\leq \in$ FPT$_{\text{uni}}$ also correspond to natural properties of the logic $L_\leq$? We address this problem in this section.

Proposition 2 motivates the introduction of the following notion. We say that $L_\leq$ is an *(effectively) depth-width* P-*bounded logic for* P if there is an algorithm $\mathbb{A}$ deciding $\models_{L_\leq}$ in such a way that there is a (computable) function h such that $\mathcal{A} \models_{L_\leq} \varphi$ can be solved in time

$$h(|\varphi|) \cdot \|\mathcal{A}\|^{O\left((1+\text{depth}(\varphi))\cdot\text{width}(\varphi)\right)}.$$

By Proposition 2, the logic LFP "is an effectively depth-width P-bounded logic for P on ordered structures." Parts (1) and (2) of the following theorem are the main result of this section, (3) and (4) are already mentioned in [7].

Theorem 10. *(1) $L_\leq$ is an effectively depth-width* P-*bounded logic for* P *if and only if p-ACC$_\leq \in$* FPT.

(2) $L_\leq$ is a depth-width P-*bounded logic for* P *if and only if p-ACC$_\leq \in$* FPT$_{\text{uni}}$.

(3) $L_\leq$ is an effectively P-*bounded logic for* P *if and only if p-ACC$_\leq \in$* XP.

(4) $L_\leq$ is a P-*bounded logic for* P *if and only if p-ACC$_\leq \in$* XP$_{\text{uni}}$.

4.1. Proof of Theorem 10 and some consequences.

The following observations will lead to a proof of the direction "from right to left" in the statements of Theorem 10.

For an $L_\leq$-sentence φ let τ_φ be the set of relation symbols distinct from $<$ that do occur in φ. For a suitable time constructible function $t : \mathbb{N} \to \mathbb{N}$ we will need an NTM $\mathbb{M}_\varphi(t)$ that, started with empty tape, operates as follows: In a first phase it writes a word of the form 1^m for some $m \geq 1$ on some tape. The second phase (the main phase) consists of at most $t(m) + 1$ steps (this can be ensured as t is time constructible). If $\mathbb{M}_\varphi(t)$ does not stop during the first $t(m)$ steps of the main phase, then it stops in the next step and rejects. During this $t(m)$ steps, $\mathbb{M}_\varphi(t)$ guesses (the encoding of) a τ_φ-structure $\mathcal{A}$ with universe $[m]$ and two orderings $<_1$ and $<_2$ on $[m]$ and checks whether $\big((\mathcal{A}, <_1) \models_{\text{LFP}} \varphi \iff (\mathcal{A}, <_2) \models_{\text{LFP}} \varphi\big)$. If this is not the case, then $\mathbb{M}_\varphi(t)$ accepts; otherwise it rejects.

The first phase takes m steps. To guess a τ_φ-structure $\mathcal{A}$ with universe $[m]$ and two orderings $<_1$ and $<_2$ requires

$O(\text{lgth}(\tau_\varphi, m) + 2m^2)$ bits (see Section 2.2); thus for some $d_1 \in \mathbb{N}$ the machine $\mathbb{M}_\varphi(t)$ needs

$$t^1_\varphi(m) := d_1 \cdot (\text{lgth}(\tau_\varphi, m) + 2m^2)$$

steps. Finally, by Proposition 2, to check the equivalence $\big((\mathcal{A}, <_1) \models_{\text{LFP}} \varphi$ iff $(\mathcal{A}, <_2) \models_{\text{LFP}} \varphi\big)$ takes at most

$$t^2_\varphi(m) := |\varphi| \cdot \text{lgth}((\tau_\varphi)_<, m)^{d_2 \cdot (1+\text{depth}(\varphi)) \cdot \text{width}(\varphi)}$$

steps for some $d_2 \in \mathbb{N}$. By the time constructibility of the function $m \mapsto \text{lgth}((\tau_\varphi)_<, m)$ we can arrange the machine in such a way that it needs exactly the number of steps given by the upper bounds $t^1_\varphi(m)$ and $t^2_\varphi(m)$ (if it is not stopped by the time bound $t(m)$). Thus, if in the first phase of a run $\mathbb{M}_\varphi(t)$ has written the word 1^m, then $\mathbb{M}_\varphi(t)$ performs exactly

$$t_\varphi(m) := t^1_\varphi(m) + t^2_\varphi(m) \tag{5}$$

additionally steps before it stops (assuming $t_\varphi(m) \leq t(m)$). Note that t_φ is increasing. Therefore we have

$$\begin{aligned}(\varphi, \leq m) \in \text{INV} \iff& \\ (\mathbb{M}_\varphi(t_\varphi), m + t_\varphi(m)) &\notin p\text{-ACC}_\leq.\end{aligned} \tag{6}$$

We collect some facts we are going to use:

(i) There is an algorithm assigning to every $L_\leq$-sentence φ the machine $\mathbb{M}_\varphi(t_\varphi)$.

(ii) For every $L_\leq$-sentence φ and all τ_φ-structures $\mathcal{A}$:

$$\mathcal{A} \models_{L_\leq} \varphi \iff \Big((\mathbb{M}_\varphi(t_\varphi), m + t_\varphi(|A|)) \notin p\text{-ACC}_\leq \text{ and } (\mathcal{A}, <_A) \models_{\text{LFP}} \varphi\Big).$$

(iii) There is a computable function g such that for every $L_\leq$-sentence φ and all τ_φ-structures $\mathcal{A}$ we have

$$t_\varphi(|A|) \leq g(|\varphi|) \cdot \|\mathcal{A}\|^{O\big((1+\text{depth}(\varphi)) \cdot \text{width}(\varphi)\big)}$$

(by the definition (5) of the function t_φ and the properties of the lgth-function mentioned in Section 2.2).

Now we can show the direction "from right to left" in the statements of Theorem 10. We give the proof for the claims (1) and (2); obvious modifications yield (3) and (4).

Assume $p\text{-ACC}_\leq \in \text{FPT}_{\text{uni}}$ ($p\text{-ACC}_\leq \in \text{FPT}$), that is, assume that $(\mathbb{M}, n) \in p\text{-ACC}_\leq$ can be solved in time

$$f(\|\mathbb{M}\|) \cdot n^e$$

for some $e \in \mathbb{N}$ and some (computable) function $f : \mathbb{N} \to \mathbb{N}$. We consider the problem $\mathcal{A} \models_{L_\leq} \varphi$ where $\mathcal{A}$ is a structure and φ an $L_\leq$-sentence. We may assume that $\mathcal{A}$ is a τ_φ-structure (if $\mathcal{A}$ contains more relations, we omit them; this can be done in $O(|\varphi| + \|\mathcal{A}\|)$ steps). Using (i), (iii), and Proposition 2 we see that there is an algorithm and a (computable) function h such that the condition

$$(\mathbb{M}_\varphi(t_\varphi), |A| + t_\varphi(|A|)) \notin p\text{-ACC}_\leq \text{ and } (\mathcal{A}, <_A) \models_{\text{LFP}} \varphi$$

and hence, by (ii), the problem $\mathcal{A} \models_{L_\leq} \varphi$ can be solved in time $h(|\varphi|) \cdot \|\mathcal{A}\|^{O\big((1+\text{depth}(\varphi)) \cdot \text{width}(\varphi)\big)}$. □

Before we proceed with the proof of Theorem 10, it is worthwhile to extract from the previous argument information relevant for the logic $L_=$. The corresponding halting problem $p\text{-ACC}_=$ is obtained from $p\text{-ACC}_\leq$ by replacing its question by:

Does $\mathbb{M}$ accept the empty input tape in *exactly* n steps?

Lemma 11. *If* $p\text{-ACC}_= \in \text{FPT}$*, then* $L_=$ *is an effectively depth-width* P*-bounded logic for* P.[1]

Proof: Note that the following variant of (6) holds:

$$(\varphi, m) \in \text{INV} \iff (\mathbb{M}_\varphi(t_\varphi), m + t_\varphi(m)) \notin p\text{-ACC}_=,$$

and thus, $\mathcal{A} \models_{L_=} \varphi$ is equivalent to

$$(\mathbb{M}_\varphi(t_\varphi), |A| + t_\varphi(|A|)) \notin p\text{-ACC}_= \text{ and } (\mathcal{A}, <_A) \models_{\text{LFP}} \varphi.$$

Thus our claim can be derived in exactly the same way as the corresponding statement for $L_\leq$. □

We turn to a proof of the directions from "left to right" in Theorem 10. Let $\mathbb{M}$ be an NTM and let $m_0 := m_0(\mathbb{M})$ be the maximum of the number of states and the number of tapes. We can assume that $[k]$ is the set of states of $\mathbb{M}$ (for some $k \leq m_0$) and that 1 is its initial state. Furthermore, we may assume that every two distinct successor configurations of a given configuration of $\mathbb{M}$ have distinct states. We let $P_0, P_1, \ldots, P_k$ be unary relation symbols. We shall see that for $\tau := \{P_0, \ldots, P_k\}$ there is a $\varphi_\mathbb{M} \in \text{LFP}[\tau_<]$ in normal form with the following properties: For every τ-structure $\mathcal{A}$

(a) If $|A| < m_0$, then $(\mathcal{A}, <^A) \models_{\text{LFP}} \varphi_\mathbb{M}$ for all orderings $<^A$ on A.

(b) If $|A| \geq m_0$ and the subsets $P^\mathcal{A}_0, \ldots, P^\mathcal{A}_k$ do not form a partition of A, then $(\mathcal{A}, <^A) \models_{\text{LFP}} \varphi_\mathbb{M}$ for all orderings $<^A$ on A.

(c) Let $|A| \geq m_0$ and assume that $P^\mathcal{A}_0, \ldots, P^\mathcal{A}_k$ form a partition of A and $<^A$ is an ordering on A. Let $a_1, \ldots, a_{|A|}$ be the enumeration of the elements of A according to the ordering $<^A$ and choose i_s such that $a_s \in P^\mathcal{A}_{i_s}$ for $s \in [|A|]$.

[1] Along the lines of the proof the reader will easily verify the analogues for $p\text{-ACC}_=$ and $L_=$ of the directions "from right to left" of all statements of Theorem 10. However, all the others will follow from Corollary 14.

(i) If there is a $j \in [|A|-1]$ such that $1, i_1, \ldots, i_j$ is the sequence of states of a complete run of $\mathbb{M}$, started with empty input tape (in particular, $i_s \neq 0$ for all $s \in [j]$), then $(\mathcal{A}, <^A) \models_{\text{LFP}} \varphi_{\mathbb{M}}$ if and only if this run of $\mathbb{M}$ is a rejecting one.

(ii) If for all $j \in [|A|-1]$ the sequence $1, i_1, \ldots, i_j$ does not correspond to a complete run of $\mathbb{M}$ with empty input tape, then $(\mathcal{A}, <^A) \models_{\text{LFP}} \varphi_{\mathbb{M}}$.

We show that for every $m \geq m_0(\mathbb{M})$

$$(\mathbb{M}, m) \in p\text{-}\mathrm{ACC}_{\leq} \iff (\varphi_{\mathbb{M}}, \leq m) \notin \mathrm{INV}. \quad (7)$$

First assume that $(\mathbb{M}, m) \in p\text{-}\mathrm{ACC}_{\leq}$. Then there are $j \in [m-1]$ and $i_1, \ldots, i_j \in [k]$ such that $1, i_1, \ldots, i_j$ is the sequence of states of an accepting run of $\mathbb{M}$. By (c)(i) there is a structure $\mathcal{A}$ on $[m]$ such that $(\mathcal{A}, <^A) \not\models_{\text{LFP}} \varphi_{\mathbb{M}}$ for the natural ordering $<^A$ on $[m]$ and $P_0^{\mathcal{A}} = \{m\}$. We choose an ordering $<'$ on $[m]$ such that m is the first element of $<'$ and hence, $i_1 = 0$ under $<'$. By (c)(ii) we see that $(\mathcal{A}, <')$ $\models_{\text{LFP}} \varphi_{\mathbb{M}}$. Hence, $(\varphi_{\mathbb{M}}, \leq m) \notin \mathrm{INV}$.

Conversely, if $(\mathbb{M}, m) \notin p\text{-}\mathrm{ACC}_{\leq}$, it is easy to see, using (a)–(c), that $\mathcal{A} \models_{L_{\leq}} \varphi_{\mathbb{M}}$ for every structure $\mathcal{A}$ with $|A| \leq m$; hence, $(\varphi_{\mathbb{M}}, \leq m) \in \mathrm{INV}$.

The sentence $\varphi_{\mathbb{M}}$ (in normal form and hence of depth 1) is obtained by standard techniques. (We will sketch its construction in the full version of the paper.) The sentence $\varphi_{\mathbb{M}}$ depends on the machine $\mathbb{M}$, however, and this is important, as in $\varphi_{\mathbb{M}}$ we have to take care of a run of at most as many steps as the cardinality of the universe, it can be defined in such a way that its width is independent of $\mathbb{M}$; thereby we use the fact that we can address the ith element in the ordering $<$ by a formula of width 3.

Now we are able to finish the proof of the directions "from left to right" in Theorem 10. Again we present the argument for claims (1) and (2) of this theorem. Assume that $L_{\leq}$ is an (effectively) depth-width P-bounded logic for P and choose $c \in \mathbb{N}$ and a (computable) function $h : \mathbb{N} \to \mathbb{N}$ such that the model-checking problem $\mathcal{A} \models_{L_{\leq}} \varphi$ for structures $\mathcal{A}$ and $L_{\leq}$-sentences φ can be solved in time

$$h(|\varphi|) \cdot \|\mathcal{A}\|^{c \cdot (1+\text{depth}(\varphi)) \cdot \text{width}(\varphi)}. \quad (8)$$

We show that $p\text{-}\mathrm{ACC}_{\leq} \in \mathrm{FPT}_{\text{uni}}$ ($p\text{-}\mathrm{ACC}_{\leq} \in \mathrm{FPT}$). Let $(\mathbb{M}, m)$ be an arbitrary instance of $p\text{-}\mathrm{ACC}_{\leq}$. If $m < m_0(\mathbb{M})$ ($\leq \|\mathbb{M}\|$), we check whether $(\mathbb{M}, m) \in p\text{-}\mathrm{ACC}_{\leq}$ by brute force. Otherwise, we construct from $\mathbb{M}$ the sentence $\varphi_{\mathbb{M}}$. We choose the τ-structure $\mathcal{A}_m$ with $A_m = [m]$ and empty relations. By property (d) of the lgth-function (see Section 2.2), we have $\|\mathcal{A}_m\| = O(\log|\tau| \cdot |\tau| \cdot |A_m|) = O(|\varphi_{\mathbb{M}}|^2 \cdot m)$. As

$$(\varphi_{\mathbb{M}}, \leq m) \in \mathrm{INV} \text{ iff } (\mathcal{A}_m \models_{L_{\leq}} \varphi_{\mathbb{M}} \text{ or } \mathcal{A}_m \models_{L_{\leq}} \neg\varphi_{\mathbb{M}}),$$

we obtain by (7)

$$(\mathbb{M}, m) \notin p\text{-}\mathrm{ACC}_{\leq} \text{ iff } (\mathcal{A}_m \models_{L_{\leq}} \varphi_{\mathbb{M}} \text{ or } \mathcal{A}_m \models_{L_{\leq}} \neg\varphi_{\mathbb{M}}).$$

Therefore, by (8), there is a (computable) function f and a constant $e \in \mathbb{N}$ (recall that for all NTMs $\mathbb{M}$ the depth of $\varphi_{\mathbb{M}}$ is one and that there is a constant bounding the width of $\varphi_{\mathbb{M}}$) such that $(\mathbb{M}, m) \in p\text{-}\mathrm{ACC}_{\leq}$ can be solved in time $f(\|\mathbb{M}\|) \cdot m^e$. This finishes the proof of Theorem 10. □

Again we extract from the proof the information on $p\text{-}\mathrm{ACC}_{=}$ and $L_{=}$ that we shall need in Section 5.

Lemma 12. *If $L_{=}$ is a logic for* P*, then $p\text{-}\mathrm{ACC}_{=} \in \mathrm{XP}_{\text{nu}}$.*

Proof: A minor change in the definition of $\varphi_{\mathbb{M}}$ in the previous proof yields an LFP-sentence $\chi_{\mathbb{M}}$ with

$$(\mathbb{M}, m) \in p\text{-}\mathrm{ACC}_{=} \iff (\chi_{\mathbb{M}}, m) \notin \mathrm{INV}$$

instead of (7), and hence

$$(\mathbb{M}, m) \notin p\text{-}\mathrm{ACC}_{=} \text{ iff } (\mathcal{A}_m \models_{L_{=}} \chi_{\mathbb{M}} \text{ or } \mathcal{A}_m \models_{L_{=}} \neg\chi_{\mathbb{M}}).$$

Assume $L_{=}$ is a logic for P. Fix $k \in \mathbb{N}$ and let $\mathbb{M}_1, \ldots, \mathbb{M}_s$ be the finitely many NTMs $\mathbb{M}$ with $\|\mathbb{M}\| = k$. As $L_{=}$ is a logic for P, for all $i \in [s]$ there is an algorithm solving $\mathcal{A} \models_{L_{=}} \chi_{\mathbb{M}_i}$ in time polynomial in $\|\mathcal{A}\|$. Then the last equivalence yields the claim $p\text{-}\mathrm{ACC}_{=} \in \mathrm{XP}_{\text{nu}}$. □

We close this section by a proof of Proposition 7 (2).

Proof of Proposition 7 (2): Let X be a set of natural numbers in binary in NE \ E. Then $X(\text{un}) \in$ NP \ P, where $X(\text{un})$ is the set of natural numbers of X in unary. Hence there is an NTM $\mathbb{M}$ that given $m \in \mathbb{N}$ in *unary* decides whether $m \in X(\text{un})$ in polynomial time, say, in time $c \cdot m^d$. We may assume that every run of $\mathbb{M}$ on input m has length $c \cdot m^d$. Similar to the $\varphi_{\mathbb{M}}$ in the proof of Theorem 7, we construct an LFP-sentence $\rho_{\mathbb{M}}$ expressing that

> if for some $m \in \mathbb{N}$ the universe has cardinality $c \cdot m^d$ and the relations $P_0, \ldots, P_k$ code a run of $\mathbb{M}$ with input 1^m, then it is not accepting.

Then for every $\{P_0, \ldots, P_s\}$-structure $\mathcal{A}$ we have

$$\mathcal{A} \models_{L_{=}} \rho_{\mathbb{M}} \iff |A| \notin \{c \cdot m^d \mid m \in X(\text{un})\}. \quad (9)$$

As the set $\{c \cdot m^d \mid m \in X(\text{un})\}$ of natural numbers in unary is not in P, we get that $\mathrm{Mod}_{L_{=}}(\rho_{\mathbb{M}})$ is not axiomatizable in L_{str} by Claim 2 in the proof of Proposition 7. □

5. The parameterized complexity of p-ACC$_{=}$.

Let $\mathbb{M}$ be an NTM. By suitably adding to $\mathbb{M}$ a state, which can be accessed and left nondeterministically, one obtains a machine $\mathbb{M}^*$ such that for all $n \in \mathbb{N}$ the machine $\mathbb{M}^*$ accepts the empty input tape in exactly n steps if and only if $\mathbb{M}$ accepts the empty input tape in $\leq n$ steps. Hence $p\text{-}\mathrm{ACC}_{\leq} \leq^{\text{fpt}} p\text{-}\mathrm{ACC}_{=}$. Recall that $p\text{-}\mathrm{ACC}_{\leq} \in \mathrm{FPT}_{\text{nu}}$. On the other hand, $p\text{-}\mathrm{ACC}_{=} \notin \mathrm{FPT}_{\text{nu}}$ if E $\neq$ NE, as shown by the main result of this section:

Theorem 13. *The following statements are equivalent:*

$$p\text{-}\mathrm{ACC}_= \notin \mathrm{FPT}, \quad p\text{-}\mathrm{ACC}_= \notin \mathrm{XP}_{\mathrm{nu}}, \quad \mathrm{E} \neq \mathrm{NE}.$$

In [1] it is shown that p-ACC$_=$ $\in$ XP implies E $=$ NE. By Lemmas 11 and 12 we get as a consequence of Theorem 13 the following improvement of Proposition 6.

Corollary 14. *$L_=$ is a logic for* P *iff* E $=$ NE *iff $L_=$ is an effectively depth-width* P*-bounded logic for* P.

We prove Theorem 13 by the following two lemmas.

Lemma 15. *If* E $=$ NE*, then* p-ACC$_=$ $\in$ FPT.

Proof: Consider the classical problem:

Instance:	An NTM $\mathbb{M}$ and $n \in \mathbb{N}$ in *binary*.
Problem:	Does $\mathbb{M}$ accept the empty input tape in exactly n many steps?

Clearly, it is NE. By the assumption E $=$ NE, we can solve it in time $2^{O(\|\mathbb{M}\|+\log n)}$. It follows that p-ACC$_=$ is decidable in time $O(n) + 2^{O(\|\mathbb{M}\|+\log n)} = 2^{O(\|\mathbb{M}\|)} \cdot n^{O(1)}$, and hence p-ACC$_=$ $\in$ FPT. □

Lemma 16. *If* p-ACC$_=$ $\in$ XP$_{\mathrm{nu}}$*, then* E $=$ NE.

Proof: Assume that p-ACC$_=$ $\in$ XP$_{\mathrm{nu}}$. Let $Q \subseteq \{0,1\}^*$ be in NE. We have to show $Q \in$ E. Without loss of generality we may assume that every $x \in Q$ starts with a "1." Let $n(x)$ be the natural number with binary representation x; then

$$n(x) \neq n(y) \text{ for } x, y \in Q \text{ with } x \neq y. \tag{10}$$

As $Q \in$ NE there is an NTM $\mathbb{M}$ and a $c \in \mathbb{N}$ such that $\mathbb{M}$ decides whether $x \in Q$ in time $2^{c\cdot|x|}$ and every run of $\mathbb{M}$ on input x has length at most $2^{c\cdot|x|}$. Note that for x starting with a "1", we have $2^{c\cdot|x|} = n(x)^c$.

We define an NTM $\mathbb{M}^*$ that started with empty input tape runs as follows:

1. Guess a string $y \in \{0,1\}^*$
2. **if** y does not start with a "1", **then** reject
3. simulate $\mathbb{M}$ on input y for $n(y)^c$ many steps
4. **if** $\mathbb{M}$ rejects, **then** reject
5. make some additional dummy steps such that so far the total running time of $\mathbb{M}^*$ is $2 \cdot n(y)^c - 1$
6. Accept.

By (10) we have for every $x \in \{0,1\}^*$ starting with a "1" that $x \in Q$ if and only if $\mathbb{M}^*$ accepts the empty input tape in exactly $2 \cdot n(x)^c$ many steps. As p-ACC$_=$ $\in$ XP$_{\mathrm{nu}}$, for some $d \in \mathbb{N}$ we can decide whether $\mathbb{M}^*$ accepts the empty string in exactly $2 \cdot n(x)^c$ many steps in time

$$(2 \cdot n(x)^c)^d.$$

Hence, $x \in Q$ can be decided in time $2^{O(|x|)}$. □

Proof of Theorem 13: Immediate by Lemmas 15 and 16. □

6. The parameterized complexity of p-ACC$_\leq$.

We already know that the parameterized problem p-ACC$_\leq$ is in FPT$_{\mathrm{nu}}$; however, is it fixed-parameter tractable or at least in XP? We address these questions in this section.

Let

$$\mathrm{P}[\mathrm{TC}] \neq \mathrm{NP}[\mathrm{TC}]$$

mean that DTIME$(h^{O(1)}) \neq$ NTIME$(h^{O(1)})$ for all time constructible and increasing functions h.

The assumption P[TC] $\neq$ NP[TC] implies P $\neq$ NP, even E $\neq$ NE, as seen by taking as h the identity function and the function 2^n, respectively. At the end of this section we are going to relate P[TC] $\neq$ NP[TC] to further statements of complexity theory. The main result of this section is:

Theorem 17. *If* P[TC] $\neq$ NP[TC]*, then* p-ACC$_\leq$ $\notin$ FPT.

The following idea underlies the proof (given in the full version of the paper) of this result. Assume that p-ACC$_\leq$ $\in$ FPT. Then, in particular we have a *deterministic* algorithm deciding p-ACC$_\leq$, the (parameterized) acceptance problem for *nondeterministic* Turing machines. This yields a way (different from brute force) to translate nondeterministic algorithms into deterministic ones; a careful analysis of this translation shows that NTIME$(h^{O(1)}) \subseteq$ DTIME$(h^{O(1)})$ for a suitable time constructible and increasing function h.

Refining the argument we get p-ACC$_\leq$ $\notin$ XP; however we need a complexity-theoretic assumption (apparently) stronger than P[TC] $\neq$ NP[TC] (again for a proof we refer to the full version of the paper).

Theorem 18. *Assume* NTIME$(h^{O(1)}) \not\subseteq$ DTIME$(h^{O(\log h)})$ *for every time constructible and increasing function h. Then* p-ACC$_\leq$ $\notin$ XP.

6.1. Relating P[TC] $\neq$ NP[TC] to other statements. We partly report on results from [2] relating P[TC] $\neq$ NP[TC] and the hypothesis in Theorem 18 to further statements of complexity theory.

Let C be a classical complexity class. Recall that a problem $Q \subseteq \{0,1\}^*$ is C-*bi-immune* if both Q and the complement of Q do not have an *infinite subset* that belongs to C. It has been conjectured that NP contains a P-bi-immune problem.

Proposition 19. *The following statement (a) implies (b).*

(a) NP *contains a* P*-bi-immune problem.*

(b) P[TC] $\neq$ NP[TC].

It seems that the statement (a) is much stronger than (b). In fact as shown in [2] "not (b)" implies

> there is an infinite $I \in$ P such that for all $Q \in$ NP at least one of the sets $Q \cap I$ and $\big(\{0,1\}^* \setminus Q\big) \cap I$ is an infinite set in P,

while "not (b)" can be reformulated as

> for all $Q \in$ NP there is an infinite $I \in$ P such that at least one of the sets $Q \cap I$ or $\big(\{0,1\}^* \setminus Q\big) \cap I$ is an infinite set in P.

Furthermore it is shown in [2]:

Proposition 20. *If* NP *contains an* E*-bi-immune problem, then* NTIME$(h^{O(1)}) \not\subseteq$ DTIME$(h^{O(\log h)})$ *for every time constructible and increasing function h.*

7. A deterministic variant of p-ACC$_\leq$

If in the problem p-ACC$_\leq$ we replace the NTM $\mathbb{M}$ by a deterministic Turing machine (DTM) simulating all computation paths of length n of $\mathbb{M}$ with empty input tape we "arrive at" p-DTM-EXP-ACC$_\leq$:

> p-DTM-EXP-ACC$_\leq$
> *Instance:* A DTM $\mathbb{M}$ and $n \in \mathbb{N}$ in unary.
> *Parameter:* $\|\mathbb{M}\|$.
> *Question:* Does $\mathbb{M}$ accept the empty input tape in at most 2^n steps?

Thus, p-ACC$_\leq$ $\leq^{\text{fpt}}$ p-DTM-EXP-ACC$_\leq$. As the latter problem is slicewise monotone, we know that it is in FPT$_{\text{nu}}$ by Lemma 9. Clearly, FPT$_{\text{nu}}$ $\subseteq$ XP$_{\text{nu}}$ and XP $\subseteq$ XP$_{\text{uni}}$ $\subseteq$ XP$_{\text{nu}}$. The problem p-DTM-EXP-ACC$_\leq$ lies in FPT$_{\text{nu}} \setminus$ XP$_{\text{uni}}$, as we show:

Theorem 21. p-DTM-EXP-ACC$_\leq \notin$ XP$_{\text{uni}}$.

Proof: One easily verifies that p-DTM-EXP-ACC$_\leq$ is fpt equivalent to

> p-DTM-INP-EXP-ACC$_\leq$
> *Instance:* A DTM $\mathbb{M}$, $x \in \{0,1\}^*$, and $n \in \mathbb{N}$ in unary.
> *Parameter:* $\|\mathbb{M}\| + |x|$.
> *Question:* Does $\mathbb{M}$ accept x in $\leq 2^n$ steps?

Thus, it suffices to show that p-DTM-INP-EXP-ACC$_\leq \notin$ XP$_{\text{uni}}$. By contradiction, assume there exists an algorithm $\mathbb{A}$ that for every instance $(\mathbb{M}, x, n)$ decides whether $(\mathbb{M}, x, n) \in p$-DTM-INP-EXP-ACC$_\leq$ in time $c \cdot n^{f(\|\mathbb{M}\|+|x|)}$ for some $f : \mathbb{N} \to \mathbb{N}$ and $c \in \mathbb{N}$.

We denote by enc$(\mathbb{M})$ the encoding of the DTM $\mathbb{M}$ by a string in $\{0,1\}^*$. We consider the following DTM $\mathbb{M}_0$:

> $\mathbb{M}_0(x)$ // $x \in \{0,1\}^*$
>
> *1.* **if** x is not the encoding of a DTM, **then** reject
> *2.* determine the DTM $\mathbb{M}$ with $x = \text{enc}(\mathbb{M})$
> *3.* $m \leftarrow$ number of steps performed by $\mathbb{M}_0$ so far
> *4.* simulate at most 2^m steps of $\mathbb{A}$ on $(\mathbb{M}, x, m+3)$
> *5.* **if** the simulation does not halt, **then** $m \leftarrow m+1$ and goto 4
> *6.* **if** $\mathbb{A}$ accepts $(\mathbb{M}, x, m+3)$ in at most 2^m steps **then** reject **else** accept.

We finish the proof by a diagonal argument: We set $x_0 :=$ enc$(\mathbb{M}_0)$ and start $\mathbb{M}_0$ with input x_0. For sufficiently large $m \in \mathbb{N}$ we have $c \cdot (m+3)^{f(\|\mathbb{M}_0\|+|x_0|)} \leq 2^m$. Therefore eventually $\mathbb{M}_0$ with input x_0 reaches an m, we call it m_0, such that the simulation in Line 4 halts, more precisely,

$$\mathbb{A} \text{ halts on } (\mathbb{M}_0, x_0, m_0+3) \text{ in at most } 2^{m_0} \text{ steps.} \tag{11}$$

At that point the number of steps (of the run of $\mathbb{M}_0$ on input x_0) is bounded by 2^{m_0+2}. Hence

$$\mathbb{M}_0 \text{ on } x_0 \text{ halts in } \leq 2 + 2^{m_0+2} \leq 2^{m_0+3} \text{ steps.} \tag{12}$$

Putting all together we get the desired contradiction:

$\mathbb{M}_0$ accepts x_0
iff $\mathbb{M}_0$ accepts x_0 in $\leq 2^{m_0+3}$ steps (by (12))
iff $\mathbb{A}$ accepts $(\mathbb{M}_0, x_0, m_0+3)$ (by definition of $\mathbb{A}$)
iff $\mathbb{A}$ accepts $(\mathbb{M}_0, x_0, m_0+3)$ in at most 2^{m_0} steps (by (11))
iff $\mathbb{M}_0$ rejects x_0 (by Line 6 in the definition of $\mathbb{M}_0$).

□

8. The construction problem associated with p-ACC$_\leq$

We consider the construction problem associated with p-ACC$_\leq$:

> p-CONSTR-ACC$_\leq$
> *Instance:* An NTM $\mathbb{M}$ and $n \in \mathbb{N}$ in unary.
> *Parameter:* $\|\mathbb{M}\|$.
> *Problem:* Construct an accepting run of $\leq n$ steps of $\mathbb{M}$ started with empty input tape if there is one (otherwise report that there is no such run).

Similarly as we showed p-ACC$_\leq$ $\in$ FPT$_{\text{nu}}$ one gets that p-CONSTR-ACC$_\leq$ is nonuniformly fixed-parameter tractable (it should be clear what this means).

Definition 22. An *fpt*$_{\text{uni}}$ *Turing reduction* (*fpt Turing reduction*) from a parameterized construction problem (Q, κ) to a parameterized decision problem (Q', κ') is a deterministic algorithm $\mathbb{T}$ with an oracle to (Q', κ') solving the construction problem (Q, κ) and with the property that there are (computable) functions $f, g : \mathbb{N} \to \mathbb{N}$, and $c \in \mathbb{N}$ such that for every instance x of Q

- the run of $\mathbb{T}$ with input x has length $\leq f(\kappa(x)) \cdot |x|^c$;
- for every oracle query "$x' \in Q'$?" of the run of $\mathbb{A}$ with input x we have $\kappa(x') \leq g(\kappa(x))$.

Often a decision problem and its construction problem have the same complexity; for p-CONSTR-ACC$_\leq$ we can show:

Theorem 23. *(1) There is an fpt*$_{\text{uni}}$ *Turing reduction from* p-CONSTR-ACC$_\leq$ *to* p-ACC$_\leq$.

(2) If p-ACC$_\leq$ $\notin$ XP*, then there is* no *fpt Turing reduction from* p-CONSTR-ACC$_\leq$ *to* p-ACC$_\leq$.

Proof: (1) On an instance $(\mathbb{M}, n)$ of p-CONSTR-ACC$_\leq$ the desired reduction $\mathbb{T}$ first asks the oracle query "$(\mathbb{M}, n) \in p$-ACC$_\leq$?". If the answer is no, then $\mathbb{T}$ answers accordingly. Otherwise $\mathbb{T}$, by brute force, constructs an accepting run of at most n steps of $\mathbb{M}$. We analyze the running time of $\mathbb{T}$. For $m \in \mathbb{N}$ let $\mathbb{M}_1, \ldots, \mathbb{M}_\ell$ be the finitely many NTMs with $\|\mathbb{M}_i\| \leq m$ and with an accepting run started with empty input tape. Let ρ_i be such a run of $\mathbb{M}_i$ of minimum length. We set $f(m) := \max\{|\rho_1|, \ldots, |\rho_\ell|\}$. Now it is not hard to see that the running time of $\mathbb{T}$ on the instance $(\mathbb{M}, n)$ can be bounded by $\|\mathbb{M}\|^{O(f(\|\mathbb{M}\|))}$.

(2) By contradiction, assume there is an fpt Turing reduction $\mathbb{T}$ from p-CONSTR-ACC$_\leq$ to p-ACC$_\leq$. We show how $\mathbb{T}$ can be turned into an algorithm witnessing p-ACC$_\leq$ $\in$ XP.

According to the definition of fpt Turing reduction there are computable functions f, g and $c \in \mathbb{N}$ such that for every instance $(\mathbb{M}, n)$ of p-CONSTR-ACC$_\leq$, the algorithm $\mathbb{T}$ will only make queries "$(\mathbb{M}', n') \in p$-ACC$_\leq$?" with

$$\|\mathbb{M}'\| \leq g(\|\mathbb{M}\|) \quad \text{and} \quad n' \leq f(\|\mathbb{M}\|) \cdot n^c.$$

There are at most $2^{g(\|\mathbb{M}\|)+1}$ machines $\mathbb{M}'$ with $\|\mathbb{M}'\| \leq g(\|\mathbb{M}\|)$. For each such machine $\mathbb{M}'$ the answer to queries of the form "$(\mathbb{M}', n') \in p$-ACC$_\leq$?" with $n' \leq f(\|\mathbb{M}\|) \cdot n^c$ is determined by everyone of the following $f(\|\mathbb{M}\|) \cdot n^c + 1$ many statements: "the length of an accepting run of $\mathbb{M}'$ of minimum length is 1",…, "the length of an accepting run of $\mathbb{M}'$ of minimum length is $f(\|\mathbb{M}\|) \cdot n^c$", and "there is no accepting run of $\mathbb{M}'$ of length $\leq f(\|\mathbb{M}\|) \cdot n^c$." Therefore the table of theoretically possible answers contains at most

$$\big(f(\|\mathbb{M}\|) \cdot n^c + 1\big)^{2^{g(\|\mathbb{M}\|)+1}}$$

entries, that is $O(n^{h(\|\mathbb{M}\|)})$ many for some computable h. For each such possibility we simulate $\mathbb{T}$ by replacing the oracle queries accordingly. For those possibilities where $\mathbb{T}$ yields a purported accepting run of $\mathbb{M}$, we check whether it is really an accepting run of $\mathbb{M}$. $\square$

The previous theorem is a special case of a result holding for slicewise monotone problems. We will present it in the full version of the paper together with further applications.

9. Conclusions

We have studied the relationship between the complexity of the model-checking problems of the logics $L_=$ and $L_\leq$ and the complexity of the parameterized problems p-ACC$_=$ and p-ACC$_\leq$. We have introduced the assumption P[TC] $\neq$ NP[TC] and seen that it implies that p-ACC$_\leq$ $\notin$ FPT . We believe that a study of the strength of this assumption and of its consequences deserves further attention.

Acknowledgement. The first author is partially supported by the National Nature Science Foundation of China (60673049).

References

[1] Y. Aumann, Y. Dombb. Fixed structure complexity. In *IWPEC 2008*, LNCS 5018, 31–42, 2008.

[2] Y. Chen, J. Flum. On the complexity of Gödel's proof predicate. To appear in *The Journal of Symbolic Logic*, 2009.

[3] J. Flum, M. Grohe. *Parameterized Complexity Theory*, Springer, 2006.

[4] M. Grohe. The quest for a logic capturing PTIME. In *LICS'08*, pages 267–271, 2008.

[5] Y. Gurevich. Logic and the challenge of computer science. In *Current Trends in Theoretical Computer Science*, Computer Science Press, 1–57, 1988.

[6] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[7] A. Nash, J. Remmel, V. Vianu. PTIME queries revisited. In *ICDT 2005*, LNCS 3363, 274–288, 2005.

[8] M.Y. Vardi. The complexity of relational query languages. In *STOC'82*, 137–146, 1982.

[9] M.Y. Vardi. On the complexity of bounded-variable queries. In *PODS'95*, 266–276, 1995.