Structure with fast elimination of quantifiers

Mihai Prunescu *

Abstract

A structure of finite signature is constructed so that: for all existential formulas $\exists \vec{y}\ \varphi(\vec{x}, \vec{y})$ and for all tuples of elements $\vec{u}$ of the same length as the tuple $\vec{x}$, one can decide in a quadratic time depending only on the length of the formula, if $\exists \vec{y}\ \varphi(\vec{u}, \vec{y})$ holds in the structure. In other words, the structure satisfies the relativized model-theoretic version of P=NP in the sense of [4]. This is a model-theoretical approach to results of Hemmerling and Gaßner.
A.M.S.-Classification: 03B05, 03B25.

1 Introduction

The technic and many objects used here were introduced by Bruno Poizat in [5] where he constructed a structure with truth-predicate for existential formulas with one free variable using a linear time coding procedure. Poizat considers a structure in the language of two independent successor-functions (see Section 2 below) plus a unary predicate V; every existential formula $\psi(x) = \exists \vec{y}\ \varphi(x, \vec{y})$ is coded by a linear-size term $\tau_\psi(x)$, such that the following holds:

$$\forall\, x\ [\exists \vec{y}\ \varphi(x, \vec{y}) \leftrightarrow V(\tau_\psi(x))].$$

Since there are only unary functions, this construction works only for formulae in one free variable. The presence of this free variable makes the construction of V rather involved.

In the present paper, a different choice of V will be made such that there will be a decision procedure in polynomial time in the length of the formula for all existential formula $\exists \vec{y}\ \varphi(\vec{x}, \vec{y})$. In other words, the structure satisfies the relativized version of P=NP.

The construction is bazed on the following principles:

1. As in Poizat's case, a general Elimination Lemma for unary structures with generic predicates. (Section 2)

2. By the Elimination Lemma, the satisfaction of $\exists \vec{y}\ \varphi(\vec{x}, \vec{y})$ depends only of some local information on $\vec{x}$. For this information to be encoded by a quantifier-free formula $\beta(\vec{x})$ of polynomial length, the generic predicate V is chosen to be sparse. (Section 3)

3. The predicate V will encode the truth value of sentences only. In fact, only some special kind of $\forall\exists$-sentences are encoded. (Section 4)

4. The elimination algorithm works as follows: to check if $\vec{x}$ satisfies the formula $\exists \vec{y}\ \varphi(\vec{x}, \vec{y})$ first compute the neighborhood formula $\beta(\vec{x})$ then use the truth predicate V to decide whether the sentence $\forall \vec{x}\ [\beta(\vec{x}) \rightarrow \exists \vec{y}\ \varphi(\vec{x}, \vec{y})]$ does or doesn't hold in the structure. (Section 5)

*Universität Freiburg, Germany; and I.M.A.R. Bucharest, Romania. This research was supported by the DFG over a post-doctoral grant in the *Graduiertenkolleg Mathematische Logik und Anwendungen*.

2 Generic predicates

We consider first the language $\{s_0, s_1, p, a\}$ where s_0, s_1 and p are unary functions, and a is a constant. The ground set

$$B := \{ s_{\varepsilon_1} \circ s_{\varepsilon_2} \circ \cdots \circ s_{\varepsilon_n}(a) \mid n \in \mathbb{N},\ \varepsilon_i \in \{0,1\} \}$$

is the set of all terms in a using the function symbols s_0 and s_1. In fact s_0 and s_1 are two independent successors that freely generate B. The number n of function symbols s_0 and s_1 necessary to write down an element $x \in B$ is called its **length** and is denoted by $|x|$. The elements of a given length build a **level**. The level n consists in 2^n many distinct elements. The set B shall be called the **block**. The predecessor function p is implicitly defined on the block by the axiom:

$$\forall\, x\ [p(x) = x \leftrightarrow x = a]\ \wedge\ p \circ s_0(x) = p \circ s_1(x) = x.$$

If u and v are two variables, all satisfiable equation $\alpha(u) = \beta(v)$, where $\alpha(u)$ and $\beta(v)$ are terms, is equivalent in B with an equation of the form:

$$\vec{s}p^k(u) = p^s(v).$$

Indeed, equations like $s_0(t) = s_1(t')$ are not satisfiable in B, and equations like $s_0(t) = s_0(t')$ are equivalent in B with $t = t'$. Also, a p followed by an s always reduce. On the other hand, we cannot substitute $p(t) = p(t')$ with $t = t'$ in B.

We expand now the language by a unary predicate V; we call L the language $\{s_0, s_1, p, a, V\}$. A **coloured block** is B with its three functions, the constant and some interpretation (denoted also by V) of the predicate V. We call **black** those elements e sucht that $V(e)$ is true. The other elements shall be called **white**. To be shorter we write in the following only (B, V) for the structure (B, s_0, s_1, p, a, V). We will consider in fact structures that consist in disjoint unions of infinitely many coloured blocks, where one root interprets the constant a and there are no constants in the language for the other roots. Such a structure shall be denoted only by (M, V) instead of (M, s_0, s_1, p, a, V).

A **triangle of height** n is a conjunctive formula $T(x)$ as follows: For all $2^{n+1} - 1$ terms $t(x)$ of length $\leq n$ in the given variable x exactly one of the atomic formulas $V(t(x))$ or $\neg V(t(x))$ occurs in the conjunction. No other atomic formula does occur in the conjunction $T(x)$. There are exactly $2^{2^{n+1}-1}$ triangles of height n.

Let us use the alphabet of 15 letters $\forall$, $\exists$, x, $'$,), (, $\neg$, $\vee$, $\wedge$, s_0, s_1, p, $=$, V, a for writing down formulas. Different variables are built by x and $'$ like: x, x', x'', ... We denote by $|\varphi(\vec{x})|$ the length of a formula $\varphi(\vec{x})$ as word over this alphabet.

For a tuple $\vec{z} \in M$ we call m-**neighborhood** of $\vec{z}$ a conjunction of all formulas $T_{2m}(p^m(z_i))$, and for all terms y, y' occurring in these triangles, formulae $p(y) = y$, $p(y) \neq y$, $y = y'$, $y \neq y'$, exactly those of them which are realized by the tuple $\vec{z}$ in M. If the tuple consists in only one element, we speak about an **individual neighborhood**.

Defined like this, the m-neighborhood of a tuple is an information of exponential length in m. Happily there are predicates V with the following property: one can completely describe m-neighborhoods by quantifier-free formulas of lengths depending only polynomially in m. For this, there are not to many neighborhoods of radius n in the structure, and they have a previsible shape. Such predicates will be introduced in the next section.

The predicate V is called **generic** if it satisfies the following condition G:

$$G: \quad \text{if } (M, V) \text{ realizes some finite individual neighborhood } \mathcal{N}(x)$$

$$\text{then } (M, V) \text{ realizes } \mathcal{N}(x) \text{ infinitely many times.}$$

A structure (M, V) that is an infinite disjoint union of identic coloured blocks has always a generic predicate.

Lemma 2.1 *Let (M,V) be a structure consisting of a disjoint union of (not necessarily identic) blocks such that V is a generic predicate. Consider a formula $\psi(\vec{x})$ which is logically equivalent with a prenex $\exists$-formula. Let $|\psi(\vec{x})| = n$. Then there is a quantifier-free formula $\lambda(\vec{x})$ such that $M \models \forall\, \vec{x}\ \psi(\vec{x}) \leftrightarrow \lambda(\vec{x})$. Moreover, all the terms in $\vec{x}$ and a occurring in $\lambda(\vec{x})$ have length smaller than $2n$.*

In other words, in order to decide if a tuple $\vec{z} \in M$ satisfies this $\psi(\vec{x})$, it is enough to know the $2n$-neighborhood of the tuple $(\vec{z}, a)$ and which individual $2n$-neighborhoods are realized in M.

Proof: This is exactly Poizat's "Lemme d'Élimination" proved in [5]. □

3 Sparse predicates

Let (M,V) be a structure consisting in an infinite union of coloured blocks. The predicate V is called **sparse** if it satisfies the following condition:

$$\forall\, x\ \ [\,V(x)\ \ \rightarrow\ \ \exists n \in \mathbb{N}\ \ \exists \vec{\varepsilon} \in \{0,1\}^n\ \ \exists r\ \ x = s_1^n s_0 s_{\varepsilon_1} \ldots s_{\varepsilon_n}(r)\ \wedge\ p(r) = r\,].$$

Lemma 3.1 *Let (M,V) be a structure such that V is sparse. For all $x \in M$ the following holds: if x is at distance $> 3m$ from its root, then the individual m-neighborhood of x contains at most one black point, which is of the form $s_1^n p^m(x)$ with $0 \leq n \leq 2m$.*

Proof: Let $h > 3m$ be the distance of x to the root. Suppose that there are two different black points in the m-neighborhood of x. At least one of them satisfies the following conditions: there is an $l \in \mathbb{N}$ such that the point is $2l+1$ far from the root, and moreover both $h - m \leq l$ and $h + m \geq 2l$ hold. From $h \leq l + m$ and $h \geq 2l - m$ one gets $2m \geq l$. Recall that $h \leq l + m$; this means $h \leq 3m$, which is a contradiction. □

Lemma 3.2 *Let (M,V) be a structure consisting in an infinite union of identical blocks such that V is sparse. There is a unit-cost algorithm such that for input $x \in M$ and $m \in \mathbb{N}$ it constructs a quantifier-free formula $\beta(x)$ which determines the individual m-neighborhood of x up to isomorphism. The algorithm works in time $O(m)$.*

Proof: Compute the sequence $x, p(x), \ldots, p^{3m}(x)$ and check at every step if you has got a root and if the argument has been an s_0 or an s_1 of the result. If you get a root, construct the $s_{0,1}$-term $t(\cdot)$ defining x, and output $x = t(p^k(x)) \wedge p^k(x) = p^{k+1}(x) \wedge p^k(x) = / \neq a$, for an appropiated $0 \leq k \leq 3m$. If you don't get any root down to depth $3m$, then look for a black point in the sequence $p^m(x), s_1 p^m(x), \ldots, s_1^{2m} p^m(x)$. If there is a black point, then output $x = t(p^m(x)) \wedge p^{3m}(x) \neq p^{3m+1}(x) \wedge V(s_1^i p^m(x))$ with appropiated $s_{0,1}$-term $t(\cdot)$ and i. If there is no black point we can write instead $x = t(p^m(x)) \wedge p^{3m}(x) \neq p^{3m+1}(x) \wedge \Sigma$. Here Σ is a new symbol meaning "there are no black points in the m-neighborhood". The value of m is clear in the given context. □

Before stating the Lemma about neighborhoods of tuples, we have to do some considerations about finite subsets of (M,V). Say that x_i and x_j are m-**dependent** if $\mathcal{N}_m(x_i) \cap \mathcal{N}_m(x_j) \neq \emptyset$. This is the case if and only if $p^m(x_i) \in \{x_j, p(x_j), \ldots, p^{3m}(x_j)\} \vee p^m(x_j) \in \{x_i, p(x_i), \ldots, p^{3m}(x_i)\}$. We observe that the m-dependence is symmetric and reflexive, but not transitive.

Consider a finite sequence $x_1, \ldots, x_k$ of elements of M. If we connect every two m-dependent elements x_i and x_j with the shortest path in M from x_i to x_j, all these paths build a finite set of binary subtrees of M. Every subtree marks an equivalence class for the transitive closure of the m-dependence, seen as relation over the finite set $\{x_1, \ldots, x_k\}$. A tree containing s of the k elements is completely described by giving $s - 1$ many equalities of the form $x_i = t(p^d(x_j))$

where $|t| + d \leq 3m$ and t is an appropiated term. If the set $\{x_1, \ldots, x_k\}$ produces q many trees according to the m-dependence, this situation can be completely described by displaying $k - q$ many equations and the information that those relations of equality between terms of the set $\mathcal{N}_m(\vec{x})$ which don't follow in M from the displayed equalities are not valid.

Lemma 3.3 *Let (M, V) be a structure like in the precedent Lemma. There is a unit-cost algorithm working in polynomial time in m such that for input consisting in a tuple $\vec{x} \in M$ of length k and $m \in \mathbb{N}$ it constructs a quantifier-free formula $\beta(\vec{x})$ which determines the m-neighborhood of $\vec{x}$ up to isomorphism. The algorithm works in time $O(mk^2)$ and the length of $\beta(\vec{x})$ is $O(mk)$.*

Proof: We get all individual m-neighborhoods of the elements x_i with the precedent lemma. By making $6m$ many equality tests for every pair (x_i, x_j) we find those pairs which are m-dependent and we compute the partition in sub-trees. For those pairs we compute the shortest true equality $x_i = t(p^d(x_j))$. This works as follows: The m-dependence means to get an equality of the form $p^m(x_i) = p^s(x_j)$ with $0 \leq s \leq 3m$. From the computations done for the individual m-neighborhoods we know the $3m$-history of both x_i and x_j; this means the equations of type $x_i = t(p^{3m}(x_i))$ with t term of length $3m$. From the meeting point we walk upwards through both histories and find the first point where they differ. For all pairs (x_i, x_j) the amount of work is in $O(m)$.

The formula $\beta(\vec{x})$ consists in a conjunction of: the k many individual m-neighborhoods, the $k - q$ many equations necessary to define the sub-trees, and the symbol Σ. Here Σ can occur only once, with the following meaning: $\bigwedge \neg V(t)$ for all terms in $\mathcal{N}_m(\vec{x})$ which were not displayed as black $\wedge \bigwedge t_1 \neq t_2$ for all pairs (t_1, t_2) of terms in $\mathcal{N}_m(\vec{x})$ such that $t_1 = t_2$ does not follow in M from the displayed equalities. Like before, the value of m is clear in the given context. □

The symbol Σ helps us to do a short description with minimal positive information for the exponentially complex object $\mathcal{N}_m(\vec{x})$.

4 The truth-predicate

We extend the 15-letter alphabet used to write down formulas with the symbols implication $\rightarrow$ and Σ. This last symbol is used to describe m-neighborhoods in a structure (M, V) with sparse predicate V. All these letters are encoded in binary words $\varepsilon_1 \ldots \varepsilon_5 \in \{0,1\}^5$.

We consider all pairs of formulas $(\beta(\vec{x}), \psi(\vec{x}))$ in the language (s_0, s_1, p, a, V) such that:

- $\psi(\vec{x})$ is logically equivalent with an existential formula $\exists \vec{y}\ \varphi(\vec{x}, \vec{y})$ where $\varphi(\vec{x}, \vec{y})$ is quantifier-free. Let n be the length of $\psi(\vec{x})$ in the 15-letter alphabet.
- $\beta(\vec{x})$ is a formula produced by Lemma 3.3 to describe the $2n$-neighborhood $\mathcal{N}_{2n}(\vec{x})$ for some tuple $\vec{x}$ of elements in some structure (M, V) consisting in an infinite union of identic blocks, with a root interpreting a and such that V is sparse.

Observe that the length of $\beta(\vec{x})$ in the 17-letter alphabet is only $O(n^2)$. Indeed, any variable occurs at most $O(n)$ times and is written in the form $x^{\prime\prime\prime\cdots\prime}$ with less than n many accents.

To get later a complete elimination of quantifiers, we encode also all the pure existential sentences $\theta = \exists \vec{y}\ \varphi(\vec{y})$. These are existential formulas with an empty set of free variables.

We consider all $\forall\exists$-sentences θ of the form:

$$\forall\, \vec{x}\ [\beta(\vec{x}) \rightarrow \exists \vec{y}\ \varphi(\vec{x}, \vec{y})].$$

Such a sentence θ of length l is encoded by the sequence of letters $\varepsilon_1 \ldots \varepsilon_{5l}$. We define the code:

$$[\theta] := s_1^{t+5l} \circ s_0 \circ s_1^t \circ s_{\varepsilon_{5l}} \circ \cdots \circ s_{\varepsilon_1}(a).$$

Here is t the smallest natural number such that $t + 5l \geq 8n^2$: if $5l$ is already greater than $8n^2$ we take $t = 0$. The elements $[\theta]$ defined here form the set of all codes.

The structure M is a countably infinite union of blocks, the first of them having a root that interprets the constant a. The construction is described as follows:

All blocks are copies of the first block: if b is a root then $V(t(a))$ if and only if $V(t(b))$. It is sufficient to define the colouration of the a-block. This makes the predicate V generic.

The elements which are not codes are coloured as follows: let all elements $s_1^n s_0^{n+1}(r)$ be black. All other non-codes are white. Only some codes will be black. This makes the predicate V sparse.

We order the sentences θ after the length n of the existential formula $\psi(\vec{x})$ inside the sentence, then according to the length of $[\theta]$, and lexicographically.

Before the first code is coloured, the structure contains only the black points given above. Points which have not been coloured yet are considered to be white.

Codes are coloured inductively.

If a sentence θ is true in the structure obtained after colouring the finitely many codes done before (and the corresponding elements in the other blocks), the code $[\theta]$ is painted black. If not, it remains white.

In the moment that a code becomes black, all the corresponding points in the other blocks become black also. □

Lemma 4.1 *Let (M, V) be a structure consisting in an infinite union of copies of a block, so that V is generic and sparse. Consider a sentence $\theta = \forall\, \vec{x}\ [\beta(\vec{x}) \rightarrow \exists\, \vec{y}\ \varphi(\vec{x}, \vec{y})]$ such that the existential sub-formula has length n. In order to know if θ is true in M it is enough to know the colour of terms $t(a)$ with $\mid t \mid < 2n^2$ and the list of isomorphism-types of individual $4n^2$-neighborhoods realized in M.*

Proof: According to Poizat's Lemma 2.1 the existential formula $\exists\, \vec{y}\ \varphi(\vec{x}, \vec{y})$ is equivalent to a quantifier-free formula $\lambda(\vec{x})$ containing terms in $(\vec{x}, a)$ of length $< 2n$. So we have to eliminate the universal quantifiers in $\forall\, \vec{x}\ \beta(\vec{x}) \rightarrow \lambda(\vec{x})$, respectively to eliminate the existential quantifiers in $\exists\, \vec{x}\ \beta(\vec{x}) \wedge \neg\lambda(\vec{x})$. We know that $\beta(\vec{x})$ is a conjunction. Worst case in a were a chain of n relations $V(t_k(x_k)) \wedge x_k = t_{k-1}(x_{k-1}) \wedge \cdots \wedge x_1 = t_0(a)$ such that all t_i are shorter than $2n$. For terms in x_i we get similarly the bound $4n^2$. □

The codes are coloured such that $V([\theta]) \leftrightarrow \theta$ in the final structure. A sentence θ is a $\forall\exists$ formula $\theta(a)$ without free variables. According to the Lemma 4.1 we could determine the truth of $\theta(a)$ if we know all about the terms $t(a)$ of length $\leq 2n^2$ (which are already done) and which individual neighborhoods of height $8n^2$ (equivalently: radius $4n^2$) will be realized in the ready structure.

Lemma 4.2 *Suppose that the predicate V has been defined for all codes corresponding to existential formulas of length $< n$, but still hasn't been defined for any sentence of length n or more. Then all individual neighborhoods of height $\leq 8n^2$ which will be realized by the (ready) structure (M, V) have been already realized in the current structure.*

Proof: All new black point which shall be painted in the future produces the following triangles of height $8n^2$: all $8n^2 + 1$ kinds of white triangle containing only one black point on the pure s_1 (right) side. But such triangles has been already produced before with the points $s_1^n s_0^{n+1}(r)$. □

Lemma 4.3 *To sum up, there is an L-structure (M, V) consisting in a disjoint union of infinitely many identic blocks such that the predicate V is generic and sparse, and for all encoded $\forall\exists$ formal L-sentences θ: $(M, V) \models \theta \leftrightarrow V([\theta])$.*

5 The satisfaction-problem

Let (M,V) be the structure constructed in Section 4. Consider the following satisfaction problem SAT over (M,V):

$$SAT \ : \text{ given an existential formula } \exists y_1,\ldots,y_s\ \varphi(x_1,\ldots x_k,y_1,\ldots,y_s) \text{ of length } 5n$$

$$\text{written binarily and a tuple } u_1,\ldots,u_k \in M, \quad \text{it is asked if } (M,V) \models \exists \vec{y}\ \varphi(\vec{u},\vec{y}).$$

Lemma 5.1 *The number k of different free variables occurring in the formula $\exists \vec{y}\ \varphi(\vec{x},\vec{y})$ of length n in the 15-letter alphabet satisfies $k(k+1) < 2n$. Consequently, the algorithm given by Lemma 3.3 for constructing the succint description $\beta(\vec{x})$ for the neighborhood $\mathcal{N}_{2n}(\vec{u})$ works in time $O(n^2)$.*

Proof: The variables encoded by the shortest words are x, x', x'' up to $x_{k-1} = x^{\prime\prime\prime\prime\cdots'}$. If every variable occured once, this makes a length of $1+2+\cdots+k < n$. The number of ordered pairs (u_i,u_j) is $k(k-1) < k(k+1) < 2n$. □

Theorem 5.2 *There is a deterministic unit-cost algorithm able to solve the problem SAT over (M,V) in uniform polynomial time $O(n^2)$ for existential formulae of length n. Consequently, the structure (M,V) satisfies* P = NP *for unit-cost computations and has fast quantifier-elimination.*

Proof: Consider an input of SAT of the form $\vec{\psi u}$ with $\psi(\vec{x}) = \exists \vec{y}\ \varphi(\vec{x},\vec{y})$ pure existential formula of length n and $\vec{u} \in M$ a tuple of the same length k as the tuple of different free variables $\vec{x}$.

Using Lemma 3.3 we get a quantifier-free formula $\beta(\vec{u})$ that determines up to isomorphism the $2n$-neighborhood of the tuple $(\vec{u},a)$. The algorithm takes time $O(n^2)$ according to Lemma 5.1. Now construct the following sentence θ:

$$\forall \vec{x}\ [\beta(\vec{x}) \to \exists \vec{y}\ \varphi(\vec{x},\vec{y})].$$

Compute the code $[\theta]$ in M and check if $V([\theta])$ does hold. In order to compute the code $[\theta]$ we must write $\beta(x)$ using the 15-letter alphabet. However, the total number of occurrences of the variables in $\beta(\vec{x})$ is only $O(n)$, so $[\theta]$ has length $O(n^2)$.

Recall that in (M,V) the sentence θ does hold if and only if $V([\theta])$ holds.

If θ holds, then $\exists \vec{y}\ \varphi(\vec{u},\vec{y})$. If θ does not hold, then there cannot be any tuple $\vec{x}$ with $2n$-neighborhood isomorphic with the corresponding neighborhood of $\vec{u}$ that satisfies $\exists \vec{y}\ \varphi(\vec{x},\vec{y})$, because the existential sentences with parameters in the structure depend only of this local information. (This is the original Lemme d'Élimination!) So $\vec{u}$ also doesn't satisfy $\exists \vec{y}\ \varphi(\vec{x},\vec{y})$. □

6 Commentaries

1. First of all, we justify the claim that the structure (M,V) has fast quantifier-elimination. For this, we recall the formalism introduced in [4]. We expand the language L to a language LC containing two new constants $0 := s_0(a)$ and $1 := s_1(a)$, the characteristic functions for the relations $=$ and V with output in $\{0,1\}$, and the standard selector function defined by $S(0,y,z) = y$, $S(1,y,z) = z$ and $S(x,y,z) = x$ for $x \notin \{0,1\}$. We observe that for $u,v \in \{0,1\}$, $\neg u = S(u,1,0)$, $u \wedge v = S(u,0,v)$ and $u \vee v = S(u,v,1)$. For an existential formula $\psi(\vec{x})$ let $\vec{\psi}$ the $\{0,1\}$-tuple of length $5n$ encoding the formula. We consider these elements $\{0,1\}$ to be elements of M.

   Now we can read the Theorem in the following way:

Corollary 6.1 *There is a recursive sequence (C_n) of circuits such that $|\, C_n \,| = O(n^4)$, C_n has $6n$ input-gates, only one output-gate, consists in gates computing a, s_0, s_1, p, $=$, V and S with unit-cost over M, and for all existential formula $\psi(\vec{x}) = \exists \vec{y}\ \varphi(\vec{x}, \vec{y})$ of length n the following holds:*

$$\forall\, \vec{x}\ [\, \exists\, \vec{y}\ \varphi(\vec{x}, \vec{y}) \leftrightarrow C_n(\vec{\psi x}) = 1\,].$$

Open questions: Poizat's limited elimination in [5] has linear term complexity. Can we obtain full elimination with polynomial term complexity instead of polynomial circuit complexity? Can we obtain full elimination in linear time (circuit, or even term)?

2. Like in the classical Theory of Complexity, P=NP produces the collapse of the whole polynomial hierarchy. Can we get more, for example a uniform collapse? We refine the question in the following form:

   Prove or disprove: Let (M, V) be a structure consisting in a disjoint union of blocks such that the predicate V is generic and eventually sparse. We already know that (M, V) has quantifier-elimination: for all formula $\psi(\vec{x})$ with arbitrary prefix there is a quantifier-free formula $\lambda(\vec{x})$ such that $M \models \forall\, \vec{x}\ \psi(\vec{x}) \leftrightarrow \lambda(\vec{x})$. Is there any polynomial $q(n)$ such that the following implication does always hold?

   $$\text{for all } \psi(\vec{x}): \qquad |\, \psi(\vec{x}) \,| = n \ \longrightarrow\ \text{all terms in } \lambda(\vec{x}) \text{ are shorter than } q(n).$$

   If this question had a positive answer, one could construct a structure like here with a truth-predicate for all formal sentences and with an uniform polynomial-time elimination procedure for all formulas; where the time depends only on the length of the formulas and not on its prefix-complexity. Of course, this would be somehow too nice to be true.

Acknowledgments: A. Hemmerling and C. Gaßner constructed structures with P=NP over the block with two independent successors in [2] and [1]. The generic predicates and the Elimination Lemma have been introduced by Poizat in [5]. The author already applied the generic predicates in [6]. The sparse predicates have been introduced by Hemmerling in [2]. The paper of S. Kripke [3] influenced the author for constructing a truth-predicate for sentences and then to use the same predicate as a satisfaction predicate for formulas. The author had many interesting conversation on these things with A. Hemmerling, C. Gaßner, B. Poizat, M. Ziegler. A lot of positive impulses came from B. Poizat and his permanent conviction that such a structure must exist. The author thanks also the unknown referee who pointed out improvements of precedent versions of this manuscript — such that the result became possible and the exposition won in clarity.

References

[1] **Christine Gaßner**: *A structure of finite signature with identity relation and with P=NP — A formal proof.* preprint, Universität Greifswald, 2004.

[2] **Armin Hemmerling**: *P=NP for some structures over the binary words.* Journal of Complexity, 21, 4, 557 - 578, 2005.

[3] **Saul Kripke**: *Outline of a theory of truth.* The Journal of Philosophy 72, 19, 690 - 716, 1975.

[4] **Bruno Poizat**: *Les petits cailloux.* ALEAS, Lyon, 1995.

[5] **Bruno Poizat**: *Une tentative malheureuse de construire une structure eliminant rapidement les quanteurs.* Lecture Notes in Computer Science 1862, 61 -70, 2000.

[6] **Mihai Prunescu**: *Non-effective quantifier elimination.* Mathematical Logic Quarterly 47, (4), 557 - 561, 2001.