ON p-OPTIMAL PROOF SYSTEMS AND LOGICS FOR PTIME

YIJIA CHEN AND JÖRG FLUM

Abstract. We prove that TAUT has a p-optimal proof system if and only if a logic related to least fixed-point logic captures polynomial time on all finite structures. Furthermore, we show that TAUT has no *effectively* p-optimal proof system if $\mathrm{NTIME}(h^{O(1)}) \not\subseteq \mathrm{DTIME}(h^{O(\log h)})$ for every time constructible and increasing function h.

1. Introduction

As the title already indicates, this paper relates two topics which at first glance seem to be unrelated. On the one hand we consider optimal proof systems. A *proof system* in the sense of Cook and Reckhow [6], say for the class TAUT of tautologies of propositional logic, is a polynomial time computable function defined on $\{0,1\}^*$ and with TAUT as range. A proof system is *p-optimal* if it simulates any other proof system in polynomial time.[1] In their fundamental paper [13] Krajíček and Pudlák derive a series of statements equivalent to the existence of a p-optimal proof system for TAUT and state the conjecture:

Conjecture 1. There is no p-optimal proof system for TAUT.

On the other hand, the question of whether there is a logic capturing polynomial time remains the central open problem in descriptive complexity. There are artificial logics capturing polynomial time, but they do not fulfill a natural requirement to logics in this context:

(1) 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 $\mathcal{A}$.

If this condition is fulfilled for a logic capturing polynomial time, we speak of a P-bounded logic for P. In [10] Gurevich states the conjecture:

Conjecture 2. There is no P-bounded logic for P.

The conjecture is false if one waives the effectivity condition (1). This is shown in [10, Section 7, Claim 2] by considering a logic introduced by Blass and Gurevich and which we denote by $L_{\leq}$. For any vocabulary the sentences of $L_{\leq}$ are the sentences of least fixed-point logic in a vocabulary with an additional binary relation symbol for orderings. In $L_{\leq}$ for a structure $\mathcal{A}$ to be a model of φ it is

[1] All notions will be defined in a precise manner in Section 2.

required that in all structures of cardinality less than or equal to that of $\mathcal{A}$, the validity of φ (as a sentence of least fixed-point logic) does not depend on the chosen ordering, and $\mathcal{A}$ with some ordering satisfies φ.

As $L_{\leq}$ satisfies all requirements of a P-bounded logic for P except (1), Gurevich implicitly states the conjecture:

Conjecture 2a. $L_{\leq}$ is not a P-bounded logic for P.

The main result of this paper (cf. Theorem 6) tells us that

(2) Conjecture 1 is true $\iff$ Conjecture 2a is true.

We mentioned that at first glance "p-optimal proof systems for TAUT" and "logics for P" seem to be unrelated topics. However, there are reformulations of Conjecture 1 and Conjecture 2 that are alike. In fact, it is known [15] that TAUT has a p-optimal proof system if and only if there is a (computable) enumeration of all subsets of TAUT that are in P by means of Turing machines that decide them. And it is not hard to see that there is a P-bounded logic for P if and only if there is an enumeration of all polynomial time decidable classes of graphs closed under isomorphisms, again an enumeration in terms of Turing machines that decide these classes. In fact the question for a logic for P was stated in this way by Chandra and Harel [2] in the context of an analysis of the complexity and expressiveness of query languages.

Hence one consequence of (2) (which we only mention in this Introduction) is:

Theorem 1. *If there is an enumeration of all polynomial time decidable subsets of* TAUT*, then there is an enumeration of all polynomial time decidable classes of graphs closed under isomorphisms.*

Using a special feature of the semantics of the logic $L_{\leq}$, one can construct (cf. Proposition 11) a logic that is an *effectively* P-bounded logic for P, if $L_{\leq}$ is a P-bounded logic for P. Here this "effectively" means that in (1) we can *compute* from φ a polynomial bounding the time to decide whether $\mathcal{A}$ is a model of φ. In this way we can strengthen the conclusion of Theorem 1 by requiring that every Turing machine in the enumeration comes with a polynomial time clock. Apparently this is a strengthening, while from any enumeration of the polynomial time decidable subsets of TAUT we obtain one with polynomial time clocks in a trivial manner, namely by systematically adding such clocks.

In general, the experts tend to believe Conjecture 1, as the existence of a p-optimal proof system for TAUT would have various consequences which seem to be unlikely (see [12, 13]). It is worthwhile to emphasize that we show that Conjecture 1 is equivalent to Conjecture 2a and do not claim its equivalence to Conjecture 2. The situation with Conjecture 2 is quite different; no known consequences of the existence of a P-bounded logic for P seem to be implausible. Moreover, due to results showing that there are logics capturing polynomial time

on always larger classes of structures, Grohe [9] "mildly leans towards believing" that there is a P-bounded logic for P.

In [3] we have shown that $L_{\leq}$ is not an effectively P-bounded logic for P under the assumption NP[TC] $\not\subseteq$ P[TC$^{\log \text{TC}}$], which means that NTIME($h^{O(1)}$) $\not\subseteq$ DTIME($h^{O(\log h)}$) for every time constructible and increasing function h. Under this assumption, we get (see Theorem 15) that TAUT has no effectively p-optimal proof system. Here a proof system P for TAUT is *effectively* p-optimal if from every other proof system for TAUT we can *compute* a polynomial time simulation by P.

On the other hand, Krajíček and Pudlák [13] showed, assuming E = NE, that TAUT has a p-optimal proof system. Using our result [3] that under the assumption E = NE the logic $\big(L_{=}$ and hence$\big)$ $L_{\leq}$ is an effectively P-bounded logic for P, we can derive (see Corollary 17) that TAUT has an *effectively* p-optimal proof system if E = NE.

In [5] we extract the main idea underlying the proof of (2), apply it to other problems, and generalize it to the "nondeterministic case," thus obtaining statements equivalent to the existence of an optimal (not necessarily p-optimal) proof system for TAUT.

2. Preliminaries

In this section we recall concepts and results from complexity theory and logic that we will use later and fix some notation.

2.1. Complexity. We denote the alphabet $\{0,1\}$ by Σ. The length of a string $x \in \Sigma^*$ is denoted by $|x|$. We identify problems with subsets Q of Σ^*. Clearly, as done mostly, we present concrete problems in a verbal, hence uncodified form. We denote by P the class of problems Q such that $x \in Q$ is solvable in polynomial time.

All Turing machines have Σ as their alphabet and are deterministic ones if not stated otherwise explicitly. If necessary we will not distinguish between a Turing machine and its code, a string in Σ^*. If $\mathbb{M}$ is a Turing machine we denote by $\|\mathbb{M}\|$ the length of its code.

By $m^{O(1)}$ we denote the class of polynomially bounded functions from $\mathbb{N}$ to $\mathbb{N}$. Sometimes statements containing a formulation like "there is $d \in \mathbb{N}$ such that for all $x \in \Sigma^* : \ \ldots \ \leq |x|^d$" can be wrong for $x \in \Sigma^*$ with $|x| \leq 1$. We trust the reader's common sense to interpret such statements reasonably.

Optimal proof systems, almost optimal algorithms and enumerations of P-easy subsets. A *proof system* for a problem $Q \subseteq \Sigma^*$ is a surjective function $P\colon \Sigma^* \to Q$ computable in polynomial time. The proof system P for Q is *polynomially optimal* or *p-optimal* if for every proof system P' for Q there is a polynomial time computable $T\colon \Sigma^* \to \Sigma^*$ such that for all $w \in \Sigma^*$

$$P(T(w)) = P'(w).$$

If $\mathbb{A}$ is any algorithm we denote by $t_{\mathbb{A}}(x)$ the number of steps of the run of $\mathbb{A}$ on input x; if $\mathbb{A}$ on x does not stop, then $t_{\mathbb{A}}(x)$ is not defined.

An algorithm $\mathbb{A}$ deciding Q is *almost optimal* or *optimal on positive instances of* Q if for every algorithm $\mathbb{B}$ deciding Q there is a polynomial $p \in \mathbb{N}[X]$ such that for all $x \in Q$

$$t_{\mathbb{A}}(x) \leq p(t_{\mathbb{B}}(x) + |x|)$$

(note that nothing is required of the relationship between $t_{\mathbb{A}}(x)$ and $t_{\mathbb{B}}(x)$ for $x \notin Q$).

By definition a subset Q' of Q is P-*easy* if $Q' \in \mathrm{P}$. An *enumeration of* P-*easy subsets of* Q is a computable function $M\colon \mathbb{N} \to \Sigma^*$ such that

- for every $i \in \mathbb{N}$ the string $M(i)$ is a polynomial time Turing machine deciding a P-easy subset of Q;
- for every P-easy subset Q' of Q there is $i \in \mathbb{N}$ such that $M(i)$ decides Q'.

We denote by TAUT the class of tautologies of propositional logic. The following theorem is well-known (cf. [13] for the equivalence of the first two statements and [15] for the equivalence to the third one):

Theorem 2. *The following are equivalent:*
(1) TAUT *has a p-optimal proof system.*
(2) TAUT *has an almost optimal algorithm.*
(3) TAUT *has an enumeration of the* P-*easy subsets.*

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 $R \in \tau$. *All structures in this paper are assumed to have finite universe.*

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 a string in Σ^* (e.g., cf. [8] for details). We only consider properties of structures that are invariant under isomorphisms, so it suffices that from the encoding of $\mathcal{A}$ we can recover $\mathcal{A}$ up to isomorphism. We can assume that there is a computable function lgth such that for every vocabulary τ and $m \geq 1$:

- $\|\mathcal{A}\| = \mathrm{lgth}(\tau, m)$ for every τ-structure $\mathcal{A}$ with universe of cardinality m;
- for fixed τ, the function $m \mapsto \mathrm{lgth}(\tau, m)$ is computable in time $m^{O(1)}$;
- $\mathrm{lgth}(\tau \cup \{R\}, m) = O(\mathrm{lgth}(\tau, m) + m^r)$ for every r-ary relation symbol R not in τ.

We assume familiarity with first-order logic and its extension *least fixed-point logic* LFP (e.g. see [7]). We denote by $\mathrm{LFP}[\tau]$ the set of sentences of vocabulary τ of LFP. As we will introduce further semantics for the formulas of least fixed-point logic, we write $\mathcal{A} \models_{\mathrm{LFP}} \varphi$ 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 [17]):

Proposition 3. *The model-checking problem* $\mathcal{A} \models_{\mathrm{LFP}} \varphi$ *for structures* $\mathcal{A}$ *and* LFP*-sentences* φ *can be solved in time*

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

Logics capturing polynomial time. For our purposes a *logic* L consists

- of an algorithm that for every vocabulary τ and every string ξ decides whether ξ is in the set $L[\tau]$, the set of L*-sentences of vocabulary* τ;
- of a *satisfaction relation* $\models_L$; if $(\mathcal{A}, \varphi) \in \models_L$, then $\mathcal{A}$ is a τ-structure and $\varphi \in L[\tau]$ for some vocabulary τ; furthermore for each τ and $\varphi \in L[\tau]$ 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 $\mathcal{A} \models_L \varphi$ $\big($that is, if $(\mathcal{A}, \varphi) \in \models_L\big)$. We set $\mathrm{Mod}_L(\varphi) := \{\mathcal{A} \mid \mathcal{A} \models_L \varphi\}$ and say that φ *axiomatizes* the class $\mathrm{Mod}_L(\varphi)$.

Definition 4. 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

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

(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 $\mathcal{A} \models_L \varphi$) and if moreover $\mathbb{A}$, for every fixed φ, 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 $\mathrm{Mod}_L(\varphi) \in \mathrm{P}$. However, we do not necessarily know ahead of time a 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.

The logic $L_\leq$ and invariant sentences. In this section we introduce the logic $L_\leq$, a variant of least fixed-point logic.

For every vocabulary τ we let $\tau_< := \tau \cup \{<\}$, where $<$ is a binary relation symbol not in τ chosen in some canonical way. We set

$$L_\leq[\tau] = \mathrm{LFP}[\tau_<]$$

for every vocabulary τ. Before we define the satisfaction relation for $L_\leq$ we introduce the notion of $\leq m$-invariant sentence.

Definition 5. Let φ be an $L_{\leq}[\tau]$-sentence.

– For $m \geq 1$ we say that φ is $\leq m$*-invariant* if for all structures $\mathcal{A}$ with $|A| \leq m$ and all orderings $<_1$ and $<_2$ on A we have

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

– φ is *invariant* if it is $\leq m$-invariant for all $m \geq 1$.

Finally we introduce the semantics for the logic $L_{\leq}$ by

$$\mathcal{A} \models_{L_{\leq}} \varphi \iff \Big(\varphi \text{ is} \leq |A|\text{-invariant and } (\mathcal{A}, <) \models_{\mathrm{LFP}} \varphi \text{ for some ordering} < \text{on } A\Big).$$

Immerman [11] and Vardi [16] have shown that LFP is an effectively P-bounded logic for P *on the class of ordered structures*, a result we will not need in the proof of our main theorem. However, using it one can easily show that $L_{\leq}$ is a logic for P.

For later purposes we remark that for every $L_{\leq}[\tau]$-sentence φ and $m \geq 1$ we have

$$\varphi \text{ is} \leq m\text{-invariant} \iff \neg\varphi \text{ is} \leq m\text{-invariant},$$

and thus for every τ-structure $\mathcal{A}$

$$\varphi \text{ is} \leq |A|\text{-invariant} \iff \big(\mathcal{A} \models_{L_{\leq}} \varphi \text{ or } \mathcal{A} \models_{L_{\leq}} \neg\varphi\big).$$

In particular,

$$\varphi \text{ is} \leq m\text{-invariant} \iff \big(\mathcal{A}(\tau, m) \models_{L_{\leq}} \varphi \text{ or } \mathcal{A}(\tau, m) \models_{L_{\leq}} \neg\varphi\big),$$

where $\mathcal{A}(\tau, m)$ is the τ-structure with universe $\{1, \ldots, m\}$, where every relation symbol in τ is interpreted by the empty relation of the corresponding arity.

Finally we remark that it can happen for $L_{\leq}$-sentences φ and ψ and a structure $\mathcal{A}$ that $\mathcal{A} \models_{L_{\leq}} (\varphi \wedge \psi)$ but neither $\mathcal{A} \models_{L_{\leq}} \varphi$ nor $\mathcal{A} \models_{L_{\leq}} \psi$.

3. The main theorem

In this section we want to show:

Theorem 6. TAUT *has a p-optimal proof system iff* $L_{\leq}$ *is a* P*-bounded logic for* P*.*

In view of Theorem 2 we get one direction of Theorem 6 with the following lemma.

Lemma 7. *If* $L_{\leq}$ *is a* P*-bounded logic for* P*, then there is an enumeration of the* P*-easy subsets of* TAUT*.*

Proof. It is easy to introduce a vocabulary τ such that in polynomial time we can associate with every propositional formula α a τ-structure $\mathcal{A}(\alpha)$ such that

– every propositional variable X of α corresponds to two distinct elements a_X, b_X of $\mathcal{A}(\alpha)$ and there is a unary relation symbol $P \in \tau$ such that $P^{\mathcal{A}(\alpha)} = \{a_X \mid X \text{ variable of } \alpha\}$;

– there is an LFP-sentence $\varphi(\mathrm{PROP})$ of vocabulary τ axiomatizing the class

$$\{\mathcal{B} \mid \mathcal{B} \cong \mathcal{A}(\alpha) \text{ for some } \alpha \in \mathrm{PROP}\}$$

(by PROP we denote the class of formulas of propositional logic);

– if $\mathcal{B} \models \varphi(\mathrm{PROP})$, then one can determine the unique $\alpha \in \mathrm{PROP}$ with $\mathcal{B} \cong \mathcal{A}(\alpha)$ in polynomial time.

Again let $\tau_< := \tau \cup \{<\}$ with a new binary $<$. Note that a $\tau_<$-structure of the form $(\mathcal{A}(\alpha), <)$ yields an assignment of the variables of α, namely the assignment sending a variable X to TRUE if and only if $a_X < b_X$. There is an $\mathrm{LFP}[\tau_<]$-formula $\varphi(\mathrm{sat})$ that for every $\alpha \in \mathrm{PROP}$ expresses in $(\mathcal{A}(\alpha), <)$ that the assignment given by $<$ satisfies α.

We introduce the $L_\leq[\tau]$-sentence

$$\varphi_0 := \big(\varphi(\mathrm{PROP}) \to \varphi(\mathrm{sat})\big).$$

By the definition of $\models_{L_\leq}$ we see that for every $\alpha \in \mathrm{PROP}$ and every $L_\leq[\tau]$-sentence φ

$$\text{if } \mathcal{A}(\alpha) \models_{L_\leq} (\varphi_0 \wedge \varphi), \text{ then } \alpha \in \mathrm{TAUT}. \tag{3}$$

We claim that the class of models of $(\varphi_0 \wedge \varphi)$, more precisely,

$$Q(\varphi) := \{\alpha \in \mathrm{PROP} \mid \mathcal{A}(\alpha) \models_{L_\leq} (\varphi_0 \wedge \varphi)\},$$

where φ ranges over all $L_\leq[\tau]$-sentences, yields the desired enumeration of P-easy subsets of TAUT. By (3), we have $Q(\varphi) \subseteq \mathrm{TAUT}$.

For $\varphi \in L_\leq[\tau]$ let the Turing machine $\mathbb{M}_\varphi$, given an input $\alpha \in \mathrm{PROP}$, first construct $\mathcal{A}(\alpha)$ and then check whether $\mathcal{A}(\alpha) \models_{L_\leq} (\varphi_0 \wedge \varphi)$. Clearly, $\mathbb{M}_\varphi$ decides $Q(\varphi)$ and does this in polynomial time, as $L_\leq$ is a P-bounded logic for P.

Conversely, let Q be a P-easy subset of TAUT. If Q is finite, it is easy to see that $Q = Q(\varphi)$ for some $\varphi \in L_\leq[\tau]$. Now let Q be infinite. The class

$$\{\mathcal{B} \mid \mathcal{B} \cong \mathcal{A}(\alpha) \text{ for some } \alpha \in Q\}$$

is in P, and therefore it is axiomatizable by an $L_\leq[\tau]$-sentence φ. As the class contains arbitrarily large structures, the formula φ is invariant. We show that $Q = Q(\varphi)$.

Assume first that $\alpha \in Q(\varphi)$, i.e., $\mathcal{A}(\alpha) \models_{L_\leq} (\varphi_0 \wedge \varphi)$. Then, by invariance of φ, we have $\mathcal{A}(\alpha) \models_{L_\leq} \varphi$ and thus $\alpha \in Q$. Conversely, assume that $\alpha \in Q$. Then $\mathcal{A}(\alpha) \models_{L_\leq} \varphi$. As $\alpha \in \mathrm{TAUT}$, in order to get $\mathcal{A}(\alpha) \models_{L_\leq} (\varphi_0 \wedge \varphi)$ $\big($and hence, $\alpha \in Q(\varphi)\big)$ it suffices to show that $(\varphi_0 \wedge \varphi)$ is $\leq |A(\alpha)|$-invariant. So let $\mathcal{B}$ be a τ-structure with $|B| \leq |A(\alpha)|$. If $\mathcal{B} \not\models_{L_\leq} \varphi$, then, by invariance of φ, we have $(\mathcal{B}, <^B) \not\models_{\mathrm{LFP}} (\varphi_0 \wedge \varphi)$ for all orderings $<^B$ on B; if $\mathcal{B} \models_{L_\leq} \varphi$, then $\mathcal{B} \cong \mathcal{A}(\beta)$ for some $\beta \in Q \subseteq \mathrm{TAUT}$. Hence, $(\mathcal{B}, <^B) \models_{\mathrm{LFP}} (\varphi_0 \wedge \varphi)$ for all orderings $<^B$ on B. □

Remark 8. In the previous proof we have used the definition of the satisfaction relation $\models_{L_\leq}$ in order to express the universal second-order quantifier in the statement "all assignments satisfy α." Similarly, we can do with every Π^1_1-sentence $\forall R\varphi$, where φ is a first-order formula or (equivalently) LFP-formula and show in this way that there is an enumeration of the P-easy subsets closed under isomorphisms of the class of models of $\forall R\varphi$, if $L_\leq$ is a P-bounded logic for P. In fact, let k be the arity of R. If a structure $\mathcal{A}$ has n elements, we consider a structure $\mathcal{B}$ with additional disjoint unary relations $U^\mathcal{B}, P_0^\mathcal{B}, P_1^\mathcal{B}$ such that

$$B = U^\mathcal{B} \cup P_0^\mathcal{B} \cup P_1^\mathcal{B}, \quad U^\mathcal{B} = A, \quad |P_0^\mathcal{B}| = n^k \quad |P_1^\mathcal{B}| = n^k$$

and with an ordering $<^\mathcal{B}$.

With the elements in $P_0^\mathcal{B}$ interpreted as 0s and the elements in $P_1^\mathcal{B}$ interpreted as 1s, the first n^k-elements of the ordering in $P_0^\mathcal{B} \cup P_1^\mathcal{B}$ represent a natural number $< 2^{n^k}$ and thus a k-ary relation R on A, which we can compute in polynomial time (polynomial in n); hence we can define R by an LFP-formula. As in this way, by changing the ordering, we have access to all such k-ary relations R on A, we can express the quantifier $\forall R$ using the invariance requirement of $\models_{L_\leq}$.

For example, let C be the class of all pairs $(\mathcal{G}, \mathcal{H})$ of graphs such that $\mathcal{H}$ is *not* a homomorphic image of $\mathcal{G}$. By the previous observation, we see that there is an enumeration of the P-easy subclasses of C closed under isomorphisms if $L_\leq$ is a P-bounded logic for P. Of course, a subclass D of C is closed under isomorphisms if

$$\mathcal{G} \cong \mathcal{G}', \quad \mathcal{H} \cong \mathcal{H}' \text{ and } (\mathcal{G}, \mathcal{H}) \in D \text{ imply } (\mathcal{G}', \mathcal{H}') \in D.$$

As the models of such a Π^1_1-sentence corresponds to a problem Q in co-NP, a simple complexity-theoretic argument shows that there is an enumeration of the P-easy subsets of Q provided there is one for the P-easy subsets of TAUT (see also [1]). However, in this way, in the previous example we would not get an enumeration of those P-easy subclasses that are *closed under isomorphisms*.

In view of Theorem 2 the remaining direction in Theorem 6 is provided by the following result.

Lemma 9. *If* TAUT *has an almost optimal algorithm, then* $L_\leq$ *is a* P*-bounded logic for* P*.*

Proof. We assume that TAUT has an almost optimal algorithm $\mathbb{O}$ and have to show that there is an algorithm that decides $\mathcal{B} \models_{L_\leq} \varphi$ and does this for fixed φ in time $\|\mathcal{B}\|^{O(1)}$.

By the definition of $\mathcal{B} \models_{L_\leq} \varphi$ and Proposition 3 it suffices to show the existence of an algorithm $\mathbb{A}$ that for every $L_\leq$-sentence φ and every $m \in \mathbb{N}$ decides whether φ is $\leq m$-invariant and does this for fixed φ in time $m^{O(1)}$.

We set

$$Q := \Big\{ \Big(\chi, \ell, \mathrm{lgth}(\tau,\ell)^{|\chi|}\Big) \;\Big|\; \tau \text{ a vocabulary, } \chi \in \mathrm{LFP}[\tau],\ \ell \geq 1,\ \mathrm{lgth}(\tau,\ell)^{|\chi|} \text{ in unary, there is a } \tau\text{-structure } \mathcal{B} \text{ with } \big(|B| \leq \ell \text{ and } \mathcal{B} \models_{\mathrm{LFP}} \chi\big) \Big\}$$

(compare Section 2.2 for the definition of the function lgth). By Proposition 3, $Q \in \mathrm{NP}$. Thus there is a polynomial time reduction $R : Q \leq^p \mathrm{SAT}$. We can assume that from $R(x)$ we can recover x in polynomial time.
Let φ be an $L_{\leq}[\tau]$-sentence. Then

$$\begin{aligned}
\varphi \text{ is not } \leq m\text{-invariant} &\iff \text{there is a } \tau\text{-structure } \mathcal{B} \text{ and orderings } <_1, <_2 \text{ with } \big(|B| \leq m \text{ and } (\mathcal{B}, <_1, <_2) \models_{\mathrm{LFP}} \underbrace{(\varphi(<_1) \wedge \neg\varphi(<_2))}_{\varphi^*}\big) \\
&\iff \big(\varphi^*, m, \mathrm{lgth}(\tau \cup \{<_1,<_2\}, m)^{|\varphi^*|}\big) \in Q \\
&\iff R\big(\varphi^*, m, \mathrm{lgth}(\tau \cup \{<_1,<_2\}, m)^{|\varphi^*|}\big) \in \mathrm{SAT}.
\end{aligned}$$

We set $\alpha(\varphi, m) := R\big(\varphi^*, m, \mathrm{lgth}(\tau \cup \{<_1,<_2\}, m)^{|\varphi^*|}\big)$. Hence

$$\varphi \text{ is } \leq m\text{-invariant} \iff \neg\alpha(\varphi, m) \in \mathrm{TAUT}. \tag{4}$$

It is clear that there is an algorithm that on input (φ, m) computes $\alpha(\varphi, m)$ and for fixed φ

$$\text{it computes } \alpha(\varphi, m) \text{ in time } m^{O(1)}, \text{ in particular, } |\alpha(\varphi, m)| \leq m^{O(1)}, \tag{5}$$

as for fixed τ, the function $m \mapsto \mathrm{lgth}(\tau, m)$ is polynomial in m.

Let $\mathbb{S}$ be the algorithm that on input φ by systematically going through all τ-structures with universe $\{1\}$, all with universe $\{1,2\}$,... and all orderings of these universes computes $m(\varphi) :=$ the least m such that φ is not $\leq m$-invariant. If φ is invariant, then $m(\varphi)$ is not defined and $\mathbb{S}$ does not stop.

We show that the following algorithm $\mathbb{A}$ has the desired properties.

$\mathbb{A}(\varphi, m)$
// φ an $L_{\leq}$-sentence, $m \in \mathbb{N}$
1. Compute $\alpha(\varphi, m)$.
2. In parallel simulate $\mathbb{S}$ on input φ and $\mathbb{O}$ on input $\neg\alpha(\varphi, m)$.
3. **if** $\mathbb{O}$ stops first, **then** output its answer.
4. **if** $\mathbb{S}$ stops first, **then**
5. **if** $m < m(\varphi)$ **then** accept **else** reject.

By our assumptions on $\mathbb{O}$ and $\mathbb{S}$ and by (4), it should be clear that $\mathbb{A}$ on input (φ, m) decides whether φ is $\leq m$-invariant. We have to show that for fixed φ it does it in time $m^{O(1)}$.

Case "φ is invariant": Then for all m we have $\neg\alpha(\varphi, m) \in \mathrm{TAUT}$. Thus the following algorithm $\mathbb{O}_\varphi$ decides TAUT: on input $\beta \in \mathrm{PROP}$ the algorithm $\mathbb{O}_\varphi$

checks whether $\beta = \neg\alpha(\varphi, m)$ for some $m \geq 1$. If so, it accepts and otherwise it runs $\mathbb{O}$ on input β and answers accordingly. By (5), we have

$$t_{\mathbb{O}_\varphi}(\neg\alpha(\varphi, m)) \leq m^{O(1)}. \tag{6}$$

As $\mathbb{O}$ is optimal, we know that there is a constant d such that for all $\beta \in \mathrm{TAUT}$

$$t_{\mathbb{O}}(\beta) \leq \big(|\beta| + t_{\mathbb{O}_\varphi}(\beta)\big)^d. \tag{7}$$

In particular, we have

$$t_{\mathbb{O}}(\neg\alpha(\varphi, m)) \leq \big(|\neg\alpha(\varphi, m)| + t_{\mathbb{O}_\varphi}(\neg\alpha(\varphi, m))\big)^d \leq m^{O(1)}.$$

By this inequality, (5) and (6), we see that for invariant φ we have $t_{\mathbb{A}}(\varphi, m) \leq m^{O(1)}$.

Case "φ is not invariant": Then $\mathbb{S}$ will stop on input φ. Thus, in the worst case, $\mathbb{A}$ on input (φ, m) has to wait till the simulation of $\mathbb{S}$ on φ stops and then must check whether the result $m(\varphi)$ of the computation of $\mathbb{S}$ is bigger than m or not and answer accordingly. So the algorithm $\mathbb{A}$ at most takes time $m^{O(1)} + O(t_{\mathbb{S}}(\varphi) + m) \leq m^{O(1)}$ (note that we fix φ, so that $t_{\mathbb{S}}(\varphi)$ is a constant). □

Corollary 10. *If* TAUT *has a p-optimal proof system, then there is an effectively* P*-bounded logic for* P*.*

This result follows from Theorem 6 using the following proposition:

Proposition 11. *If $L_\leq$ is a* P*-bounded logic for* P*, then there is an effectively* P*-bounded logic for* P*.*

Proof. In Section 2.2 we have seen that for every $L_\leq$-sentence φ and $m \geq 1$ it holds that

$$\varphi \text{ is } \leq m\text{-invariant} \iff \big(\mathcal{A}(\tau, m) \models_{L_\leq} \varphi \text{ or } \mathcal{A}(\tau, m) \models_{L_\leq} \neg\varphi\big), \tag{8}$$

where $\mathcal{A}(\tau, m)$ denotes the "empty structure" of vocabulary τ with universe $\{1, \ldots, m\}$.

Now assume that $L_\leq$ is a P-bounded logic for P and let $\mathbb{A}$ be an algorithm witnessing that $L_\leq$ is a P-bounded logic for P. By (8), there is a function h assigning to every $L_\leq$-sentence φ a polynomial $h(\varphi) \in \mathbb{N}[X]$ such that $\mathbb{A}$ decides whether φ is $\leq m$-invariant in time $h(\varphi)(m)$.

We consider the logic $T(L_\leq)$, *time-clocked* $L_\leq$, defined as follows:

- for every vocabulary τ

$$T(L_\leq)[\tau] := \{(\varphi, p) \mid \varphi \in L_\leq[\tau] \text{ and } p \in \mathbb{N}[X]\};$$

- $\mathcal{A} \models_{T(L_\leq)} (\varphi, p)$ iff (a) and (b) are fulfilled, where
  - (a) $\mathbb{A}$ shows via (8) in $\leq p(|A|)$ steps that φ is $\leq |A|$-invariant;
  - (b) $(\mathcal{A}, <) \models_{\mathrm{LFP}} \varphi$ for some ordering $<$, say with the ordering of A given by the encoding of $\mathcal{A}$.

It is not hard to verify that $T(L_\leq)$ is an effectively P-bounded logic for P. □

Remark 12. In a slightly different way but using the same idea one can define the time-clocked version $T(L)$ for any P-bounded logic L for P. However, in general, $T(L)$ is not even a logic, as it can happen that the class of models of a $T(L)$-sentence is not closed under isomorphisms. In the case of $T(L_{\leq})$ this is guaranteed by the fact that condition (a) in the definition of $\mathcal{A} \models_{T(L_{\leq})} (\varphi, p)$ only refers to the cardinality of the universe of $\mathcal{A}$.

There is a further consequence of Theorem 6. By a reformulation of the statement "$L_{\leq}$ is a P-bounded logic for P" due to Nash et al. [14] (see [3] for a proof), we get:

Theorem 13. *The following are equivalent:*
(a) TAUT *has a p-optimal proof system.*
(b) *There is an algorithm deciding for every nondeterministic Turing machine* $\mathbb{M}$ *and every natural number* m *whether* $\mathbb{M}$ *accepts the empty input tape in* $\leq m$ *steps and the algorithm does this for every fixed* $\mathbb{M}$ *in time* $m^{O(1)}$.

4. Effective versions

Let $\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{P}[\mathrm{TC}^{\log \mathrm{TC}}]$ mean that $\mathrm{NTIME}(h^{O(1)}) \not\subseteq \mathrm{DTIME}(h^{O(\log h)})$ for every time constructible and increasing function h. In [3] we have shown:

Proposition 14. *Assume that* $\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{P}[\mathrm{TC}^{\log \mathrm{TC}}]$. *Then* $L_{\leq}$ *is not an effectively* P*-bounded logic for* P.

Are there *natural* effective versions of the properties of TAUT listed in Theorem 2 equivalent to the statement "$L_{\leq}$ is not an effectively P-bounded logic for P" and which therefore, by Proposition 14, could not hold under the assumption $\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{P}[\mathrm{TC}^{\log \mathrm{TC}}]$? We did not find them. However, by analyzing the proof of Proposition 14, we isolate a property of an effectively P-bounded logic for P that cannot be fulfilled if $\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{P}[\mathrm{TC}^{\log \mathrm{TC}}]$. It turns out that this is equivalent to natural effective versions of the properties on TAUT under consideration. We already state the result we aim at and then define the concepts appearing in it and present the generalization of Theorem 2 on which its proof is based. Due to space limitations all proofs of results in this section will be given in the full version of the paper.

Theorem 15. *If* $\mathrm{NP}[\mathrm{TC}] \not\subseteq \mathrm{P}[\mathrm{TC}^{\log \mathrm{TC}}]$, *then* TAUT *has no effectively p-optimal proof system.*

Let $Q \subseteq \Sigma^*$. A proof system P for Q is *effectively p-optimal* if there are two computable functions $S\colon \Sigma^* \times \mathbb{N}[X] \to \Sigma^*$ and $b\colon \Sigma^* \times \mathbb{N}[X] \to \mathbb{N}[X]$ such that for every proof system P' for Q with time bound $p \in \mathbb{N}[X]$ and every $w' \in \Sigma^*$, we have

$$P'(w') = P\big(S(P', p)(w')\big),$$

where $S(P', p)$ is (the code of) a Turing machine with time bound $b(P', p)$ and $S(P', p)(w')$ denotes the output of $S(P', p)$ on input w'.

An algorithm $\mathbb{A}$ deciding Q is *effectively almost optimal* if there is a computable function $b\colon \Sigma^* \to \mathbb{N}[X]$ such that for every algorithm $\mathbb{B}$ deciding Q we have for every $x \in Q$ we have

$$t_{\mathbb{A}}(x) \leq b(\mathbb{B})\big(t_{\mathbb{B}}(x) + |x|\big).$$

We say that Q has an *effective enumeration of* P*-easy subsets*, if it has an enumeration $M\colon \mathbb{N} \to \Sigma^*$ of P-easy subsets of Q such that there are functions $I\colon \Sigma^* \times \mathbb{N}[X] \to \mathbb{N}$ and $: \Sigma^* \times \mathbb{N}[X] \to \mathbb{N}[X]$ such that for every Turing machine $\mathbb{M}$ and polynomial $p \in \mathbb{N}[X]$,

> if the Turing machine $\mathbb{M}$ recognizes a subset $Q' \subseteq Q$ with time bound p, then the machine $M(I(\mathbb{M}, p))$ recognizes Q' with time bound $b(\mathbb{M}, p)$.

We can prove the effective analogue of Theorem 2:

Theorem 16. *The following are equivalent:*
(1) TAUT *has an effectively p-optimal proof system.*
(2) TAUT *has an effectively almost optimal algorithm.*
(3) TAUT *has an effective enumeration of the P-easy subsets.*

In [3] we have shown that if E = NE, then (the logic $L_=$ and hence) $L_\leq$ are effectively P-bounded logics for P. The proof of the previous result shows that TAUT has an effectively p-optimal proof system if $L_\leq$ is an effectively P-bounded logic for P. Therefore we obtain the following "effective version" of a result due to Krajíček and Pudlák.

Corollary 17. *If* E = NE*, then* TAUT *has an effectively p-optimal proof system.*

Acknowledgement. This research has been partially supported by the National Nature Science Foundation of China (60970011), the Sino-German Center for Research Promotion (GZ400), and the John Templeton Foundation through Project # 13152, the Infinity Project at the Centre de Recerca Matemàtica.

References

[1] O. Beyersdorff and Z. Sadowski. Characterizing the existence of optimal proof systems and complete sets for promise classes. In *Proceedings of the 4th Computer Science Symposium in Russia (CSR'09)*, Lecture Notes in Computer Science 5675, 47–58, 2009.
[2] A.K. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.
[3] Y. Chen and J. Flum. A logic for PTIME and a parameterized halting problem. In *Proceedings of the 24th IEEE Symposium on Logic in Computer Science (LICS'09)*, pages 397–406, 2009.
[4] Y. Chen and J. Flum. On the complexity of Gödel's proof predicate. *The Journal of Symbolic Logic*, 75(1): 239–254, 2010.
[5] Y. Chen and J. Flum. On slicewise monotone parameterized problems and optimal proof systems for TAUT. Available at `http://basics.sjtu.edu.cn/~chen/papers`, 2010.
[6] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44:36–50, 1979.
[7] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*, 2nd edition, Springer, 1999.

[8] J. Flum and M. Grohe. *Parameterized Complexity Theory*, Springer, 2006.
[9] M. Grohe. Fixed-point definability and polynomial time. In *Proceedings of the 23rd International Workshop on Computer Science Logic (CSL'09)*, Lecture Notes in Computer Science 5771, pages 20–23, 2009.
[10] Y. Gurevich. Logic and the challenge of computer science. In *Current Trends in Theoretical Computer Science*, Computer Science Press, 1–57, 1988.
[11] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.
[12] J. Köbler, J. Messner, and J. Torán. Optimal proof systems imply complete sets for promise classes. *Information and Computation*, 184:71–92, 2003.
[13] J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54:1063–1088, 1989.
[14] A. Nash, J. Remmel, and V. Vianu. PTIME queries revisited. In *Proceedings of the 10th International Conference on Database Theory (ICDT'05)*, T. Eiter and L. Libkin (eds.), Lecture Notes in Computer Science 3363:274–288, 2005.
[15] Z. Sadowski. On an optimal propositionl proof system and the structure of easy subsets. *Theoretical Computer Science*, 288(1):181–193, 2002.
[16] M.Y. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing (STOC'82)*, pages 137–146, 1982.
[17] M.Y. Vardi. On the complexity of bounded-variable queries. In *Proceedings of the 14th ACM Symposium on Principles of Database Systems (PODS'95)*, pages 266–276, 1995.

Yijia Chen
Department of Computer Science
Shanghai Jiaotong University
China
E-mail address: yijia.chen@cs.sjtu.edu.cn

Jörg Flum
Mathematisches Institut
Albert-Ludwigs Universität Freiburg
Germany
E-mail address: joerg.flum@math.uni-freiburg.de